The Journal of Computing Sciences in Colleges

Papers of the 38th Annual CCSC Eastern Conference

October 21st-22nd, 2022
DeSales University
Center Valley, PA

Baochuan Lu, Editor
Southwest Baptist University

John Wright, Regional Editor
Juniata College

Volume 38, Number 3 **November 2022**

The Journal of Computing Sciences in Colleges (ISSN 1937-4771 print, 1937-4763 digital) is published at least six times per year and constitutes the refereed papers of regional conferences sponsored by the Consortium for Computing Sciences in Colleges.

Table of Contents

The Consortium for Computing Sciences in Colleges Board of Directors

2

Following is a listing of the contact information for the members of the Board of Directors and the Officers of the Consortium for Computing Sciences in Colleges (along with the years of expiration of their terms), as well as members serving CCSC:

Karina Assiter, President (2022), (802)387-7112, karinaassiter@landmark.edu.
Chris Healy, Vice President (2022), chris.healy@furman.edu, Computer Science Department, 3300 Poinsett Highway Greenville, SC 29613.
Baochuan Lu, Publications Chair (2024), (417)328-1676, blu@sbuniv.edu, Southwest Baptist University - Division of Computing & Mathematics, 1600 University Ave., Bolivar, MO 65613.
Brian Hare, Treasurer (2023), (816)235-2362, hareb@umkc.edu, University of Missouri-Kansas City, School of Computing & Engineering, 450E Flarsheim Hall, 5110 Rockhill Rd., Kansas City MO 64110.
Cathy Bareiss, Membership Secretary (2022), cathy.bareiss@betheluniversity.edu, Department of Mathematical Engineering Sciences, 1001 Bethel Circle, Mishawaka, IN 46545.
Judy Mullins, Central Plains Representative (2023), Associate Treasurer, (816)390-4386, mullinsj@umkc.edu, UMKC, Retired.
Michael Flinn, Eastern Representative (2023), mflinn@frostburg.edu, Department of Computer Science Information Technologies, Frostburg State University, 101 Braddock Road, Frostburg, MD 21532.
David R. Naugler, Midsouth Representative(2022), (317) 456-2125, dnaugler@semo.edu, 5293 Green Hills Drive, Brownsburg IN 46112.
Grace Mirsky, Midwest Representative(2023), gmirsky@ben.edu, Mathematical and Computational Sciences, 5700 College Rd. Lisle, IL 60532.
Lawrence D'Antonio, Northeastern Representative (2022), (201)684-7714, ldant@ramapo.edu, Computer Science Department, Ramapo College of New Jersey, Mahwah, NJ 07430.
Shereen Khoja, Northwestern Representative(2024), shereen@pacificu.edu, Computer Science, 2043 College Way, Forest Grove, OR 97116.
Mohamed Lotfy, Rocky Mountain Representative (2022), Information Systems & Technology Department, College of Engineering & Technology, Utah Valley University, Orem, UT 84058.
Tina Johnson, South Central Representative (2024), (940)397-6201, tina.johnson@mwsu.edu, Dept. of Computer Science, Midwestern State University, 3410 Taft Boulevard, Wichita Falls, TX 76308.
Kevin Treu, Southeastern Representative (2024), (864)294-3220, kevin.treu@furman.edu, Furman University, Dept of Computer Science, Greenville, SC 29613.
Bryan Dixon, Southwestern Representative (2023), (530)898-4864,

CCSC National Partners

The Consortium is very happy to have the following as National Partners. If you have the opportunity please thank them for their support of computing in teaching institutions. As National Partners they are invited to participate in our regional conferences. Visit with their representatives there.

Platinum Level Partner

Google Cloud
GitHub
NSF – National Science Foundation

Gold Level Partner

zyBooks
Rephactor

Associate Level Partners

Mercury Learning and Information
Mercy College

Welcome to the 2022 CCSC Eastern Conference

On behalf of the CCSCE 2022 Conference Committee, we would like to extend a warm welcome to all attendees of this year's conference, which is being held at DeSales University in Center Valley, Pennsylvania. We look forward to seeing each of you at the conference and are excited about the sessions available to faculty and students. We hope you will take some time to enjoy the Lehigh valley area which is a part of the Great Appalachian Valley.

Thank you to everyone who submitted a proposal to the conference. We have participants from more than 24 schools in the region. The two-day program includes a keynote speaker, paper presentations, workshops, tutorials, poster presentations, and a programming competition. We have 5 faculty paper sessions and 2 student sessions this year. We received 24 excellent papers that underwent a double-blind review, 14 of which we were able to accept for presentations, yielding an acceptance rate of 58%. The paper topics include computer science, pedagogy, new initiatives, security and more. The selected papers will be included in the Journal of Computing Sciences in Colleges.

This year we will have 11 workshops and tutorials: The Blockchain Art Simulation (BARTS) and Experiential Exercises, Do you know your cookies?, Adding Interactive Content to Dive into Systems, a free online textbook for introducing students to computer systems, Git-Keeper: Streamlined Software for Automated Assessment Workflows, Incremental Picross Puzzle Development, Reflective Curriculum Review for Liberal Arts Computing Programs, Interesting Exercise to Demonstrate Self-balancing Trees, Using Metal's Expanded Map-based algorithm Visualization in Computer Science Courses. We hope these will be of interest to you.

Alongside the student papers, we continue to host the programming competition which will feature college and high school student teams from around the eastern region. We will also have informative workshops presented by Google and NSF.

This conference would not be possible without the help of our volunteers, reviewers, and conference committee. We appreciate their help and conference planning expertise. A special thank you to John Wright (Juniata College) and Mike Flinn (Frostburg State University) whose support, guidance, and advice were invaluable.

We hope you enjoy the conference and enjoy the beautiful Lehigh valley. Please let us know how we could make it better in future.

Pranshu Gupta and Kathleen Ryan
DeSales University
Conference Co-chairs

2022 CCSC Eastern Conference Committee

Pranshu Gupta, Co-Chair .... DeSales University
Kathleen Ryan, Co-Chair .... DeSales University
John Wright, Papers, Web Site .... Juniata College
George Dimitoglou, Papers .... Hood College
Susan Conrad, Posters .... Marymount University
Natalia Bell, Posters .... Marymount University
Dave Hovemeyer, Programming Contest .... Johns Hopkins University
Steven Kennedy, Programming Contest .... Frostburg State University
Timothy Highley, Workshops/Tutorials/Nifty Ideas/Lightning talks . La Salle University
Nathan Green, Regional Treasurer .... Marymount University
Michael Flinn, Regional Board Representative .... Frostburg State University

2022 CCSC Eastern Conference Steering Committee

Elizabeth Adams .... James Madison University
Steven Andrianoff .... St. Bonaventure University
Karen Anewalt .... University of Mary Washington
George Benjamin .... Muhlenberg College
Elizabeth Chang .... Hood College
Vincent Cicirello .... Stockton University
Susan Conrad .... Marymount University
George Dimitoglou .... Hood College
Michael Flinn .... Frostburg State University
Sister Jane Fritz .... St. Joseph's College
Nathan Green .... Marymount University
David Hovemeyer .... Johns Hopkins University
John Meinke .... University of Maryland University College
Karen Paullet .... Robert Morris University
Donna Schaeffer .... Marymount University
Jennifer Polack-Wahl .... University of Mary Washington
John Wright .... Juniata College

Leveraging IT to support USGS Science*

Keynote Speaker

Timothy Quinn
Associate Chief Information Officer (ACIO)
of the Office of Enterprise Information (OEI)
USGS

Biography

Tim leads the bureau in finding solutions for and managing optimal use of the U.S. Geological Survey's Information Management and Technology (IMT).

Career History and Highlights

Prior to his role as ACIO, Tim served as the Deputy Director for IT Shared Services within the Office of the Chief Information Officer (OCIO) at the Department of the Interior. There, he was responsible for service desk, telecommunications, hosting and end user services.

Previously, he worked in a variety of positions within the United States Department of Agriculture, Forest Service and the Bureau of Land Management in Washington, D.C., California, Oregon, and Washington. Duties included database development, operations research modeling, programming, providing communications support for wildland-firefighting, service desk, records/Freedom of Information Act management, and telecommunications.

Tim also worked as the GIS Administrator for the Regional Ecosystem Office coordinating interagency information management activities for the President's Northwest Forest Plan. Prior to Federal service, Tim was an associate in a forestry consulting firm providing a variety of forest inventory, marketing, reforestation, and timber harvesting services to private landowners, industry, and tribal governments.

Education

Tim holds a bachelor's of science degree from the University of California, Berkeley, in forest management with experience in forest genetics research and a master's of science in management information systems from California State University, Sacramento.

About the U.S. Geological Survey

Created by an act of Congress in 1879, the U.S. Geological Survey has evolved over the decades, matching its talent and knowledge to the progress of science and technology. The USGS is the sole science agency for the Department of the Interior. It is sought out by thousands of partners and customers for its natural science expertise and its vast earth and biological data holdings.

Music as Motivation in CS Education*

Banquet Speaker

John P. Dougherty
Department of Computer Science
Haverford College
Haverford, PA 19041

Abstract

As young students, we all were asked to sing in school, sometimes as a break from other activities, but also as a way to engage with the material differently. I have been using songs, and to some degree songwriting, in computer science courses for about 20 years. In this presentation, I am asking for more than a receptive audience – I am hoping for one that sings with me. In this way, we may all learn about how to use music (and maybe other unplugged activities) to help students learn computer science concepts.

Biography

John Dougherty, or "J.D.", has worked at Haverford College since 1997. He is a Senior Member of the ACM and has been involved with ACM/SIGCSE. He presently studies dependable parallel computing (i.e., performability), computing education, and accessible computing.

Before Haverford, J.D. was an assistant professor of mathematics and computer science at the Philadelphia College of Textiles & Science, now Thomas Jefferson University, East Falls Campus. He received his B.A. in Mathematics and Computer Science from La Salle College, his M.S. in Computer Science from Drexel University, and his Ph.D. in Computer and Information Sciences from Temple University.

An Algorithm for Determining if a B-Tree Value Can be Changed in Place*

Aaron Harman and Daniel Spiegel
Computer Science and Information Technology
Kutztown University of Pennsylvania
Kutztown, PA 19530
aharm477@live.kutztown.edu spiegel@kutztown.edu

Abstract

Tree structures are used in many applications. Speed of access is a goal and is better predicted when a tree data structure is balanced. Among many variations, the B-Tree is guaranteed to always be balanced. It also permits quick storage and retrieval of data. But in most literature pertaining to tree data structures no consideration is given to methods to streamline changing data. In this work, an algorithm for determining whether changing values in place or within the same B-Tree node will be developed. It provides experience in algorithmic development for students and may lead to improvements in efficiency in applications herein noted.

1 Introduction

A graph consists of a set of *nodes* and a set of *edges* [11]. An edge provides a connection between two nodes. Nodes need not be connected to other nodes; they can also be connected to one or more elements in the set of nodes. A connected graph is one where every node is coincident with at least one edge. Part of graph theory, a *tree* is a connected graph that contains no cycles. A tree consists of a set of nodes and a set of directed edges, where each edge connects a pair of nodes with the path specifying a source and destination.

The trees referred to in this work are rooted, meaning that:

- one node r is designated as the *root*
- every other node c in the tree is connected by a single edge from one other node p.
  - p is denoted as the *parent* of c, and c is denoted as a *child* of p.

The depiction of a general tree is ordinarily vertically oriented with the root node on top. All nodes are connected to the root by one or more edges and are depicted vertically in levels below the root, where Level l consists of all nodes that are reachable by traversing l edges from the root.

A *binary tree* is a tree where a node can have no more than two children, i.e. any node can have zero, one, or two children. Each node has a left and a right subtree, with the corresponding child as the root; note that one or both can be empty. Those with empty subtrees, i.e. no child nodes, are leaves, found at the lowest levels of a tree in a *leaf* node.

A *binary search tree* (BST) is a binary tree ordered according to its data. That ordering is achieved via the following rules, which apply to every node n in a BST:

- The data in every node in a node's left subtree is less than (based on the defined outcome of 'less than' for the data type) the data in n.
- The data in every node in the right subtree is greater (based on the defined outcome of 'greater than' for the data type) than the data in n.

This invariant strictly orders the data, and results in a BST being an extremely efficient container for storing data ordered alphanumerically, according to the data, with $O(\log_2(n))$ insertion and search algorithms. Insertion of new data always occurs with the placement of a new leaf node in the tree.

A *B-Tree* is a generalization of the BST where a node is not limited to a single datum or two children. Its *order* is the maximum number of children each node may have. A node with k children has k-1 datum. For a B-Tree of order n where $n>0$, each node has between $n/2$ and n-1 datum. The exception is the root node, which has between 1 and n-1 datum. The reason behind these facts is outside the scope of this work but is found on the web and in any data structures text that approaches this topic, including [11].

The BST tree invariant still holds, as applied to each datum. If a node has k children, then a datum in the i^{th} position, elt[i], 0 <= i <= k-1, has children i and i+1. All elements in the i^{th} subtree are less than elt[i] and all elements in the $i+1^{st}$ subtree are greater than elt[i], $\forall_i$.

Insertion occurs following a search, where the tree is negotiated according to the value of the new data to insert. Starting from the root, if the root is NULL, i.i. it does not exist, the new data is used to form a root node. Otherwise the new data is compared with the data in the current node and the correct direction according to data values is followed until a leaf node is encountered. At that point the item is placed in the proper spot and rebalancing of the tree occurs, if required to maintain it as a B-Tree.

This methodology is also followed for a search for data. The difference is that a search can succeed at a level higher than the lowest in the tree if the data is matched. The manner of providing information that the search was successful (or not) is outside the scope of this work.

Printing traversals, as for any container, are $O(n)$. Deletion from a B-Tree has several cases that often require moving data and readjusting the tree Many data structures textbooks, such as [11], have a detailed description of the deletion process, which is omitted here, as it is again outside the scope of this work.

An algorithm that doesn't commonly appear (if it appears at all) in literature is editing (or changing) a value already stored in a B-Tree in place. A naïve solution is to delete the datum from the tree and re-insert it, allowing the insertion algorithm to do the work. But there may be a better way. An algorithm to determine ability to edit a value in place applicable to a BST was presented in [9]. Deleting and re-inserting data is relatively inefficient if updating the data can occur in-place while maintaining the tree invariant.

In this work, several tree applications will be briefly discussed, followed by a description of an algorithm that can be employed to determine whether a value in a B-Tree can be changed/edited in place or at least within its node while maintaining the tree invariant, or property, that all elements in the subtrees of a datum in the tree are lesser in the one to the left and greater to the right. The efficiency of the algorithms will be compared to the naïve delete and re-insert method, and finally, conclusions will be drawn.

2 Tree Applications

Tree structures are used in many areas to implement real-world applications, and this goes back decades. For example, in [1] a method for using a tree search to implement queries in a data collection is described. Binary Indexed Trees, introduced in [2], maintain numerical sequences on which operations can be carried out using tree-style operations, of course with time complexity of $O(\log_2(n))$.

Also, of historical significance are tree representations of expressions. [7] describes the creation of trees to hold arithmetic expressions in such a manner as to minimize the number of required accumulators. Later, Genetic Programming was an evolutionary technique that employed tree structures, as described in [5].

One of the most prominent tree applications is found in genealogy. [8] described manipulating genealogical data for optimal storage via tree structures. Use of tree structures to optimize memory use was mandatory at a time when the cost of memory was astronomical and its availability in any digital computer quite limited, particularly relative to current times.[4] discussed traversal of tree structures in memory using shift registers. This sort of innovation was necessary to maintain computing time within reasonable bounds.

B-Trees are specifically used in database and other storage implementations. A comprehensive treatment is found in [6]. The algorithm to be presented herein may well be applicable to two other works, [3] and [10], which approach the concept that simple rebalancing of B-Trees, which must be considered on insertion and removal of values, can be augmented by periodically rebalancing the trees for more efficient access. A reorganization might benefit from being able to change values in place rather than deleting and re-inserting.

3 Algorithm to Determine Whether a Change can Occur in Place

In order to determine whether a change/edit can occur in-place in a B-Tree, the closest values, greater and less than, to the value to be affected must be determined. Once that has occurred, if the replacement value is between the values closest to the current value, the change/edit can occur in place.

Thus, the algorithm must provide a method for determining the closest values in the tree to the data to be changed. The algorithm makes the determination of neighboring values based on a target value's relative location in the B-Tree.

3.1 Relative Location Algorithm

This algorithm determines closest values by looking at the location of the value within its node and also at the whole node's location relative to other nodes.

1. Data to be Changed is Within Non-Leaf Node

   Each individual value in a non-leaf node in a B-Tree, by the B-Tree rules, must have a child to its immediate left and right. As with the BST, the upper and lower bounds for a replacement value in a non-leaf node are the rightmost value in the left sub-tree and the leftmost value in the right sub-tree, respectively. Consider the B-Tree in Figure 1. In this tree, if I is the value to be changed, it can be changed in place if its new value falls between H and J.

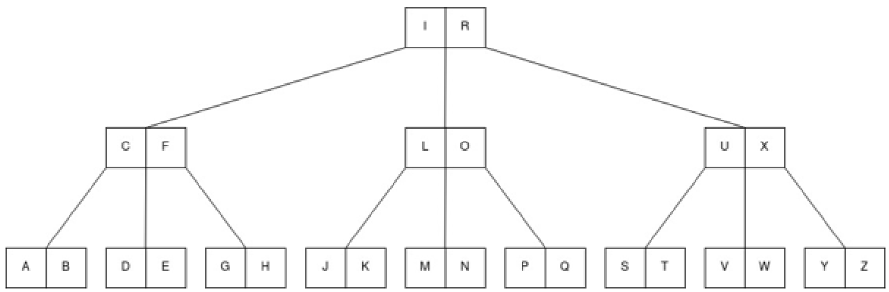


Figure 1: Three Level B-Tree

To find the lower bound for the new value in this position, one must enter the child node to the left of the location in question. From here, follow the rightmost child of the node(s) encountered until you have reached a leaf node. In that leaf node, the rightmost value is the lower bound.

Finding the upper bound is the same process but moving in opposite directions. First, enter the child node to the right of the location in question. From here, follow the leftmost child of the node(s) encountered until you have reached a leaf node. In that leaf node, the leftmost value is the upper bound.

2. Data to be Changed is Within Leaf Node

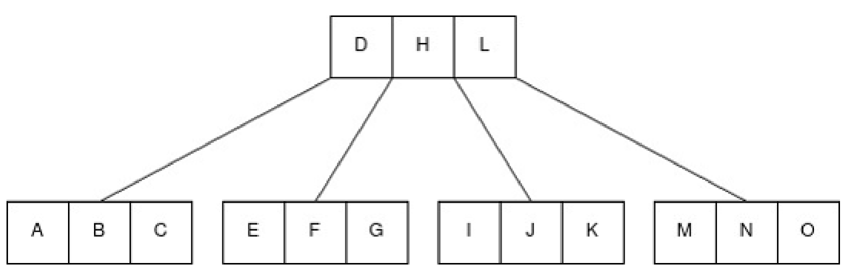


Figure 2: Two Level B-Tree

For a value in a leaf node (one on the lowest level of the B-Tree) determining whether it can be changed in place depends on neighboring values or a neighbor and/or a value in a parent or parents (if the leaf only has one value) when a value is at the edge of a node.

Consider the B-Tree in Figure 2. If the value to be replaced is F, its new value must fall between E and G. In a leaf node, if the target value has a neighboring value within the node to either side, these two values are its upper and lower bound. The value to its left will be its lower bound, and the value to the right its upper bound.

This method for leaf nodes must be changed when the value in question is missing a neighbor to either or both sides. Consider again the B-Tree in Figure 1. In this tree, if J is the value to be replaced, its upper bound can be K, as that is its neighbor to the right within its node. However, it is missing a neighbor to the left within its node. In this case, the lower bound will be the most recent left parent of this node. The most recent left parent refers to value of the closest ancestor of this node for which this node is found in its right child sub-tree. In the case of value J, this would be value I.

This same algorithm can be applied for a missing right neighbor. Consider value E in the B-Tree in Figure 1. This value has a lower bound in value D, its neighbor to the left. Its upper bound would be the most recent right parent of its node. In this case, that would be value F. If the leaf node only contains a single value, this method can be used for both the upper and lower bounds.

When finding a most recent parent in place of a neighbor within the node, it is possible that there will be no such parent. This indicates that the value is an extreme value and is either the greatest or least value in the tree, given a missing upper or lower bound, respectively. In these cases, a missing upper bound should be treated as infinity, and a missing lower bound should be treated as negative infinity. This alternative will always be necessary for the leftmost and rightmost values in the node of a B-Tree with only one node.

3.2 Replacing a Value in the Same Node by Rearranging

If the value to be replaced is in a leaf node, it can be safely replaced without entirely removing and re-entering a new value in some cases, even when it cannot be replaced in the same location. In this case, the new value must land between the most recent left parent and most recent right parent of the target node. For example, a replacement value in node J-K in Figure 1 would have to fall between the values of I and L. The old value will then be removed from the node, and the new value inserted in such a place to keep the contents of the node sorted. Replacing a value in this way will not violate the tree invariant.

```
function replace_node(Key, Replacer)
{
    find(Key, Replacer, -8, 8, Root)
}

function find(Key, Replacer, LastLeft, LastRight, Node)
{
    if (Node exists) {
        i = 0;
        loop until Node.Data[i] >= Key or i == Node.EltsInNode
        // If we didn't go through all of the nodes
        if (i < Node.EltsInNode) {
            if (Node.Data[i] == Key)
                return replace(i, Replacer, LastLeft, LastRight, Node);
            // Data[i] is 1st greater than key...go to its left
            if (i == 0) {
                // If we are in the leftmost child, dont pass in a new left parent
                return find(Key, Replacer, LastLeft, Node.Data[i], Node.Data[i].LeftChild);
            } else {
                // Otherwise, pass the data on either side as the most recent parent
                return find(Key, Replacer, Node.Data[i-1], Node.Data[i], Node.Data[i].LeftChild);
            }
        }
        // If we went through all the nodes, we must need to go into the rightmost child
        // Pass along the rightmost data as the most recent left parent, keep the right parent as-is
        return find(Key, Replacer, Node.Data[Node.EltsInNode-1], LastRight, Node.LastChild);
    }
    else
        return false ;
}

function replace(Index, Replacer, LastLeft, LastRight, Node)
{
    Key = Node.Data[Index];
    lowerBound = -8;
    upperBound = 8;

    // find the upper and lower bound
    if (Node has Children) {
        // find the lower bound
        Check = Node.Data[Index].LeftChild;
        // find the rightmost child node of the left subtree
        loop {
            if (Check has Children)
                Check = Check.LastChild;
            else
                break;
        }
        // the lower bound is the rightmost data of the rightmost node of the left subtree
        lowerBound = Check.Data[Check.EltsInNode-1];

        // find the upper bound
        Check = Node.Data[Index].RightChild;
        // find the leftmost child node of the right subtree
        loop {
            if (Check has Children)
                Check = Check.FirstChild;
            else
                break;
        }
        // the upper bound is the leftmost data of the leftmost node of the right subtree
        upperBound = Check.Data[0];

    }
    // for a childless node
    else {
        // leftmost data
        if (Index == 0) {
            lowerBound = LastLeft;
            upperBound = Node.Data[Index+1];
        }
        // rightmost data
        else if (Index == Node.EltsInNode-1) {
            lowerBound = Node.Data[Index-1];
            upperBound = LastRight;
        }
        // anything in-between
        else {
            lowerBound = Node.Data[Index-1];
            upperBound = Node.Data[Index+1];
        }
    }

    if (lowerBound < Replacer && Replacer < upperBound) {
        Node.Data[Index] = Replacer;
        return true;
    }
    else
        return false;

}
```

Figure 3: Complete Algorithm

3.3 Duplicate Data

Duplicate data was considered but not applied in developing the algorithm. It does not add complexity to include the possibility of duplicate data, it only requires a set of rules. If it is desired to change a value in a tree that appears multiple times, it must first be established how the multiples are recorded, i.e. whether nodes have a field within which a count can be maintained, or each occurrence appears in its own right. For the latter, the algorithm can be applied directly, but if nodes contain a counter for multiples the user then can be queried whether they wish to update one or all occurrences, and if it is only one occurrence, then the counter can be decremented, and the new value inserted. On the other hand, if all occurrences are to be updated, then the in-place algorithm can be applied.

3.4 Complete Algorithm

The algorithm in its entirety is in Figure 3. A call of replace_node() will cause a call to find() which in turn calls replace() if the search key (the value to be changed) is found in the B-Tree. Replace() determines the allowable bounds wherein a change can be made without invalidating the B-Tree.

We use a recursive method keep track of the recent left and right parenthesis to handle the more complex cases. The value is updated at the end of replace() if it is feasible. Otherwise the caller can receive confirmation that change in place isn't possible.

4 Conclusion

The motivation for development of this algorithm was educational in nature but considering the cited publications that discuss reordering trees to gain better performance, the concept presented here may have a role to play in B-Tree optimization. Further research is warranted to determine its veracity.

Clearly the complexity is no worse than that of the insertion algorithm, which is a worst case upper bound. Often this algorithm will be more efficient, particularly in the leaf node case but also in other situations analyzed.

The authors do state with certainty that discussion and development of this algorithm is an excellent educational exercise in algorithmic development.

References

[1] R.G. Casey. “Design of Tree Structures for Efficient Querying”. In: *Communications of the ACM* 16.9 (1973), pp. 549–556.

[2] P.M. Fenwick. “A new data structure for cumulative frequency table”. In: *Software-Practice and Experience* 24.3 (1994), pp. 327–336.

[3] E. Gudes and S. Thur. “Experiments with B-Tree Reorganization”. In: *Proceedings of the 1980 ACM SIGMOD International Conference on Management of Data* (1980), pp. 200–206.

[4] W.E. Kluge. “Traversing Binary Tree Structures with Shift Register Memories”. In: *ISCA ’76: Proceedings of the 3rd Annual Symposium on Computer Architecture*. ICSA ’76. 1976, p. 121.1.

[5] U-M O’Reilly. *Genetic Programming A Tutorial Introduction*. Power Point Presentation. 2011.

[6] Alex Petrov. *Database Internals: A deep dive into how distributed data systems work*. Sebastopol, California: O’Reilly Media, 2019.

[7] R.R. Redziejowski. “On Arithmetic Expressions and Trees”. In: *Communications of the ACM* 12.2 (1969), pp. 81–84.

[8] K. Shockley. “The Family Binary Tree”. In: *ACM ’76: Proceedings of the 1976 Annual Conference*. ACM ’76. 1976, pp. 546–550.

[9] Daniel Spiegel. “An Algorithm for Determining if a BST Node’s Value can be Changed in Place”. In: *The 16th International Conference on Frontiers in Education: Computer Science & Computer Engineering (STEM, ABET, ...)* FECS ’20. 2020, pp. 319–326.

[10] S. Wantanabe and T. Miura. “Reordering B-Tree Files”. In: *SAC ’02: Proceedings of the 2002 ACM symposium on Applied Computing*. SAC ’02. 2002, pp. 681–686.

[11] Mark A. Weiss. *Data Structures and Problem Solving Using Java, Second Edition*. Reading, Massachusetts: Addison-Wesley, 2000.

Increasing Engagement in Online Classrooms with Augmented Reality Filters*

David P. Eisenberg and Michael J. Lee
Department of Informatics
New Jersey Institute of Technology
Newark, NJ 07102
{de63, mjlee}@njit.edu

Pias Sarkar Tanmoy
Department of Computer Science
University of Asia Pacific
Dhaka, Bangladesh
pias1600@gmail.com

Abstract

The importance of engaging technology in education has never been more obvious than during the COVID-19 pandemic-driven era of distance learning. In this paper, we examine how an entertaining animated reward within a live online college classroom (using Facebook Group Messaging) affects students' participation and engagement. Students who answered questions correctly were rewarded with wearing an augmented reality (AR) filter of an animated college graduation cap until a subsequent student answered a question correctly and received the cap. We found that this reward system resulted in enhanced classroom engagement when it was used, particularly for students with little prior interest in the course material. Alternatively, however, students who were already interested in and studying the material prior to that class showed to be distracted by the same AR technique.

1 Introduction

Online learning has grown in popularity due to its flexibility, convenience, and accessibility. However, because they lack in-person, physical communication, online classes can struggle to maintain student motivation. The success of in-person courses to better maintain students' motivation and enrollment compared to an online course may be at least partly due to a lack of rewarding positive feedback and social acknowledgement which tend to be more common in an in-person environment. From this perspective, an online learning medium promoting social affirmation and positive rewards could potentially reinforce students' learning [22, 6, 15]. In more conventional online classes however, such as in Massive Open Online Courses (MOOCs), the technological environments have not lent themselves to delivering such social praise and acknowledgment. This has led to speculation that online learning environments may be responsible for a lack of student engagement [18] and negative learning outcomes [12].

As new media introduces innovative, synchronous learning environments [7], a greater sense of immersion is possible through virtual platforms [4]–such as augmented or virtual reality [5]. Cutting-edge technologies have the potential to make learning at least as effective as in-person environments, and even more engaging than traditional classrooms [19]. As decreased motivation in online course environments can lead to higher attrition [18, 25], educators must find innovative means of determining what virtual platforms, activities, and resources can best optimize engagement and learning for students [11, 16, 23].

During the COVID-19 pandemic, engaging remote students became essential, as schools around the world shifted to online learning. To explore how online synchronous learning can best promote student engagement, we used Facebook messenger, a very popular platform for personal use by college students in our target demographic. This platform offers entertaining augmented reality effects. In classes of approximately twenty students each, we tested two separate forty-five minute classes over two weeks of one semester.

2 Related Works

2.1 Benefits of Augmented (AR) in Educational Contexts

Numerous studies have shown that augmented reality (AR) environments can benefit student learning. Hung et al. observed 72 fifth-grade students who were separated randomly into three groups, each instructed with AR graphic books, picture books, or physical interactions [13]. The AR graphic book provided a practical and hands-on way for children to explore and learn and improved students' learning outcomes more than the other alternatives. Students in the AR graphics book group expressed high levels of pleasure and interest in adopting AR in the classroom in follow-up interviews.

Next, Eras et al. [2] separated 100 students into smart classrooms with AR interactions, and a control group of students in a traditional smart classroom. For students who interacted with AR, motivation, attention, and memorization increased, and average student satisfaction rose by 93.3% in relation to a control group that did not interact with AR in class. Similarly, Wen et al. [24] examined the effects of an AR-supported Chinese character learning game tailored for young classroom learners. With 53 second-grade pupils and two teachers from a Singapore government primary school, they found a significant increase in students' cognitive involvement in AR-supported activities. Moreover, students were more consistently engaged in learning activities that enabled self-generated contexts than expert-created content.

Kabtane et al. [8] tested two MOOCs, one with traditional videos, and the other with virtual simulations and activities that integrated AR. Researchers found that virtual simulations and AR helped students increase knowledge and understanding, resulting in significant reduction in dropout rates. Fedelli & Rossi [10] studied 50 ten-year-old children using an AR application to make their own digital films. Students' physical experiences shaped their virtual world experiences and the multimedia artifacts they generated.

Yang & Liao [26] created a virtual English classroom that used augmented reality and computer vision to improve language learning. In relation to a control group that received no AR or virtual effects, forty-four engineering graduate students who were learning English as a second language demonstrated increased effectiveness of cultural learning and promoted interpersonal communication between teachers and students.

Although beneficial in terms of communicating knowledge to students at a distance and with ease, distance learning suffers from a lack of engagement, active learning groups, visual experience, and physical presence. Chang et al. [3] found that 47 eighth-grade students exhibited positive emotional and cognitive participation when learning with mobile augmented reality technology. Sidhu & Kang [21] found that AR helped students to obtain three-dimensional information and perceive objects from the user's perspective, helping students learn more effectively. Santos et al. [20] reviewed 87 papers on AR learning experiences (ARLEs). They identified three educational benefits of AR: real-world annotation, contextual visualization, and vision-haptic visualization, demonstrating a mean effect size of 0.56, or moderate effect on learning.

2.2 Benefits of Positive Feedback and Rewards

There has been extensive work in the area of feedback and rewards on students' engagement and learning. Recent research examining 10,334 pupils using the School-Wide Positive Behavior Interventions and Supports (SWPBIS) method of school discipline, found that punitive consequences correlated with less intrinsic pro-social drive and slightly more extrinsic motivation [1]. It concluded that utilizing punitive measures to manage student conduct regularly promotes extrinsic pro-social incentive while suppressing intrinsic pro-social motivation. Another study observing 46 participants, found that positive social interactions influence social learning with peers [14]. Results showed that participants replied more quickly to peers who gave positive social reinforcement. Therefore, AR combined with positive feedback and rewards has the potential to increase students' engagement and learning.

3 Method

3.1 Video Chat Platform: Facebook Messenger

With its options to create open or private groups, Facebook has been used in academic classroom settings [17], offering educators the ability to create forums to post or answer student questions. While the Facebook video chat application remains less popular than competitors (e.g., Zoom and Skype) [9], it was designed around an engaging and popular social network platform, which was frequently used and well understood by students. We posited that Facebook messenger's AR effects may increase college computer science student class engagement, allowing instructors to offer behavior-reinforcing rewards, and could lead to increased student learning.

3.2 Participants and Their Environment

A sample of 62 student participants completed our entire study, which was a majority of the 86 students enrolled in the class. All 86 students in the class already had a Facebook Messenger account and were familiar with its use, since Facebook is a very popular communication tool among college students in Bangladesh. The use of such a familiar tool, where every student had an existing account, eliminated the need for any students needing to create accounts, as well as any learning curve for using Facebook Messenger as the platform.

Every student was enrolled and participating in an entry-level computer science course when the experiment was performed over two class periods in the middle of their semester. All of the students were undergraduates between the ages of 18-21. The students were all Bengali, studying at a university in Dhaka, Bangladesh. Participants were all native Bengali speakers, who were also able to read and understand English. However, we decided to conduct the entire study in Bengali, as the students were used to their classes being taught in that language. The same professor taught every class to minimize any instructor effects. Students were informed of the research study and given the option to opt-out of participating in data collection activities (i.e., the questionnaires, explained below). The 62 students considered part of the study consisted of only those who did not opt-out of the study. The 62 students included in the study also fully attended both the first and second online classes that used Facebook Messenger Group Video Chat, and completed all surveys about their experience. No participants received any compensation for their participation.

The study was conducted over two 45-minute class periods, taught 10 days apart, in the middle of the semester. For simplicity, we refer to them as "Day 1" and "Day 2" for the remainder of the paper. All sessions of the class, except for the classes included in the study, were taught online but using Google Meet and never included any AR or VR effects. Day 1 was taught using Facebook Messenger with no AR visual effects, and covered introduction to object-oriented programming. The use of Facebook Messenger during the first session of this class was, in part, to eliminate the novelty effects that could come from newly introducing the Facebook video conferencing platform to the students.

During Day 2 of the studied classes, students were again taught using Facebook Messenger, but now utilized its AR filter feature in the form of a virtual graduation cap, and also covered introductory object-oriented programming. In order to eliminate any effects that course content could have on student interest or participation, the curriculum taught during Day 1 and Day 2 were specifically designed to be a continuation of the same object-oriented introduction, with the same topic, content area, and level of difficulty. In Day 2, a student who correctly answered a question from the instructor was rewarded with a virtual graduation cap on their video chat feed (i.e., on top of their head) for everyone else to see. That student would continue to "wear" the cap until another student answered a question correctly (who was then subsequently rewarded the cap). Only one person in the class had the cap at any given time.

We distributed three questionnaires during the study, primarily consisting of Likert scales, with ratings of statements/questions from 1 (low) to 7 (high). The first questionnaire (QS1) was given to students at the beginning of Day 1 (prior to any other class activity). The second questionnaire (QS2) was given after the end of Day 1, with students asked to respond within 72 hours of the end of class. The third questionnaire (QS3) was distributed at the end of Day 2, and students were once again asked to respond within 72 hours of the end of the class. 62 students completed all questions from all questionaires, and were therefore included in the results.

Table 1: First Questionnaire

QS1. Administered before Day 1 activities
1. How interested in the course material did you feel before you enrolled in this course?
2. How much of a background did you have in the course material?
3. Do you prefer lecture-only classes, or ones that provide activities during the class?
4. How much time do you spend learning the new day's material prior to class?
5. How much time do you spend reviewing class material AFTER class?

Table 2: Second Questionnaire

QS2. Administered after Day 1 (class without using AR filter features)
1. Did you find yourself more or less engaged by the use of classwork during class?
2. How interested are you in the topic that today's class covers?
3. How much of today's class material did you feel you already knew before class today?
4. How much prior experience have you had using Facebook messenger for video chat?
5. How did you find Facebook messenger as a tool for conducting today's class?
6. How often do you use video chat for personal use?
7. Which of the following video chat programs is your preferred platform for personal use?
8. Which of the following video chat programs is your preferred platform for a virtual class?

Table 3: Third Questionnaire

QS3. Administered after Day 2 (class with AR virtual graduation cap)
1. How much did visual effects during class today help you stay engaged and interested?
2. Were you more likely to answer questions or try harder in class, so that you could wear the graduation cap?
3. Did it distract you, and thus make it more difficult to follow the material, to see the graduation cap above another student's head, when a question was answered correctly?
4. Did you wear the graduation cap today?
5. Did you offer an answer to any questions during class today (wrong or right)?

4 Results & Discussion

For our quantitative analyses, we performed a combination of Pearson Rho Correlations, as well as Logistic and Ordinal Regression. We used a confidence of $\alpha = 0.05$ for all data analyses. The study used a within-subjects design, with 62 total participants. The following analysis discusses the results from these findings.

4.1 Relationships Among Variables

Numerous relationships were found among the variables by one-tailed Spearman's Rho Correlations, achieving significant results. We utilized Spearman Rho Correlations, rather than Pearson correlations because the variables were not continuous, and one-tailed tests for greater statistical power.

As detailed in Table 4, students who were most engaged in class without the AR feature correlated to being less motivated when classwork became a

Table 4: Variables Correlated to Engagement in Classwork without AR

	Coefficient	**p-val**
Studying and preparing before class	0.315	0.006
Studying after class	0.307	0.008
Topic interest	0.594	0.000
Enjoyed Facebook video chat for class without AR	0.324	0.005
Winning the graduation cap in the AR competition	−0.019	0.019

competition to obtain the graduation cap. These same students (who were engaged without AR) showed a relationship to having interest in the course material, studying both before and after class, and enjoying the Facebook video chat online classroom when AR was not used. This suggests that those students who were previously motivated in the online class may not have been better engaged by the use of the AR competition for the graduation cap.

Table 5: Variables Correlated to Answering Questions in Class with AR

	Coefficient	**p-val**
Studying before class	−0.313	0.007
Topic interest	−0.305	0.008
Better engaged in the class with the AR competition	0.321	0.006
Working harder in class to obtain the graduation cap	0.355	0.002
Distracted by other students winning the graduation cap	0.299	0.009
Winning the graduation cap in the AR competition	0.270	0.017

Answering questions in the class with the AR feature to compete for the virtual graduation cap, was negatively correlated with students studying before class and their interest in the class topic (see Table 5). However, answering questions in the class with AR correlated to the desire to obtain the graduation cap, being more engaged in the class with AR, being distracted by others winning, and ultimately winning the graduation cap. This suggests that those who were engaged by the AR may have been students who were less studious and less likely to enjoy the traditional online class environment, but might have become more motivated by the AR itself.

4.2 Binary Logistic Regression Analyses

The following analyses used binary logistic regression, since the dependent variable was a binary (yes/no) question, while the predictors were ordinal:

Binary logistic regression had a significant model fit ($p = 0.027$) with a χ^2 value of 27.2 (with more specific results shown in Table 6). The data shows that when students were more engaged with AR use, and also when students were more frequently using video chat, they were more likely to answer

Table 6: Predictors for Answering Questions with AR

	Wald	**B**	**p-val**
Frequency of personal video chat use	4.181	0.658	0.041
Improved class engagement when AR was used	4.275	1.051	0.039

more questions when the AR was used. When predicting students' answering questions correctly and obtaining the virtual cap, significant results showed that students who had prior knowledge or a background in the class performed better (detailed results in Table 7). A significant model fit was found ($p = 0.010$), with a χ^2 value of 30.67. We also found a near-significant result ($p = 0.050$) for students working harder to obtain the graduation cap is shown.

Table 7: Predictors for Obtaining the Virtual Cap in Class

	Wald	**B**	**p-val**
Prior knowledge of topic of today's class	7.232	0.993	0.007
Background in course material	3.885	0.448	0.049
Working harder in class to get the graduation cap	3.827	0.888	0.050

4.3 Ordinal Logistic Regression Analyses

To better analyze dependent variables that were ordinal in nature, ordinal logistic regression was used. In Table 8, the first predictor (top) had a significant model fit ($p < 0.001$), with a χ^2 value of 72.4. The second predictor (bottom) had a significant model fit ($p < 0.001$), with a χ^2 value of 71.96. One particularly useful result was that students who reported greater engagement due to the use of AR in class were more likely to work harder in class to obtain the AR-virtual graduation cap. This effect worked in both directions, as seen in Table 8:

Table 8: Ordinal Logistic Regression

Predictor for being engaged with AR in class	**Wald**	**Est**	**p-val**
Working harder in class to get Graduation Cap	35.44	1.715	< 0.001

Predictor for working harder for Grad Cap	**Wald**	**Est**	**p-val**
Being More Engaged in class with AR used	35.70	2.14	< 0.001

5 Limitations

Our study has several limitations. To minimize observer effects and response bias, we made questionnaires optional and anonymous. This resulted in only 62 out of 86 students participating and completing the questionnaires, and also for our data to not include demographic data such as age and gender. Therefore, our sample is only representative to those who participated in the questionnaires (which may reflect engagement), and does not include analyses comparing demographics (e.g., differences in male and female responses).

Moreover, while we did confirm with observations whether students participated during the class and received the cap as a reward, the actual data used, due to our desire to maintain anonymity, was self-reported in these students' questionnaire responses. In the future, we would plan to use identifiers to keep the identities of participants confidential, but also match observational data from the actual class. This would allow us to yield results not only from the surveys but also from real-time student reactions to the actual classes, and to the changes made between the different kinds of classes, while still maintaining student privacy and confidentiality.

Next, all of our students were from Dhaka, Bangladesh, which meant that our findings may not be generalizable to other locations. To address these concerns, we aim to conduct future studies at multiple locations, include more diverse populations and demographics, and collect more detailed information about the participants.

Furthermore, the use of the augmented reality filters were tested in one experimental class, and compared to one class where we used Facebook Messenger for group video chat. Within these constraints, we did our best to eliminate instructor (by using the same instructor and language throughout) and novelty effects (by comparing the results to a class where students were using Facebook messenger video chat but not with AR filters). However, a longer study that lasted one or even two semesters, or compared entire courses, over many individual classes, of students using AR filters, may reveal changes in student response, as well as longer term consequences and outcomes that our study cannot determine. We aim, therefore, to also conduct longer-term studies to see if these results continue, decrease or increase over time.

Lastly, we only tested one subject area, computer science, which cannot conclude if our findings are specific to teaching that one subject, or if our findings would have the same effect regardless of the subject taught. Therefore, future research should test multiple subjects and content-areas.

6 Conclusion

In this study, we found that augmented reality effects in an online classroom—specifically competing to wear an augmented reality graduation cap after answering question correctly—can predict enhanced classroom engagement, even for students with little prior interest in the course material. A variety of significant relationships between variables were identified, particularly related to using AR in virtual classrooms.

Interestingly, we found that those already engaged with course materials and were more studious reported that they did not prefer the AR feature and found the competition to win the cap distracting. However, these same students were the ones who answered questions correctly more often, resulting in their winning the hat frequently. This suggests that although these already-engaged students *report* that they do not like the AR feature, they *do* participate and are successful, and therefore the competitive AR feature does not have a major negative effect in their class participation or outcomes. On the other hand, those who were less engaged with course materials initially, found the AR feature and competition to win the cap made the course topic(s) more engaging and resulted in more participation (even though they might not necessarily provide the correct answer). Combined with the previous result, this suggests that adding a competitive AR feature leads to a net gain for students, because the already-motivated students participate and do well, while the initially less-motivated students increase their class participation and report enjoying the material more. Therefore, it may be beneficial for educators and developers to utilize and create competitive AR features for online courses. However, the utilization of this technology should be done so with the knowledge that certain already-high-performing students may find the intervention distracting, and therefore, not helpful.

References

[1] George G Bear, Jessica C Slaughter, Lindsey S Mantz, and Elizabeth Farley-Ripple. Rewards, praise, and punitive consequences: Relations with intrinsic and extrinsic motivation. *Teaching and Teacher Education*, 2017.

[2] Luis Chamba-Eras and Jose Aguilar. Augmented reality in a smart classroom—case study: Saci. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, 12(4):165–172, 2017.

[3] Hsin-Yi Chang, Jyh-Chong Liang, and Chin-Chung Tsai. Students' context-specific epistemic justifications, prior knowledge, engagement, and socioscientific reasoning in a mobile augmented reality learning environment. *Journal of Science Education and Technology*, 29(3):399–408, 2020.

[4] Chris Dede and Sasha Barab. Emerging technologies for learning science: A time of rapid advances. *Journal of Science Education & Technology*, 18(4):301–304, 2009.

[5] Chris Dede, Tina A Grotzer, Amy Kamarainen, and Shari Metcalf. Ecoxpt: Designing for deeper learning through experimentation in an immersive virtual ecosystem. *Journal of Educational Technology & Society*, 20(4):166–178, 2017.

[6] Darina Dicheva, Rebecca Caldwell, and Breonte Guy. Do badges increase student engagement and motivation? In *Proceedings of the 21st Annual Conference on Information Technology Education*, pages 81–86, 2020.

[7] David Eisenberg. School-wide classroom tv programming without tv's: Streaming media and fuze meeting. In *EdMedia+ Innovate Learning*, pages 887–887. Association for the Advancement of Computing in Education (AACE), 2012.

[8] Hamada El Kabtane, Mohamed El Adnani, Mohamed Sadgal, and Youssef Mourdi. Augmented reality-based approach for interactivity in moocs. *International Journal of Web Information Systems*, 2019.

[9] Betil Eröz-Tuğa and Randall Sadler. Comparing six video chat tools: A critical evaluation by language teachers. *Computers & Education*, 53(3):787–798, 2009.

[10] Laura Fedeli and Pier Giuseppe Rossi. A study on real/virtual relationships through a mobile augmented reality application. *International Journal of Digital Literacy and Digital Competence (IJDLDC)*, 5(1):10–20, 2014.

[11] Reshad Hassan, Sakib Hasan, Md Jubaer Hasan, Md Rafat Jamader, David Eisenberg, and Tanmoy Pias. Human attention recognition with machine learning from brain-eeg signals. In *IEEE ECBIOS*, pages 16–19, 2020.

[12] Starr Roxanne Hiltz. Collaborative learning in asynchronous learning networks: Building learning communities. Technical report, NJIT, 1998.

[13] Y-H Hung, C-H Chen, and S-W Huang. Applying augmented reality to enhance learning: a study of different teaching materials. *Journal of Computer Assisted Learning*, 33(3):252–266, 2017.

[14] Rebecca M Jones, Leah H Somerville, Jian Li, Erika J Ruberry, Victoria Libby, Gary Glover, Henning U Voss, Douglas J Ballon, and BJ Casey. Behavioral and neural properties of social reinforcement learning. *Journal of Neuroscience*, 31(37):13039–13045, 2011.

[15] Jihoon Kim and Darla M Castelli. Effects of gamification on behavioral change in education: A meta-analysis. *International Journal of Environmental Research and Public Health*, 18(7):3550, 2021.

[16] Michael J Lee. (re) engaging novice online learners in an educational programming game. *Journal of computing sciences in colleges*, 35(8), 2020.

[17] Shailey Minocha. A study of the effective use of social software to support student learning and engagement. *JISC*, 2009.

[18] Lin Y Muilenburg and Zane L Berge. Student barriers to online learning: A factor analytic study. *Distance education*, 26(1):29–48, 2005.

[19] Eric Nersesian, Margarita Vinnikov, Jessica Ross-Nersesian, Adam Spryszynski, and Michael J Lee. Middle school students learn binary counting using virtual reality. In *IEEE ISEC*, pages 1–8, 2020.

[20] Marc Ericson C Santos, Angie Chen, Takafumi Taketomi, Goshiro Yamamoto, Jun Miyazaki, and Hirokazu Kato. Augmented reality learning experiences: Survey of prototype design and evaluation. *IEEE Transactions on learning technologies*, 7(1):38–56, 2013.

[21] Manjit Singh Sidhu and Lee Chen Kang. Emerging trends and technologies for enhancing engineering education: An overview. *International Journal of Information and Communication Technology Education*, 6(4):38–48, 2010.

[22] Richard S Sutton and Andrew G Barto. Reinforcement learning: an introduction, ser. adaptive computation and machine learning. *Cambridge, Massachusetts: MIT Press*, 6:15–17, 1998.

[23] Faiza Tahir, Antonija Mitrovic, and Valerie Sotardi. Investigating the causal relationships between badges and learning outcomes in sql-tutor. *Research and Practice in Technology Enhanced Learning*, 17(1):1–23, 2022.

[24] Yun Wen. Augmented reality enhanced cognitive engagement: Designing classroom-based collaborative learning activities for young language learners. *Educational Technology Research and Development*, 69(2):843–860, 2021.

[25] An Yan, Michael J Lee, and Amy J Ko. Predicting abandonment in online coding tutorials. In *IEEE VL/HCC*, pages 191–199, 2017.

[26] Mau-Tsuen Yang and Wan-Che Liao. Computer-assisted culture learning in an online augmented reality environment based on free-hand gesture interaction. *IEEE Transactions on Learning Technologies*, 7(2):107–117, 2014.

Are Research Trends in the Consortium for Computing Sciences in Colleges Regionalized?

Nathan Green[1] and Karen Works[2]
[1]School of Technology and Innovation
Marymount University
Arlington, VI 22207
ngreen@marymount.edu
[2]Department of Computer Science
Florida State University
Panama City, FL 32405
keworks@fsu.edu
Both authors contributed equally to this research.

Abstract

Keeping up to date with research in the computing sciences can be a daunting task. Many academics turn to regional conferences that focus on improving computing education as a source of best practices and new technical trends. Do the different regions within the USA follow the same trends or does each region have its own specialty? We evaluate the proceedings of the nine regions of the Consortium for Computing Sciences in Colleges (CCSC) and compare research from 2019-2022 across each region using non-negative matrix factorization (NMF) topic modeling. Further we examine the Log-likelihood ratio for each region for detailed keyword analysis. Our results show how topics differ per region and what specialty trends these regions focus on.

1 Introduction

The Consortium for Computing Sciences in Colleges (CCSC) seeks to improve computer-oriented curricula and the use of computing as an educational resource; and to collaborate at the regional and national level with like-minded organizations [4]. CCSC is organized into nine regions; central plains (CP), eastern (E), midsouth (MS), midwest (MW), northeastern (NE), northwestern (NW), rocky mountain (RM), south central (SC), southeastern (SE), and southwestern (SW). CCSC members are predominantly from colleges and universities oriented towards teaching rather than research.

Computer science (CS) research and technology trends are evolving at an insurmountable rate. Beyond their course related duties, teaching CS professors devote a lot of time to service to their departments and universities. This leaves little time to keep up with technology trends. Attending regional CCSC conferences helps these professors stay up-to-date with technology trends. Are these regional conferences highlighting local or national computer science trends? To answer this question, we evaluated the proceedings of the nine regions of CCSC and compared the research across each individual region from 2019-2022.

2 Related Work

The need to discover trends in computer science research dates back at least to the 1990s through the use of surveys [5]. For example, [21] evaluated the ratings of 54 higher education experts at the New Media Consortium. They found six emerging technologies in university education: extended reality technology, application of AI in education, student achievement analysis, adaptive learning technology, improvement of instructional design, learning engineering and user experience design, and open educational resources. Modern techniques exist that evaluate much larger data sets in an unbiased, unopinionated manner.

Recently, automated techniques have been used to discover trends. Two approaches in topic modeling include Latent Dirichlet Allocation (LDA) [2] (a probabilistic approach) and Non-negative Matrix Factorization (NMF) [19] (a matrix factorization approach). For example, LDA has been applied to the discovery of trends in many domains [6, 1, 17], and NMF has been used to model for specific subtopics in computer science, such as human-robot interactions [10].

Text mining has been applied to various aspects of monitoring education trends in the computing sciences. Unsupervised text mining was used to identify trends in how ethics is integrated into introductory CS courses[16]. Using LDA [2], selected online forum posts were polled to identify common CS topics of interest and trends over time [13]. A standard database query of articles with keywords related to ‘computer’ and ‘instructional technologies’ between 2016 and 2020 was performed to locate trends in technology education [15]. [11] used NMF to model topics in 200 programming problems scrapped from the Aizu Online Judge system. Both LDA and NMF models were used to model common topics in online beginning Python programming tutorials [14]. Industry and academic trends were analyzed for student job readiness [8] and curricular changes [9] using unsupervised term analysis.

To the best of our knowledge, no one has examined CS Education research trends of a particular conference across different regions.

3 Methodology

3.1 Data Source

The Journal of Computing Sciences in Colleges (JCSC) contains the conference proceedings for each of the CCSC regional conferences. We used all nine regions from 2019-2022, with the exception of the MS and SC region, which were canceled for 2021. For MS and SC, we analyzed 2019-2022 where data were available. At the time of analysis, only two conferences had held their 2022 conference. Only the SW and SC regions have 2022 results included in this study. Python was used to extract the textual content from the English PDF version of each JCSC issue. Each region has various lengths due to the regional size and number of accepted papers. Table 1 shows both the total number of tokens extracted and the unique vocabulary size of each region.

Table 1: Tokens extracted Per Region

Region	EA	MS	MW	NE	NW	RM	SC	SE	SW
Tokens	80,309	25,938	39,240	80,659	37,502	49,085	31,143	66,494	17,953
Unique	19,440	7,971	11,125	19,303	11,088	13,234	9,635	16,864	6,167

3.2 Data Cleaning

The extracted text was converted to lower case and common stop words [12] and CCSC Journal specific phrases were removed. Then, synonyms were replaced with a single term [3]. The final result was divided into bi-grams [20]. Bi-grams were chosen over uni-grams to provide additional context to the topic. For instance, 'machine' may be used in many contexts as a uni-gram but 'machine learning' is a specific CS topic as a bi-gram.

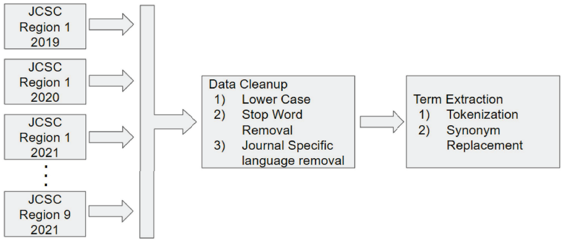


Figure 1: Data Cleanup Process Flow for JCSC Journals to Tokens

3.3 Topic Modeling

Both LDA [2] and NMF [19] models were developed, but after empirical experiments, NMF gave clearer and cleaner topics overall for our data set. From the results, we identified the topics and simultaneously classified the documents among these different topics (See Table 2). We then quantified the topics within each of the regional conferences.

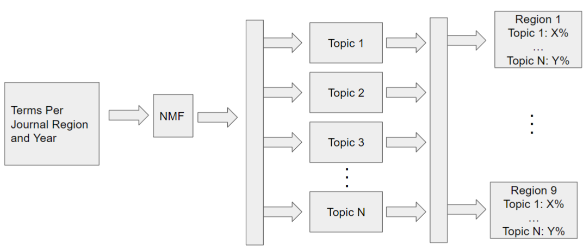


Figure 2: Process flow for converting journal tokens into topic models

3.4 Keyness

To evaluate differences in regions' language, we used the log-likelihood ratio statistic [7, 18] as a measure of keyness. This relative frequency ratio allows us to quantify overused and underused bi-grams for a region in respect to all other regions combined, namely, it identifies specialty topics per region.

4 Results and Discussion

We started with a baseline of CCSC as whole by looking at the overall frequencies of bi-grams for all proceedings and years across regions. These raw numbers give us a good idea of what the conference is about overall. Figure

3 shows the descending order of frequencies with each region indicated in a different color. While the conferences are generally thought of as computer education and pedagogy, that term is in second place to machine learning. Similarly neural network comes in 3rd place. After the top three we see a wide range of topics from software engineering, data structures, to student learning. The general theme would indicate researchers concerned with the education of a wide range of topics but with particular focus on machine learning and programming between 2019 and 2022.

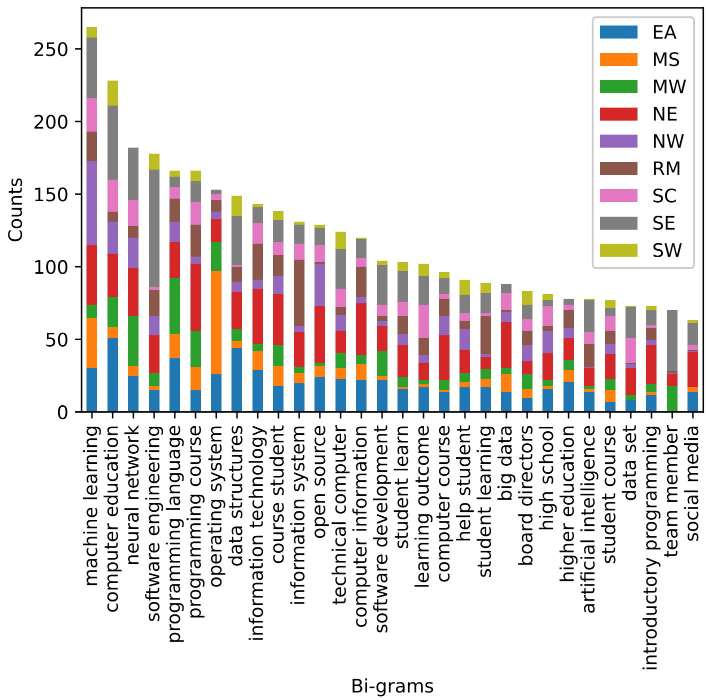


Figure 3: Top 30 terms across all CCSC regions after data cleaning

Table 2 contains the 10 automatically extracted topics via NMF, our self annotated label of the topic, and the top extracted 5 words in each topic. We see a variety of technical subjects, from machine learning, natural language processing, social media, to industrial control systems. Most of the rest of the topics are oriented toward pedagogy and evaluation. At points the authors looked past the top five words to come up with a label for the topic. If a clear topic was not apparent, the topic was labeled as miscellaneous.

Table 2: Topics and Terms

Topic	Label of Topic	Terms per Topic
1	Machine Learning	machine learning, software engineering, neural network, self-driving car, specifications grading
2	NLP	stop word, entity resolution, operating system, data analytics, scoring matrix
3	ICS	ics cybersecurity, control system, industrial control, time spent, real world
4	Programming	google firebase, usability testing, regular language, information system, student learning
5	Assessment	peer tutoring, success rate, graduate course, github classroom, final exam
6	Software Engineering	user story, team member, game design, hybrid course, regular expressions
7	Education	home automation, similarity score, experiential learning, raspberry pi, computer education
8	Data Pipeline	feature selection, local storage, web application, sentiment analysis, data set
9	Miscellaneous	female student, file system, planar graph, data sets, breast cancer
10	Workforce	number system, career fair, security operations, industry partners, formative assessment

Table 3 shows the topic breakdown per region. Each Region and year combination has it's main topic shown. The majority of proceedings' main topic is topic 1 which focuses on Machine Learning and Software Engineering. The second highest main topic was topic 7 which focuses on computer education and experiential learning. More importantly, each region has a distinct distribution across the 10 topics. This would indicate different levels of focus and trends in the different regions. For the most part, the scores across topics rise and fall more with region than with year, also supporting the idea that the topics are regional and not time-based.

Table 3: Distribution of Topic Weights by Region and Year

	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic7	Topic8	Topic9	Topic10	Main Topic
EA2019	27.40	8.20	8.92	19.21	10.92	2.08	20.91	18.13	9.78	6.04	1
EA2020	28.96	6.84	5.38	18.33	8.16	5.12	20.97	6.16	18.40	12.06	1
EA2021	27.78	6.20	3.87	16.10	11.36	0.12	18.28	7.39	8.94	12.76	1
MS2019	12.89	9.77	4.15	9.67	2.82	2.80	9.79	6.92	1.79	2.05	1
MS2020	13.64	14.86	3.46	9.90	0.05	0.52	8.59	9.31	6.22	11.21	2
MW2019	14.35	1.43	4.79	8.74	0.14	12.07	12.68	11.34	7.90	6.82	1
MW2020	12.34	3.35	12.00	8.83	2.58	1.83	13.39	5.96	10.16	6.40	7
MW2021	10.71	1.93	3.98	7.15	3.11	5.66	6.15	3.86	3.69	2.38	1
NE2019	30.27	3.58	5.43	15.90	6.79	5.30	19.18	11.48	6.64	5.45	1
NE2020	43.81	12.45	11.98	34.88	12.87	9.45	26.10	15.58	21.70	7.00	1
NE2021	16.78	4.87	11.61	8.44	6.93	4.37	11.31	8.98	5.22	5.56	1
NW2019	15.64	2.53	2.91	8.96	8.93	10.95	18.72	7.09	3.04	10.75	7
NW2020	7.96	3.82	2.94	1.54	0.20	0.00	4.63	1.66	3.51	1.56	1
NW2021	24.65	3.57	4.35	9.26	4.12	0.41	12.32	1.94	8.25	4.69	1
RM2019	21.10	0.70	3.70	11.91	5.56	1.17	8.62	7.10	11.33	6.49	1
RM2020	19.75	3.19	11.91	16.21	0.79	1.26	10.58	8.57	8.14	6.20	1
RM2021	16.26	2.34	0.83	17.40	7.14	7.16	16.71	7.91	8.72	6.27	4
SC2019	10.68	0.01	0.72	2.31	2.45	1.43	9.47	4.24	9.94	2.73	1
SC2020	17.94	4.82	3.51	6.33	3.80	0.82	18.88	5.32	3.67	5.78	7
SC2022	10.01	4.44	3.81	5.39	1.65	0.97	8.26	4.08	4.95	4.12	1
SE2019	18.64	1.84	0.81	15.76	0.47	17.64	14.17	10.22	3.79	6.32	1
SE2020	33.28	4.57	4.73	19.49	12.68	7.95	16.76	14.70	19.30	8.02	1
SE2021	16.54	1.64	5.73	7.26	1.31	0.62	10.43	6.74	5.60	2.80	1
SW2020	9.44	2.71	3.64	6.05	5.39	0.84	11.82	5.02	3.32	3.24	7
SW2021	4.11	0.76	7.17	3.19	2.54	0.39	2.04	1.12	2.22	4.43	3
SW2022	6.65	2.89	0.09	4.53	3.67	1.44	4.19	2.59	2.05	1.50	1
Top	NE	MS	MW	NE	NE	NE	NE	EA	NE	EA	

Table 4 shows bi-grams that were focused on or overused versus bi-grams that were used relatively less than other regions or underused. This does not indicated the main topic of the region but a few specialty topics of each region that are less talked about by other regions. For instance, the eastern region seems to have more papers on topics related to female students, while relatively less papers on software engineering. The Northeastern region has more papers on mixed reality and jupyter notebooks but less research dealing directly with data structures.

Table 4: Keyness by log-likelihood ration for each region

Region	Overused	Underused
EA	female students, spatial resolution, student paper, planar graph, file system	specifications grading, team member, data mining, entity resolution, software engineering
MS	entity resolution, operating systems, stop words, data analytics, scoring matrix	operating system, cyber security, nifty assignment, machine learning, academic programs
MW	career fairs, open career, code animation, virtual world, response phubbed	game design, network architecture, programming languages, recursive method, student student
NE	mixed reality, jupyter notebooks, bootcamp graduates, breast cancer, career education	real world, data structure, computer courses, entity resolution, big data
NW	anomaly detection, gallery walk, students interests, data mining, smart home	version control, machine learning, command line, logistic regression, course course
RM	google firebase, knowledge sharing, penetration testing, simulation game, erp simulation	information systems, student outcomes, active learning, open source, computer education
SC	self-driving car, feature selection, attitude change, boundary nodes, number systems	test cases, learning outcomes, computer security, data set, game development
SE	specifications grading, user stories, home assistant, home automation, face polygons	software engineering, experiential learning, operating systems, raspberry pi, cyber security
SW	string matching, pure c++, hybrid coral/c++, c++ group, exam similarity	final exam, peer tutoring, time spent, success rate, industrial control

5 Conclusion

Regionalized Computing Science Education conferences are important for the spread of pedagogy and technology trends. We performed an evaluation of JCSC issues from 2019-2022 with NMF for automatic topic modeling and log-likelihood ration as a measure of keyness and specialties of particular regions. The results show clearly each region focuses on computer education with a strong focus on machine learning, software engineering, and programming languages. While these are mostly universal across regions, our keyness analysis also shows that each region has its own specialties that are not universally shared. This can also be seen in the composition of topics per region, where each region has distinct distributions across the ten topics extracted.

References

[1] Saqib Aziz et al. "Machine Learning in Finance: A Topic Modeling Approach". In: *Econometrics: Econometric & Statistical Methods - Special Topics eJournal* (2019).

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3.null (Mar. 2003), pp. 993–1022. ISSN: 1532-4435.

[3] Jaime Carbonell et al. "Context-Based Machine Translation". In: *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. Cambridge, Massachusetts, USA: Association for Machine Translation in the Americas, Aug. 2006, pp. 19–28. URL: `https://aclanthology.org/2006.amta-papers.3`.

[4] *Consortium for Computing Sciences in Colleges (CCSC)*. `http://http://www.ccsc.org/`. Accessed: 2022-05-17.

[5] Sukhen Dey and Lawrence R. Mand. "Current Trends in Computer Science Curriculum: A Survey of Four-Year Programs". In: *SIGCSE Bull.* 24.1 (Mar. 1992), pp. 9–14. ISSN: 0097-8418. DOI: `10.1145/135250.134513`.

[6] Michael Dowling et al. *Machine learning in finance: A topic modeling approach*. Feb. 2019. DOI: `10.13140/RG.2.2.22110.69447`.

[7] Ted Dunning. "Accurate Methods for the Statistics of Surprise and Coincidence". In: *Comput. Linguist.* 19.1 (Mar. 1993), pp. 61–74. ISSN: 0891-2017. URL: `http://dl.acm.org/citation.cfm?id=972450.972454`.

[8] Nathan Green, Michelle Liu, and Diane Murphy. "Using an Electronic Resume Analyzer Portal (eRAP) to Improve College Graduates Employability." In: *Information Systems Education Journal* 18.3 (2020), pp. 28–37.

[9] Nathan Green, Xiang Liu, and Diane Murphy. "Revisiting an Educator's Dilemma: Using Natural Language Processing to Analyze the Needs of Employers and Inform Curriculum Development". In: *J. Comput. Sci. Coll.* 34.3 (Jan. 2019), pp. 97–107. ISSN: 1937-4771.

[10] Chan Hsu et al. “A Machine Learning Approach to Model HRI Research Trends in 2010 2021”. In: *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’22. Sapporo, Hokkaido, Japan: IEEE Press, 2022, pp. 812–815.

[11] Chowdhury Md Intisar et al. “Classification of Programming Problems Based on Topic Modeling”. In: *Proceedings of the 2019 7th International Conference on Information and Education Technology*. ICIET 2019. Aizu-Wakamatsu, Japan: Association for Computing Machinery, 2019, pp. 275–283. ISBN: 9781450366397. DOI: 10.1145/3323771.3323795. URL: https://doi.org/10.1145/3323771.3323795.

[12] Karen Sparck Jones, Peter Willett, et al. *Readings in information retrieval*. Morgan Kaufmann, 1997.

[13] Habib Karbasian and Aditya Johri. “Keeping Curriculum Relevant: Identifying Longitudinal Shifts in Computer Science Topics through Analysis of Q amp;A Communities”. In: *2021 IEEE Frontiers in Education Conference (FIE)*. 2021, pp. 1–7. DOI: 10.1109/FIE49875.2021.9637144.

[14] Laura Oliveira Moraes and Carlos Eduardo Pedreira. “Clustering Introductory Computer Science Exercises Using Topic Modeling Methods”. In: *IEEE Transactions on Learning Technologies* 14.1 (2021), pp. 42–54.

[15] C Nurzhanov et al. “Research and Trends in Computer Science and Educational Technology during 2016-2020: Results of a Content Analysis.” In: *World Journal on Educational Technology: Current Issues* 13.1 (2021), pp. 115–128.

[16] Sarah Parsons and Natalia Khuri. “Discovery of Research Trends in Computer Science Education on Ethics Using Topic Modeling”. In: *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2020, pp. 885–891. DOI: 10.1109/CSCI51800.2020.00166.

[17] Paritosh Pramanik and Rabin K. Jana. “Identifying research trends of machine learning in business: a topic modeling approach”. In: *Measuring Business Excellence* (2022).

[18] Paul Rayson and Roger Garside. “Comparing Corpora Using Frequency Profiling”. In: *Proceedings of the Workshop on Comparing Corpora - Volume 9*. WCC ’00. Hong Kong: Association for Computational Linguistics, 2000, pp. 1–6. DOI: 10.3115/1117729.1117730. URL: https://doi.org/10.3115/1117729.1117730.

[19] D Seung and L Lee. “Algorithms for non-negative matrix factorization”. In: *Advances in neural information processing systems* 13 (2001), pp. 556–562.

[20] Chade-Meng Tan, Yuan-Fang Wang, and Chan-Do Lee. "The use of bigrams to enhance text categorization". In: *Information processing & management* 38.4 (2002), pp. 529–546.

[21] Hongli Yan. "The Trends and Challenges of Emerging Technologies in Higher Education". In: *2021 2nd International Conference on Education Development and Studies*. ICEDS 2021. Hilo, HI, USA: Association for Computing Machinery, 2021, pp. 89–95. ISBN: 9781450389617. DOI: 10.1145/3459043.3459060. URL: https://doi.org/10.1145/3459043.3459060.

Student Perceptions of Computer Science Course Experiences During and After the COVID Pandemic*

Jennifer Polack and Karen Anewalt
Computer Science Department
University of Mary Washington
Fredericksburg, VA 22401
jennifer.polack@gmail.com, anewalt@umw.edu

Abstract

The COVID-19 pandemic provided motivation to shift course modalities and assignments to online experiences as many universities suddenly required all course instruction to move online. In response to the continuing global health crisis, many universities offered the majority of their courses with online modality for over twelve months following the beginning of the pandemic. In addition to classroom changes, students and faculty faced health, economic, and social stresses not previously experienced. We compare survey data of student perceptions of course experiences during their online experiences and after returning to a traditional in-person course environment. We identify student perceptions of tools that contributed to their success during online courses as well as challenges faced. Our survey data provides a window into how students felt during the online course experience and after returning to in-person classes during this unique period.

1 Introduction and Background

The global COVID-19 pandemic has a significant impact on higher education and provided never before experienced motivation to shift course modalities and assignments to online experiences. In response to the continuing global health crisis, many universities offered the majority of their courses in an online format for twelve to eighteen months following the beginning of the pandemic in 2020. Online course enrollments at the undergraduate level rose by 367 percent during fall 2020[3]. In addition to classroom changes, students and faculty alike faced health, economic, and social stresses not previously experienced. For example, a nation-wide survey of over 200,000 university students showed that students reported an increase in mental health struggles including general anxiety, social anxiety, and academic distress from 2020 through the following year[2].

In order to understand the impact of the modality shift to primarily online learning during 2020 and the impact of the return to primarily in-person learning in fall of 2021, we surveyed students at the University of Mary Washington, a small liberal arts college, about their perceptions of course experiences. Two surveys were conducted. The first survey data was collected in spring 2021, after students had experienced over a year of primarily online instruction. The second survey data was collected in spring 2022, after students had experienced approximately 9 months of primary in-person course instruction. By comparing perceptions, we identify changes that worked well and those that were less well received by students. The survey data also allows us to analyze how students felt during the online course experience and after returning to in-person classes during this unique period.

2 Majority Online Course Experience Survey

In order to evaluate student course experiences, we developed a survey to assess student perceptions of online learning experiences. Prior to 2020, the University of Mary Washington offered very few online courses. During the spring 2020 semester, all of our courses were required to shift to online delivery and the majority of courses were offered in a fully online modality during the fall 2020 and spring 2021 semesters. Data collected from the spring 2021 semester provides a unique view in how students perceived the shift from having nearly all of their courses in-person to nearly all of their courses online.

During the spring 2021 semester, the survey was sent to all undergraduate students majoring in computer science at the University of Mary Washington. The survey was sent to 206 students and 122 students responded to the survey (59% response rate). Of the 122 students who responded, 4 students were first-year students, 39 were sophomores, 47 were juniors, and 32 were seniors. Because students at the institution are not eligible to declare their major until after they have earned 28 credits, few first-year students were included in these survey results.

The initial survey included the following five questions:

1. What course modality was used for your online courses?
2. What organizational structure was used for course content?
3. For courses that used weekly Canvas modules, rate your perception of the course organization.
4. For courses that used the Canvas Assignments page, rate your perception of the course organization.
5. Should the University continue using teaching methods used during COVID after returning to in-person teaching?

3 Majority Online Course Experience Survey Results

Student survey responses were analyzed along with additional free-response comments. The question, "What course modality was used for your online courses?", required students to select from the following options: synchronous lectures, asynchronous lectures, or a combination. Students reported that 62% percent of their course were taught in a synchronous online format, 25% were taught asynchronously, and 13% of their courses used a combination of synchronous and asynchronous online instruction.

Instructors at the university were required to use the Canvas learning management system to communicate course content, Zoom for synchronous course meetings, and YouTube for asynchronous video content. In response to the question, "What organizational structure was used for course content?", 93.2% of students indicated that faculty used weekly modules to organize their course content, while the remaining 6.8% used the Assignments page within the Canvas course or another method.

Students were asked to rate their perception of course organization on a scale of effective, neutral and ineffective for courses that used weekly modules and also those that used the Assignments page within Canvas. Responses are shown in Figure 1. More than 90% of students rated courses using weekly modules as being effectively organized while only 36% of students rated courses using the Canvas Assignments page as effectively organized. Because students reported that weekly modules were used more frequently than other organizational mechanisms, they may have perceived this approach as being more organized than alternatives because it was more familiar. In additional comments, students reported that they preferred the weekly organization because it more closely paralleled reminders that an instructor would deliver verbally in an in-person course. They also indicated that the weekly organization of course content allowed them to concentrate more easily and was less overwhelming. Students perceived weekly course modules as an effective way to organize course materials and said that this organization style would be beneficial for in-person courses as well. Instructors at other institutions also concluded that students' perception of online course organization impacted their overall impression of the online learning experience[1].

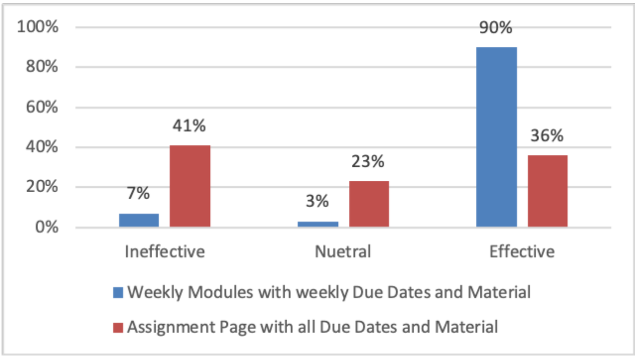


Figure 1: Student Perceptions of Course Organization During Online Learning.

In response to the question, "Should the University continue using teaching methods used during COVID after returning to in-person teaching?", students had different opinions. Students responded on a five point Likert scale of strongly agree, agree, neutral, disagree, and strongly disagree. Thirty-nine percent of students responded that teaching methods should continue, with 18% responding strongly agreed and 21% responding agreed. Twenty-seven percent indicated a neutral response. And 34% of students responded negatively, with 11% disagreeing and 23% strongly disagreeing. Interesting, the students' perceptions were highly dependent on their class year, with junior and senior-level students being more likely to want to continue with teaching methods used during online learning. In additional comments, upper class students reported that they enjoyed aspects of the shift to online courses because it provided more schedule flexibility for internships and other project experiences.

4 Return to Majority In-Person Course Experience Survey

As risks due to the public health crisis decreased mid-2021, the University of Mary Washington returned the majority of its courses to an in-person delivery. This provided a unique opportunity to gather additional data for students' perceptions at the conclusion of the 2021-2022 academic year when students had experienced a return to in-person instruction for approximately nine months.

A new survey was designed to collect data about student perceptions of their learning environment during the spring 2022 semester. The survey asked students about their perceptions of social connections and their impact on learning, their perception of online course experiences, and their feelings about returning to in-person course instruction.

Eleven questions were included in the survey:

1. What percentage of the students did you know in your computer science courses during COVID online learning?
2. During online learning, did knowing other students encourage you to ask for help when needed in your courses?
3. During online learning, did knowing other students contribute to your academic success?
4. Did you make friendships with other students in the class during online courses that continued once the university returned to in-person course instruction?
5. What methods helped with student interaction when the course was online?
6. Were these methods effective during COVID online instruction?
7. Do you still use those methods of interaction now that classes have returned to in-person instruction?
8. How hard was it to adapt to returning to campus and in-person instruction?
9. What was the most challenging part of returning to in-person course instruction?
10. Select how you felt while taking online classes during the pandemic?
11. Select how you felt returning to in-person classes in the 2021-22 academic year?

The survey was given to 282 computer science students and yielded 158 responses (56% response rate). First year students were not included in the survey because they did not experience online courses at University of Mary Washington during the pandemic. The survey responses were received from 46 sophomores, 52 juniors, and 60 seniors.

5 Perception of Return to Majority In-Person Course Experience Survey Results

As shown in Figure 2, fifty-eight percent of students indicated that they knew at least 15% of their classmates in computer science courses during COVID. Based on the university's average course size, this indicates that most students knew about 4 other students in each of their courses in the online learning environment. 20% of the students reported that they knew less than 15% of the students in their courses.

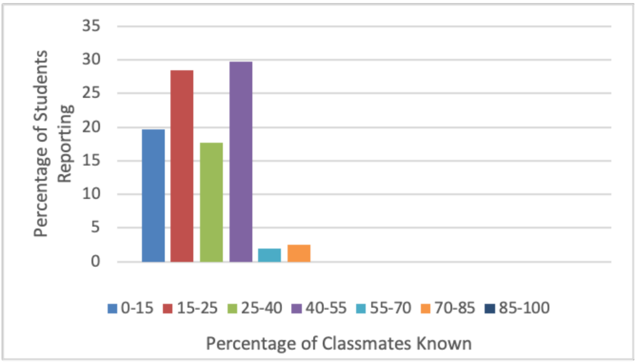


Figure 2: Percentage of Classmates Known.

Students were given three choices for the question, "During online learning, did knowing other students encourage you to ask for help when needed in your courses?" The response options were: yes, neutral, and no. The majority of students, 61%, reported that knowing other students did encourage them to ask for help. Students were given options to select yes or no in response to the question, "During online learning, did knowing other students contribute to your academic success?" Again 61% reported that knowing other students contributed to their academic success. These results suggest that having a small cohort of social connects in a class positively contributes to students' perception that they have the potential to succeed in the course and encourages them to seek help in courses when needed.

Students were asked to indicate (yes or no) in response to the question, "Did you make friendships with other students in the class during online courses that continued once the university returned to in-person course instruction?" The majority, 75%, reported that after returning to in-person classes, they had maintained friendships formed during online instruction. While instructors may have perceived limited opportunities for students to develop social connections during online instruction, the data suggests that students were able to successfully make lasting social connections during online course experiences.

In response to the question, "What methods helped with student interaction when the course was online?" students could select multiple answers from the following list: email, Canvas groups, Canvas discussion boards, Discord, Slack, and Trello. Students reported using all of these tools except Slack. Email was the most frequently used (100%), followed by Discord (40%), and Trello (33%).

Canvas discussion boards and groups were less frequently used with 21% and 10% respectively. While email was the most commonly reported method for course interaction, students rated it as the least successful for class engagement and student interaction. Discord was rated as highly successful for class engagement and student interaction. This is not surprising as Discord allowed students to discuss course topics and as questions related to material, but also provided students and instructors with more options for social interaction through off-topic channels that instructors could create at the request of students. Since returning to in-person instruction, 62% of the students indicated that they have continued to use these methods of interaction with peers.

Many instructors observed that students struggled as they returned to in-person instruction during the 2021-22 academic year[4]. We asked students to respond via a Likert scale to the question, "How hard was it to adapt to returning to campus and in-person instruction?" where 1 represented very difficult and 5 represented very easy. Fifty-four percent of students said it difficult or very difficult to adapt to returning to campus. This was evenly spread between sophomores, juniors, and seniors. When asked to respond to the question, "What was the most challenging part of returning to in-person course instruction?" students were given a list of six possible options and could select multiple options. The choices were attending classes in-person, adjusting to the workload, completing homework, lack of knowledge from prerequisite courses, speaking in class, and mental health. Students perceived their biggest challenges to be adjusting to workload (88%) and mental health challenges (65%). Student responses are shown in Figure 3.

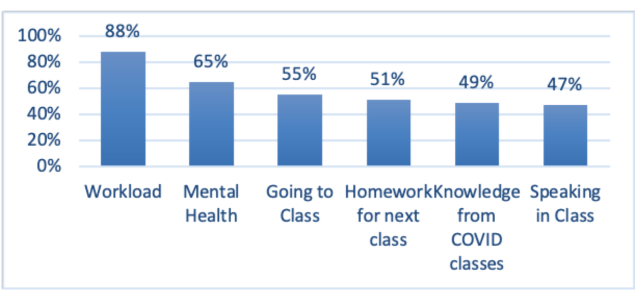


Figure 3: Most Challenge Aspects of Returning to In-person Instruction.

The final two questions asked students about how they felt during online courses during the pandemic and how they felt returning to in-person classes in the 2021-22 academic year. Students were given a list of both positive and negative feelings and selected those that applied. The choices were: overwhelmed, exhausted, defeated, bored, engaged, excited, interested, and optimistic. The responses are in Figure 4.

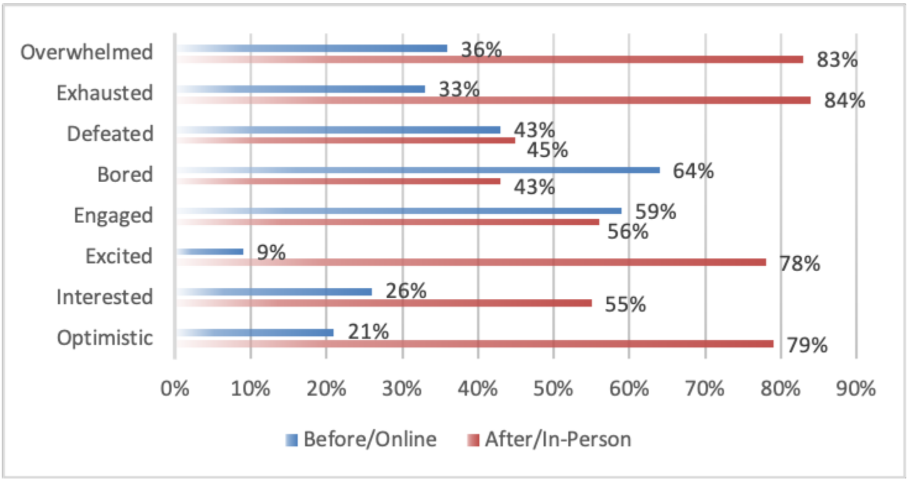


Figure 4: Students Feelings About Online Learning and In-Person Learning.

During the pandemic students reported feeling bored (64%) and engaged (59%) most frequently. After returning to in-person instructions, students reported that they felt exhausted (84%) and overwhelmed (83%) but also optimistic (79%) and excited (78%). The data is interesting because it suggests that students experienced both the online-courses during the pandemic and the return to in-person courses in opposing ways – with the majority reporting being both bored and engaged during online courses and exhausted, overwhelmed, and optimistic and excited all after returning to in-person courses.

6 Discussion of Responses to Both Surveys

Through conducting the second survey during the 2021-22 academic year, we confirmed that knowing students in their courses affected students' perceptions of success and their willingness to engage and ask questions in courses. While students had very different experiences positive and negative experiences during online courses, their social connections made them more willing to communicate questions and ask for help when needed. Students reported that tools like Discord that allowed them to communicate about course-related topics and non-course related topics contributed the most to their success and reported that they continued to use these tools after returning to in-person courses.

Other studies have shown that students experienced significant challenges to maintain personal wellness during the pandemic [6]. Our survey data suggests that many students are struggling as universities return to in-person instruction, with 84% reporting feeling exhausted, 83% reporting feeling overwhelmed, 88% reporting struggling with their workload, and 65% struggling with mental health challenges. These observations are supported by other studies that show that students have stress levels that are equal to or greater than during the pandemic[7].

Anecdotally, students reported that their feelings of exhaustion and being overwhelmed were caused by adapting to new course formats or content delivery mechanisms, returning to new and different living environments, and other new responsibilities. As some universities, including the University of Mary Washington, used alternate grading schemes (options for pass/fail grading rather than letter grades) during the pandemic, it is not surprising that students felt that returning to more traditional classrooms and grading schemes was different and sometimes overwhelming. Some universities also encouraged instructors to be particularly lenient and accommodating during the pandemic in order to decrease stress on students who were already overwhelmed by the public health crisis. Returning to more traditional classroom policies may also have played a role in how students perceived their return to in-person courses.

Some studies have suggested that universities should help students develop the necessary time and task management abilities to help them better cope with highly challenging situations like the pandemic[7]. Other colleges seem hesitant to challenge students right now because they are already experiencing significant stress[5]. To improve students' abilities to cope with stress, instructors could take it upon themselves to teach students how to deal with the stress of time management and self-regulation[7]. At this point we have identified some aspects of how students perceived their experiences during the online learning of the COVID pandemic and how they perceived their experiences after returning to in-person courses in 2021-22. We now need to consider how to best support students while encouraging their continued academic success in the years ahead.

7 Conclusion

The global COVID-19 pandemic prompted changes in university learning environments never before experienced. Student perceptions of tools that were helpful in promoting academic success and maintaining social connections during online course experiences can continue to be utilized in courses moving forward, regardless of course modality. Many instructors reported that students experienced difficulty in returning to in-person courses during the 2021-

22 academic year and our data about computer science student perceptions supports these observations. In order to best support students and encourage positive academic experiences, instructors will need to consider how to handle the range of emotions that students have related to course experiences. Course instructors should also consider how to support students in developing positive strategies for managing course workload and mental health, as these identified challenges are likely to continue in the near future.

References

[1] Yan Bai, Chunming Gao, and Bryan Goda. "Lessons Learned from Teaching Cybersecurity Courses During Covid-19". In: *Proceedings of the 21st Annual Conference on Information Technology Education*. SIGITE '20. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 308–313. ISBN: 9781450370455. DOI: 10.1145/3368308.3415394. URL: https://doi-org.umw.idm.oclc.org/10.1145/3368308.3415394.

[2] Center for Collegiate Mental Health. *Fall 2021: COVID-19 Impact on College Student Mental Health*. https://ccmh.psu.edu/index.php?option=com_dailyplanetblog&tag=covid-19.

[3] Doug Lederman. "Detailing Last Fall's Online Enrollment Surge". In: *Inside Higher Ed* (Sep. 16, 2021). URL: https://insidehighered.com/news/2021/09/16/new-data-offer-sense-how-covid-expanded-online-learning.

[4] Beth McMurtie. "A Stunning Level of Student Disconnection". In: *The Chronicle of Higher Education* (Apr. 5, 2022).

[5] Beth McMurtrie. "The Pandemic is Dragging On. Professors are Burning Out". In: *The Chronicle of Higher Education* (Nov. 5, 2020).

[6] Angela A. Siegel et al. "Teaching through a Global Pandemic: Educational Landscapes Before, During and After COVID-19". In: *Proceedings of the 2021 Working Group Reports on Innovation and Technology in Computer Science Education*. ITiCSE-WGR '21. Virtual Event, Germany: Association for Computing Machinery, 2022, pp. 1–25. ISBN: 9781450392020. DOI: 10.1145/3502870.3506565. URL: https://doi-org.umw.idm.oclc.org/10.1145/3502870.3506565.

[7] Luise VonKeyserlingk et al. "Stress of University Students Before and After Campus Closure in Response to COVID-19". In: *Journal of Community Psychology* 50.1 (2022), pp. 285–301.

Reinforcement Learning to Generate 3D Shapes: Towards a Spatial Visualization VR Application*

Charles Silverman[1], Liam Stewart [2], and Christian Lopez[1,3]
[1]Computer Science Department
[2]Electrical and Computer Engineering Department
[3]Mechanical Engineering Department
Lafayette College, Easton, PA 18042
lopezbec@lafayette.edu

Abstract

The objective of this work is to introduce a Reinforcement Learning (RL) approach to automatically generate 3D shapes of different complexities. This is with the goal to help tailor the spatial visualization task of a Virtual Reality (VR) application designed to help develop students' spatial skills. Spatial visualization skills are important skills needed and frequently used in the STEM fields. While VR has been used to help develop these skills, most of the existing applications do not necessarily tailor their content to students' skills level. Automatically generating 3D shapes can help VR applications tailor spatial visualization tasks to the skills level of students. The results of this work indicate that an RL agent is capable of creating an action policy that can generate 3D shapes with complexities similar to a given desired complexity provided. However, the results also show that the task of automatically generating 3D shapes that meet a given complexity is not trivial given the issues of sparsity in the reward space. Nevertheless, this work lays the foundation to leverage RL to automatically generate 3D shapes for VR applications designed to help develop students' spatial visualization skills.

1 Introduction

An individual's capacity to mentally manipulate and understand 3D shapes is known as their spatial or visuo-spatial ability [26]. Rotating and generating cross-sections of a shape are usual metrics for evaluating one's spatial ability or skill [18]. In STEM programs, both GPA and capacity to conduct self-monitored learning have been shown to relate to a student's spatial skills [33, 28]. Further reinforcing the importance of spatial skills, several fields, including Chemical Engineering, Civil Engineering, and Computer Science, have found a correlation between students' spatial skills and academic success [34, 18, 33]. Unfortunately, many STEM students lack these critical skills at the beginning of their studies. Moreover, standard instruction and class coursework at the introductory level might not be enough to thoroughly develop these skills [32].

Spatial skills could be learned through the manipulation of 3D objects [28]. For some, this is as simple as playing with building blocks as a child, whereas for others, it may need to be learned in an academic setting with repetitive exercises. Thanks to recent advancements and the penetration of Virtual Reality (VR) technology, educators are leveraging VR as a pedagogical tool [24, 16]. Moreover, it has been found that students could perform better within VR settings when compared to other traditional settings [20, 19]. The greater the immersion of the VR application, the greater the performance increase of the students [19]. Thanks to its unique characteristics that facilitate "first-person" experiences and allow users to interact with 3D virtual objects, VR is suited to teach and help develop spatial visualization skills. For example, some researchers have already started exploring this and have shown promising results [8]. However, researchers have also shown that students can have different levels of expertise which can directly impact their state of flow in a given application [25]. The Flow theory of motivation indicates that students would best execute the tasks whose difficulty aligns with their skills level. Consequently, any educational VR applications (e.g., ones designed to help develop spatial visualizations skills) that do not provide any tailored content or only enable users to interact with a limited set of content, might motivate users just for a short period of time (e.g., "novelty-effects").

Furthermore, the cost of VR technology has steadily decreased, thus increasing the economic accessibility of the devices for different socio-economic groups[21]. However, the resources needed to generate new VR content are still high. The existing challenges for generating new VR content might prevent educators to tailor their VR content to their students. Moreover, novelty effects could set in and lower students' motivation as they grow accustomed to the VR environments they are learning from [14]. Thus, new content might be required to counteract potential novelty effects.

The gaming industry has already leveraged methods to generate new content automatically. These Procedural Content Generation (PCG) methods have been used since the '80s, and are able to not only help reduce the resources needed to create content but also help with users' long-term engagement and motivation [9]. VR applications could greatly benefit from methods that can help generate content automatically. Specifically, VR applications designed to help develop spatial visualization skills could benefit from methods to automatically generate new 3D shapes. These 3D shapes could also be tailored based on students' skill levels, which could help improve their state of flow. One potential solution to achieve this would be to leverage PCG methods to generate new content [6, 17, 15], like 3D shapes of different complexities that students can interact with while developing their spatial visualization skills.

Based on the importance of spatial visualization skills and the advantages of VR to help develop these skills, the authors of this work have introduced a VR application designed to help develop spatial visualization skills in which students can use their hands to interact with 3D shapes and perform spatial visualization tasks [35]. This work extends previous efforts by introducing a PCG method based on a Reinforcement Learning approach to automatically generate 3D shapes of different complexities that students can interact with while developing their spatial visualization skills.

2 Literature Review

2.1 Virtual Reality for Education

As VR technology has grown more available, they have been introduced to educational mediums[27]. During the first uses of VR in education in the previous decade, close to half of them pertained to engineering[22]. Although the term immersive and not been formalized in this context at the time, most of those pioneering studies compared immersive vs non-immersive platforms [27]. The standardized definition of immersive VR has two components: the capacity for the user to interact with the virtual environment, and the incorporation of a head-mounted display [19, 38]. Striving for immersion is important because it is directly correlated to the benefits of using VR; specifically, a user's motivation, long-term retention, and enjoyment of the material [19]. Regardless of these benefits, there are still limitations to the currently available VR educational applications. For example, the novelty effect causes the benefits of VR to decrease in magnitude as users adjust to the new learning environment. Moreover, the cost of generating new VR content has not decreased along with the cost of headsets, which limits the development of VR educational applications [30]. Both limitations might be surpassed with help of Machine Learning methods capable of automatically generating new content for VR applications.

2.2 Development of Spatial Skills

In most cases, spatial skills are learned via the interactions an individual has with objects during their youth [37, 7]. Nonetheless, previous studies have proven that an individual can improve their spatial skill beyond what they developed as a child [8]. The main way that one improves their spatial skill is through the completion of spatial tasks, like the ones present in the Mental Rotation Test or the Purdue Spatial Visualization test [4]. STEM professionals are more likely to have more developed spatial skills than the population in general, mainly because they develop spatial skills when working with spatial tasks like those in STEM fields [2]. Moreover, in STEM programs, both GPA and capacity to conduct self-monitored learning have been shown to relate to a student's spatial skills [33, 28]. This is one of the reasons why it is important to ensure and help students entering STEM programs develop their spatial visualization skills. Knowing that spatial skills can be taught to students via the repetition of spatial tasks, researchers have leveraged VR to help develop students' spatial visualization skills

2.3 VR for Spatial Skills

Using VR as a platform to teach spatial skills is not a novel idea, in fact, several studies have already detailed its effect [38]. The result of these studies indicates that VR is an efficient platform for teaching spatial skills [8]. In a study that compared the improvement of students that used immersive VR and those that used traditional mediums, the students that used VR had greater improvement in their spatial skills [23]. The findings of this work indicate that learning in VR gives the students greater spatial perception than learning on a 2D platform (e.g., a desktop computer) [23].

One of the metrics frequently used to assess and develop spatial visualization skills are the Mental Rotation Test and the Purdue Spatial Visualization test. The application introduced in a previous work by the authors [35], also leverages these tests to help develop students' spatial skills by practicing spatial tasks. For example, Fig.1, from [35] figure 3, shows an example of the mental rotation tasks users are able to perform while using the desktop version of the application. The application introduced in [35], also leveraged hand tracking as a more natural user interface that should increase the user's immersion when compared to using controllers.

While VR has already been shown to help develop spatial skills by allowing students to practice spatial tasks in a 3D virtual environment, it has been also shown that students can have different skill levels that can directly impact their state of flow. The Flow theory of motivation suggests that individuals would be more engaged to execute tasks whose difficulty aligns with their skills level [25].

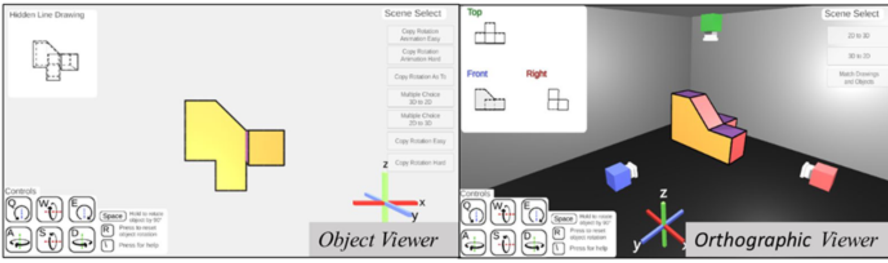


Figure 1: Spatial Visualization Tasks Examples from [17]

Unfortunately, most of the existing VR applications designed to help develop spatial visualization skills only enable users to interact with a limited set of 3D shapes that are not necessarily tailored to their skill level. Nevertheless, previous studies have shown promising results of using Procedural Content Generation and Reinforcement Learning methods to generate new content for VR applications [6, 17, 15].

2.4 Reinforcement Learning

Methods to automatically generate new content, known as Procedural Content Generation (PCG), have been used extensively by the gaming industry [29, 10], and most recently in educational applications as well [11]. In addition, researchers have started leveraging Machine Learning to automatically generate new content. However, methods based on Supervised Machine Learning approaches, require collecting or generating data a priori to train their models[11, 10].

Unlike other forms of Machine Learning, Reinforcement Learning (RL) does not need an established training dataset per se [15]. It instead leverages a simulation environment to generate an action policy to effectively address complex situations [15]. An RL agent uses the current state of a simulation environment and a reward function it tries to maximize. The agent implements a series of “trial-and-error” runs of the simulation environment to generate an action policy that would maximize its long-term reward [1]. Reinforcement Learning agents also have the benefit of generating an action policy that enables them to act in simulated environments with different states without the need for additional training. Studies have shown that Reinforcement Learning agents can effectively generate action policies to perform a task in complicated tasks, such as puzzles, retro video games, and board games like Go [22, 13, 31].

Table 1. shows some of the existing work done to help develop students’ spatial visualization skills, as well as the use of PCG and RL in educational

applications. While several works have explored the use of VR to help develop spatial visualization skills, most of the proposed solutions have several limitations that arise from the challenges of using a finite and limited set of 3D shapes users can interact with. Hence, in this work, the authors introduce a PCG method based on a Reinforcement Learning approach, that would be capable of automatically generating new 3D shapes of a given complexity. This would also enable to tailor of 3D shapes based on students' skill levels in order to improve their state of flow and motivation while interacting with a VR application designed to develop their spatial visualization skills.

Table 1: Summary of existing works

Reference	Solution for SVS	VR for SVS	PCG	PCG with RL
[33, 28, 7]	X			
[8, 5, 3]	X	X		
[10, 11, 12, 37]			X	
[6, 17, 15]*			X	X
This work	X	X	X	X

* *Author's previous work, Spatial Visualization Skills=SVS*

3 Method

This work introduces a Procedural Content Generation (PCG) method based on a Reinforcement Learning (RL) approach that automatically generates 3D shapes of a given desired complexity. The RL agent uses a 3D shape generator environment to create and test different 3D shapes. A metric that measures the symmetry of the shape is used as a complexity metric since previous studies have shown that 3D shapes that are symmetrical are perceived as less complex than shapes that are not symmetrical [36]. The reward function used to train the RL agent considers the difference between the desired complexity and the complexity of the shape generated, as well as other factors to help mitigate the sparsity problems. The RL agent and training process are explained in more detail next.

3.1 3D Shapes Generation

To algorithmically generate 3D shapes, like those present in the Purdue Spatial Visualization test [4], a system of wedges and voxels was implemented. Figure 3 shows a representation of this system. There are a total of 8 voxels arranged in a cube configuration. This means, that the biggest 3D shape that can be generated would have 2 voxels of length on any axis. Moreover, each voxel is composed of 12 different wedges that can be enabled or disabled to generate different 3D shapes. These 12 wedges come from having a wedge (i.e., cube cut in diagonal) in each of the three axes that can be rotated 90 degrees at least four times (i.e., 3*4=12). To have a complete voxel, at least two complementary wedges need to be enabled. A video of the 3D shape generation can be seen here `https://youtu.be/Z9n2zUuqk-E`. This process can generate shapes without curvatures, which are the most predominant shapes in the Purdue Spatial Visualization Test (i.e., 70% of the shapes do not have any curvature).

Based on previous literature, a complexity metric was developed to measure the symmetric shapes generated [38]. Figure 2, shows a visualization of the complexity metric. This metric aims to measure complexity based on the symmetry of 3D shapes. For each shape, a moving cutting plane in each of the x, y, and z axes was used to scan the shape. This sweep allows to potentially capture complexity arising from internal parts of the 3D shape (e.g., if a shape has a whole) and to account for the difference in the shape along an axis (i.e., the front is different than the back). For each of the cutting planes, the voxel area used to generate the 3D shape was identified. This is to help the metric to be size and translation invariant (i.e., a shape should have the same complexity no matter its size or location). Subsequently, the pixel level Euclidean difference between areas of the shape that were divided by each of the 4 potential lines of symmetry was calculated. The difference was then normalized to have values between 0-1.

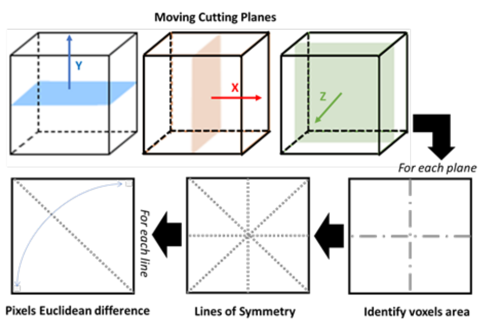


Figure 2: Visualization of Complexity Calculation

A 3D shape with a complexity of 0 would indicate a shape than in all its axes, any cutting plane taken, would have a total of 4 lines of symmetry (i.e., symmetrical horizontally, vertically, and both diagonals). A 3D shape that would have a complexity of 0 would be a cube. Figure 3 shows an example representation of the complexity metric for a cube and a wedge. As shown in Fig. 5, any cutting plane of the cube would have all 4 lines of symmetry, while the wedge would only have 1 (i.e., gray dotted lines). Therefore, the complexity value of the wedge is greater than the cube.

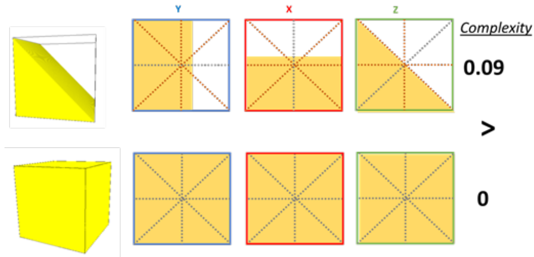


Figure 3: Complexity Calculation Examples

3.2 RL Agent & Reward Function

Since there are a total of 8 voxels each with12 wedges that can be enabled or disabled to create a 3D shape, the size of the possible combinations is equal to 2 to the power of 12 by 8, or $7.9*10^{28}$ (2^{96}) . The possible wedges combinations grow rapidly as more voxels are added. This presents a valuable opportunity to use RL to generate shapes since the number of potential combinations to generate different shapes grows rapidly with the number of voxels used, which makes it intractable to test all possible combinations.

The method introduced in this work uses an RL agent that implements a Neural Network that takes as input a vector $[S, C]$, where S is an 8 by 12 matrix containing binary variables $S_{(i,j)}$, $i \in \{1..8\}$ and $j \in \{1..12\}$ that indicates if a given wedge j of voxel i is enabled or disabled. C is the desired complexity of the 3D shape the agent needs to generate and can range between values of 0 and 1. The network outputs an 8 by 12 matrix of probabilities X, where $X_{(i,j)}$, $i \in \{1..8\}$ and $j \in \{1..12\}$ represent the probability if a given wedge j of a voxel i should be enabled or disabled. This probability vector is then used to update the vector S using a threshold of 0.5. This configuration creates an environment with a multi-discrete action space, which allows the agent to make a shape at each step and make changes to the network weights after each step, based on the reward function.

Equation (1) show the reward function that takes as input the current shape and the desired complexity $[S, C]$, and returns a reward based on how close the complexity of the shape generated is to the desired complexity. Small penalties are detracted from the overall reward for each wedge used, and any additional wedges of a voxel used beyond the complementary wedges. This is to help mitigate the sparsity problem since having any additional wedges enabled in a voxel after two complementary wedges are enabled would not generate a different shape. This is because once two complementary wedges on a voxel are enabled a cube is the only shape that can be generated using that voxel. Moreover, if an empty shape is returned (i.e., no wedges enable), it receives a penalty to deter it from returning no wedges enabled (i.e., no shapes).

$$R(S, C) = \begin{cases} 0 \text{ if } \sum_{i=1}^{8} \sum_{j=1}^{12} S_{i,j} = M \\ 1 - |C - c(S)| - \beta(\frac{\sum_{j=1}^{12} S_{i,j}}{100}) - \theta(\gamma_i) \text{ otherwise} \end{cases} \quad (1)$$

Where:

- S is the matrix containing the shape being passed in.
- $S_{i,j}$ is a binary variable that equals 1 if the wedge j of voxel i is enabled, $i \in \{1..8\}$ and $j \in \{1..12\}$. passed in.
- C is the desired complexity.
- $c(S)$is the complexity of the given shape S
- γ_i counts the number of wedges enable in voxel i beyond any pair of complementary wedges.
- M is a negative number that serves as a penalty.
- β and θ are positive values that serves as weights

During each training epoch, the agent starts by randomly generating a desired complexity C between 0 and 1 from a discrete uniform distribution, and feeds that into the Neural Network with a S matrix that has no wedges enabled (i.e., no shape). The matrix S is “flattened” to a vector of length 96 before passing as input to the network. The agent creates a probability vector for the wedges and generates a new shape as a result, then feeds it back into the Neural Network with C.

4 Results and Discussion

The RL agent was trained on a Windows computer with an Intel© Core™ i7-9750H 2.6 GHz CPU and 32 GB of RAM. The RL agent and Neural Network were implemented in Python using the Keras library.

Some initial hyperparameter tunning was performed in which the number of layers and number of neurons per layer of the RL agent neural network, the β and θ weights of the reward functions, and the optimization algorithms learning rate were explored. From these initial results, an RL agent with a fully dense Neural Network with 10 hidden layers, each having 30 neurons was implemented. The hidden layers used a rectified linear unit activation function while the output layer used a SoftMax activation function. After each training epoch, the loss function for the neural network is calculated from the weights of the network, and gradients are collected until the network is updated. The β and θ weights of the reward functions are set to 0.6. Each epoch the agent will try to find a shape that maximizes the reward (i.e., max value of 1) up to 10 times before the desired complexity changes. The weights of the network are updated after every 10 training episodes, using an Adam optimization algorithm with a learning rate of 0.001.

Figure 4, shows the reward of the RL agent over a range of 3,000 training episodes. On the first 200th training iterations, the RL agent achieved an average reward of -2.56 (Min=-3.650, Max= -1.04, SD=0.54). However, in the last 200th iterations, the RL agent achieved an average reward of 0.67 (Min=0.37, Max= 0.93, SD=0.15). This indicates that the RL agent generated an action policy that significantly improves the reward functions since, on average, the differences in the rewards in the first and last 200th iterations are statistically different ($p-value<0.001$). Similarly, the variance is statistically different ($p-value<0.001$) indicating that the training process allowed the RL agent to generate an action policy that is more consistent at generating 3D shapes that maximize the reward (e.g., with complexity similar to the desired complexity).

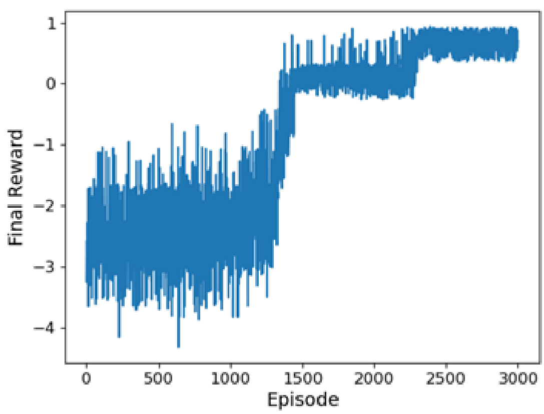


Figure 4: Reward vs Training Episodes

Nevertheless, it is important to highlight that even at the end of the training process the action policy is not able to generate a 3D shape with the same complexity as the desired complexity in 100% of the cases. This could be attributed to the RL agent still not finding the optimal policy and requiring more training or hyperparametric tuning, and because the training environment is randomly picking the desired complexity between 0 and 1 from a discrete uniform distribution with two decimal points of precision. Hence, the training environment might ask the RL agent to generate a shape with a complexity of 0.89 and the latter one with a complexity of 0.9. However, it might not be feasible with the current 3D shape generator systems to create two 3D shapes one with a complexity of 0.89 and another with a complexity of 0.9.

While these results showcase the challenges that raise from the sparse reward space and the current training environment, they also indicate that the RL agent managed to create an action policy that is capable of generating different shapes that have a complexity similar to the given desired complexity. Hence, this work lays the foundation for using RL agents to automatically generate new 3D shapes of the desired complexity. This could potentially be used in VR applications designed to help develop spatial visualization skills.

5 Conclusion and Future Work

Spatial visualization skills are of great importance in STEM fields. While VR has been shown to help develop these skills, most of the existing applications do not necessarily tailor their content to students' skills level. Automatically generating 3D shapes with Procedural Content Generation (PCG) methods, could help VR applications tailor spatial visualization tasks to the skills level of students, as well as generate a wider range of tasks.

This work introduces a PCG method based on a Reinforcement Learning (RL) approach to automatically generate 3D shapes of different complexities. The results indicate that an RL-agent is capable of creating an action policy that can generate 3D shapes with complexities similar to a given desired complexity provided. This could potentially help a VR application design to help develop students' spatial visualization and automatically generated different 3D shapes for the spatial visualization tasks that are in line with students' skill levels. However, the results also show that the task of automatically generating 3D shapes that meet a given complexity is not trivial. Moreover, they show that there are a lot of areas for improvement.

One area that could help improve generalizability and the training of the RL agent, would be the 3D shape generation system. It can be improved by increasing the number of voxels it uses to generate shapes, which will allow a wider range of shapes to be created. In addition, it should be updated so that 3D shapes with curvature are feasible to generate. Lastly, the generation system should be improved to reduce the sparsity of the solution space. With regards to the complexity metric, the authors are already exploring how the complexity metric can be updated to not only help with the sparsity issues but also to better capture individuals' perceived complexity of shapes used in spatial visualization tasks.

Acknowledgement

This research was partially funded by the EXCEL scholar program at Lafayette College. Any opinions, findings, or conclusions found in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[2] Kinnari Atit, David I Miller, Nora S Newcombe, and David H Uttal. Teachers' spatial skills across disciplines and education levels: Exploring nationally representative data. *Archives of Scientific Psychology*, 6(1):130, 2018.

[3] Patrick Carlson, Anicia Peters, Stephen B Gilbert, Judy M Vance, and Andy Luse. Virtual training: Learning transfer of assembly tasks. *IEEE transactions on visualization and computer graphics*, 21(6):770–782, 2015.

[4] Lynn A Cooper. Mental rotation of random two-dimensional shapes. *Cognitive psychology*, 7(1):20–43, 1975.

[5] Matthew Coxon, Nathan Kelly, and Sarah Page. Individual differences in virtual reality: Are spatial presence and spatial ability linked? *Virtual Reality*, 20(4):203–212, 2016.

[6] James Cunningham, Christian Lopez, Omar Ashour, and Conrad S Tucker. Multi-context generation in virtual reality environments using deep reinforcement learning. In *International Design Eng. Technical Conf. and Comp. and Information in Eng. Conf.*, volume 83983, page V009T09A072. American Society of Mechanical Engineers, 2020.

[7] Zachary S Gold, James Elicker, Ashleigh M Kellerman, Sharon Christ, Aura A Mishra, and Nina Howe. Engineering play, mathematics, and spatial skills in children with and without disabilities. *Early Education and Development*, 32(1):49–65, 2021.

[8] Tibor Guzsvinecz, Éva Orbán-Mihálykó, Erika Perge, and Cecilia Sik-Lányi. Analyzing the spatial skills of university students with a virtual reality application using a desktop display and the gear vr. *Acta Polytechnica Hungarica*, 17(2):35–56, 2020.

[9] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1):1–22, 2013.

[10] Danial Hooshyar, Moslem Yousefi, and Heuiseok Lim. A procedural content generation-based framework for educational games: Toward a tailored data-driven game for developing early english reading skills. *Journal of Educational Computing Research*, 56(2):293–310, 2018.

[11] Danial Hooshyar, Moslem Yousefi, Minhong Wang, and Heuiseok Lim. A data-driven procedural-content-generation approach for educational games. *Journal of Comp. Assisted Learning*, 34(6):731–739, 2018.

[12] Britton Horn, Steve Dahlskog, Noor Shaker, Gillian Smith, and Julian Togelius. A comparative evaluation of procedural level generators in the mario ai framework. In *Foundations of Digital Games 2014, Ft. Lauderdale, Florida, USA (2014)*, pages 1–8. Society for the Advancement of the Science of Digital Games, 2014.

[13] Ionel-Alexandru Hosu and Traian Rebedea. Playing atari games with deep reinforcement learning and human checkpoint replay. *arXiv preprint arXiv:1607.05077*, 2016.

[14] Wen Huang. *Investigating the novelty effect in virtual reality on stem learning*. PhD thesis, Arizona State University, 2020.

[15] Christian E Lopez, Omar Ashour, and Conrad S Tucker. Reinforcement learning content generation for virtual reality applications. In *International Design Eng. Technical Conf. and Comp. and Information in Eng. Conf.*, volume 59179, page V001T02A009. American Society of Mechanical Engineers, 2019.

[16] Christian E Lopez, Omar M Ashour, and Conrad Tucker. An introduction to the click approach: Leveraging virtual reality to integrate the industrial engineering curriculum. In *ASEE annual Conf. & exposition*, 2019.

[17] Christian E López, James Cunningham, Omar Ashour, and Conrad S Tucker. Deep reinforcement learning for procedural content generation of 3d virtual environments. *Journal of Computing and Information Science in Eng.*, 20(5), 2020.

[18] Yukiko Maeda, So Yoon Yoon, Gyenam Kim-Kang, and PK Imbrie. Psychometric properties of the revised psvt: R for measuring first year engineering students' spatial ability. *International Journal of Eng. Education*, 29(3):763–776, 2013.

[19] Guido Makransky, Stefan Borre-Gude, and Richard E Mayer. Motivational and cognitive benefits of training in immersive virtual reality based on multiple assessments. *Journal of Comp. Assisted Learning*, 35(6):691–707, 2019.

[20] Guido Makransky and Lau Lilleholt. A structural equation modeling investigation of the emotional value of immersive virtual reality in education. *Educational Technology Research and Development*, 66(5):1141–1164, 2018.

[21] Tassos A Mikropoulos and Antonis Natsis. Educational virtual environments: A ten-year review of empirical research (1999–2009). *Computers & Education*, 56(3):769–780, 2011.

[22] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[23] Rafael Molina-Carmona, María Luisa Pertegal-Felices, Antonio Jimeno-Morenilla, and Higinio Mora-Mora. Virtual reality learning activities for multimedia students to enhance spatial ability. *Sustainability*, 10(4):1074, 2018.

[24] Takudzwa Mujuru and Christian Lopez. Creating virtual reality teaching modules for low-cost headsets. In *International Design Eng. Technical Conf. and Comp. and Information in Eng. Conf.*, volume 85376, page V002T02A083. American Society of Mechanical Engineers, 2021.

[25] Beatrice Ottiger, Erwin Van Wegen, Katja Keller, Tobias Nef, Thomas Nyffeler, Gert Kwakkel, and Tim Vanbellingen. Getting into a "flow" state: a systematic review of flow experience in neurological diseases. *Journal of neuroengineering and rehabilitation*, 18(1):1–21, 2021.

[26] James W Pellegrino, David L Alderton, and Valerie J Shute. Understanding spatial ability. *Educational psychologist*, 19(4):239–253, 1984.

[27] Jaziar Radianti, Tim A Majchrzak, Jennifer Fromm, and Isabell Wohlgenannt. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147:103778, 2020.

[28] Cristina Roca-González, Jorge Martín Gutiérrez, Melchor García-Dominguez, and María del Carmen Mato Carrodeguas. Virtual technologies to develop visual-spatial ability in engineering students. *Eurasia Journal of Mathematics, Science and Technology Education*, 2017.

[29] Luiz Rodrigues, Robson Parmezan Bonidia, and Jacques Duílio Brancher. A math educacional computer game using procedural content generation. In *Brazilian Symposium on Comp. in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 28, page 756, 2017.

[30] Vicente Román-Ibáñez, Francisco A Pujol-López, Higinio Mora-Mora, Maria Luisa Pertegal-Felices, and Antonio Jimeno-Morenilla. A low-cost immersive virtual reality system for teaching robotic manipulators programming. *Sustainability*, 10(4):1102, 2018.

[31] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[32] Sheryl Sorby. A new and improved course for developing spatial visualization skills. In *2001 Annual Conf.*, pages 6–66, 2001.

[33] Sheryl Sorby, Norma Veurink, and Scott Streiner. Does spatial skills instruction improve stem outcomes? the answer is 'yes'. *Learning and Individual Differences*, 67:209–222, 2018.

[34] Sheryl A Sorby, Gavin Duffy, Norman Loney, Lance Perez, et al. Spatial skills and their correlation with engineering problem-solving. In *29th Australasian Association for Eng. Education Conf.*, volume 2018, page 10, 2018.

[35] Liam Stewart and Christian Lopez. Developing spatial visualization skills with virtual reality and hand tracking. In *International Conf. on HCI*, pages 390–398. Springer, 2021.

[36] Sreenivas R Sukumar, David L Page, Andreas F Koschan, and Mongi A Abidi. Towards understanding what makes 3d objects appear simple or complex. In *2008 IEEE Comp. Society Conf. on Comp. Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.

[37] Brian N Verdine, Roberta Michnick Golinkoff, Kathy Hirsh-Pasek, and Nora S Newcombe. I. spatial skills, their development, and their links to mathematics. *Monographs of the society for research in child development*, 82(1):7–30, 2017.

[38] Diego Vergara, Manuel Pablo Rubio, and Miguel Lorenzo. On the design of virtual reality learning environments in engineering. *Multimodal technologies and interaction*, 1(2):11, 2017.

Using Information Structure Design To Reduce Bribery*

Timothy Highley and Kidanewold Demesie
Department of Mathematics and Computer Science
La Salle University
Philadelphia, PA 19002

Abstract

This paper investigates the use of information structure design to reduce corruption involving bribery. In information structure design (also called signaling or persuasion), there is a principal agent that has access to private information. The principal agent can choose to release that information to other agents in order to meet a desired objective. This paper introduces a model where, in addition to the principal agent, there are two sets of agents: officials and clients. All agents are Bayesian and strategic. The officials provide a service to the clients. Clients may offer bribes, and agents may solicit bribes. Transactions may be monitored, but there are not sufficient resources to review every transaction. It is shown that in some circumstances the principles of information structure design can be used in conjunction with transaction monitoring to reduce the number of bribes more than with monitoring alone. In particular, truthful signaling should be used whenever the number of transactions that can be monitored is below a threshold value. Signaling should not be used if the number of transactions that can be monitored is above the threshold value. For untruthful signaling, an approach is presented that minimizes the number of attempted bribes.

1 Introduction

In some contexts, bribery is pervasive [7]. Game theory applies in the unfortunate reality of contexts where bribery is pervasive: if everyone else participates in bribery, it is to each individual's advantage to also participate in bribery [6]. In other words, there are two stable equilibria: the clean equilibrium where no one participates in bribery, and the corrupt equilibrium where everyone participates in bribery. (Naturally, in the real world neither equilibrium is as stable as the theoretical model would indicate. There are instances of bribery in contexts where bribery is rare, and there are interactions without bribery in contexts where bribery is pervasive.)

This paper considers a model of bribery involving clients and officials. Some examples of interactions between clients and officials include a police officer interacting with a citizen during a traffic stop, and a teller at a government agency providing a service to a customer, such as handling a visa application.

In game theory, the area of signaling is also known as information structure design. Signaling has been studied in a number of contexts, such as maximizing revenue [5] or in elections [1]. In information structure design, one particular agent is designated as the principal agent, and that agent has access to private information [3]. Each agent is Bayesian and strategic, and pursues its own goals. The principal agent may strategically give signals to reveal private information to other agents. Under private signaling, the principal agent may give signals to other agents individually. However, under public signaling, when the principal agent gives a signal, all other agents see that signal [2] [4]. This paper deals with private signaling in the context of reducing bribery.

2 Model

2.1 No-Observation Model

Consider a system where officials provide services to clients. Officials may choose to be clean or corrupt, and clients may choose to be clean or corrupt. Corrupt officials expect and/or solicit bribes, and if they do not receive a bribe they punish the client with bad service. Bad service may include unnecessary delays in service, refusal to answer questions, or overcharging the client for unnecessary services. Bad service is depicted by a parameter $\beta \geq 0$, where a larger value of β indicates worse service. For comparison purposes, assume that β is expressed as a monetary value that quantifies the extent to which the client is inconvenienced. Corrupt clients may offer bribes to avoid receiving bad service. Each encounter between an official and client is represented by a game, where the client and official each independently choose whether to be clean or corrupt. If a corrupt agent (either official or client) encounters a clean agent, the corrupt agent is caught and faces a penalty in the form of fine amounting to f. Assume that when bribes are offered they are in a standard and constant amount: d. We will assume at first that no transactions between officials and clients are monitored, so we call this the No-Observation Model, and it has the payout matrix depicted in Table 1.

Table 1: Payout Matrix for No-Observation Model

Official	Client	
	Clean	Corrupt
Clean	(0, 0)	$(0,-f)$
Corrupt	$(-f, -\beta)$	$(d, -d)$

Let a be the probability that a client encounters a corrupt official. That makes the payout for a clean client $(a)(-\beta)$. The payout for a corrupt client is $(a)(-d) + (1 - a)(-f)$. Let b be the probability that an official encounters a corrupt client. That makes the payout for a clean official 0, and the payout for a corrupt official $(b)(d) + (1 - b)(-f)$. In this model, if $\beta \leq d$, then the client has no incentive to be corrupt, and therefore the official has no incentive to be corrupt. Therefore, assume that $\beta > d$. That is, the bad service is worse for the client than the cost of paying the bribe. In this situation, a client or official wants to match what the other agent does. Therefore, the best strategy is to be corrupt if the other agent is corrupt, or to be clean if the other agent is clean. Over time, populations will tend toward one of two stable equilibria: either $a = 0$ and $b = 0$, or $a = 1$ and $b = 1$. If $a = 0$ and $b = 0$ then the desired result is already attained, so going forward we may as well assume the corrupt equilibrium state, $a = 1$ and $b = 1$.

2.2 Observation Model

For the Observation Model, we begin with the No-Observation model but assume that there are cameras present at all transactions. However, there are not enough resources to actually monitor all of the transactions. Assume that there are enough resources available to monitor a proportion of the transactions given by n, where $0 \leq n \leq 1$. The value of n is known by all of the officials. Any watched bribe will be caught 100% of the time, but not until after the client has experienced the bad service imposed by a corrupt official. When a corrupt transaction is caught, the bribe is returned to the client, and both the client and official suffer a penalty.

This changes the payouts as follows: Clean clients still suffer bad service when encountering corrupt officials, so their payout doesn't change: $(a)(-\beta)$. Corrupt clients will suffer the penalty for getting caught when monitored.

When not monitored, their payout is the same as the No-Observation model. The corrupt client's payout is then:

$$(1-n)[(a)(-d)+(1-a)(-f)]+(n)[-f].$$

A clean official never faces a penalty and never gains the benefit of a bribe, so the clean official's payout is still always 0. A corrupt official will always suffer the penalty of being caught if the corrupt action is monitored. If the transaction is not monitored, it is the same as the payout for a corrupt official in the No-Observation model. The corrupt official's payout is then:

$$(1-n)[(b)(d)+(1-b)(-f)]+(n)[-f].$$

3 Signaling Strategies

3.1 No Signaling vs. Truthful Signaling

We will now assume that there is a signal light on the camera that can be set to either on or off, and we consider how that signal can be used to reduce bribery. The principal agent has an option to use the on/off light on the camera to signal as follows: The light being on is meant to indicate that the transaction is being observed, while the light being off is meant to indicate that the transaction is not being observed. How should the principal agent use the light to reduce the number of corrupt transactions? Initially, we consider only the option of truthful signaling: the light is on when the transaction is observed and off when it is not observed. Is truthful signaling better than not signaling at all?

Table 2 shows the payout matrix when the light is on under truthful signaling. Corrupt agents are always caught and pay the penalty.

Table 2: Observation Model When Observed (Truthful "On" Signal)

Official	Client	
	Clean	Corrupt
Clean	(0, 0)	$(0,-f)$
Corrupt	$(-f,-\beta)$	$(-f,-f)$

The payout matrix when the light is off exactly matches the payout matrix for the No-Observation model (Table 1). Under truthful signaling, if it is assumed that all agents tend toward corruption, then when the light is on, the dominant strategy for all agents is to be clean, and when the light is off, the dominant strategy for all agents is to be corrupt.

As noted earlier, if no signals are given at all, then clients and officials will assume that for every transaction the probability that they are being monitored is equal to n. The payout matrix for the Observation Model with no signaling is given in Table 3.

Table 3: Observation Model without Signaling

Official	Client	
	Clean	Corrupt
Clean	(0, 0)	$(0,-f)$
Corrupt	$(-f, -\beta)$	$((1-n)d + n(-f), (1-n)(-d) + n(-f))$

With no signals, if n is vanishingly small, the cameras will have no impact on behavior. If n is very large, everyone will be clean because of the cameras. Therefore, the principal agent should choose no signaling if n is sufficiently large. If n is not very large, then in order to minimize the number of corrupt transactions, truthful signaling should be used, because it will at least persuade the agents to be clean in those transactions that are monitored. What is the cutoff where the principal agent should stop signaling?

Even if all officials are corrupt ($a = 1$), a client will choose to be clean if $(-\beta) > [(1-n)(-d) + n(-f)] = -d + n(d-f)$. Assuming $f > d$, the client will choose to be clean if $n > (\beta - d)/(f - d)$.

Even if all clients are corrupt ($b = 1$), an official will choose to be clean if $0 > (1-n)(d) + (n)(-f)$. That is, if $n > d/(d+f)$.

If either the clients or officials choose to be clean, the other group will follow suit, so it is better to use no signaling instead of truthful signaling whenever $n > min((\beta - d)/(f - d), d/(d + f))$. Since the term β refers to how much a client is inconvenienced by bad service, and this is difficult to quantify and will vary from one client to the next, it is safer and easier to use no signaling whenever $n > d/(d + f)$. This value can be computed exactly if d, the dollar-value of a typical bribe, is known and f is also known. (If the penalty were being fired or jailtime, then determining the appropriate value for f would be more difficult to quantify and would vary from person to person.) If n is less than $d/(d+f)$, it is better to use truthful signaling so that at least the observed interactions will be clean.

As an example, assume there is a police force where bribes are often taken during traffic stops. The police force has added body cameras of the type indicated earlier, but there are not enough resources to store and/or watch all of the recorded interactions. As described previously, the body cameras have a light that can indicate when an interaction is being monitored. Suppose that initially, there are few resources available to monitor transactions, so n is very low. As more resources are added and more interactions can be effectively monitored, n will rise. Suppose an officer gets \$50 per bribe, but if caught taking a bribe the penalty is \$500. In that case, signaling should be turned off once the police force can monitor at least (50/(50+500)), or about 9%, of the interactions.

3.2 Untruthful Signaling

The previous analysis limited the choices to truthful signaling or no signaling. Note that under truthful signaling, all signaled transactions are monitored and are therefore clean, since the agents have no incentive to be corrupt when they know they will be caught. Can the principal agent send an "on" signal and cause the transaction to be clean even when it is not being monitored? We will show that the answer is yes, as long as the principal agent does not send so many "on" signals that credibility is lost.

Based on the above analysis, we start with the result that officials will be clean if they believe the probability that they are being monitored is greater than $d/(d+f)$, and will be corrupt otherwise. In all of those signaled transactions, the officials will choose to be clean as long as their Bayesian reasoning leads them to conclude that the probability that they are being watched is at least $d/(d+f)$. Therefore, if $n < d/(d+f)$, we can signal the n portion of transactions that are actually being observed and also untruthfully signal some additional transactions.

Let $P(X)$ denote the probability that event X occurs. Let $P(X|Y)$ denote the conditional probability that event X occurs, given that event Y occurs. If W is the event that an official is being watched, and S is the event that the official receives an "on" signal, then

$$P(W|S) > d/(d+f).$$

For now, assume that all monitored transactions are signaled, so $P(S|W) = 1$. Based on that fact and Bayes' Theorem,

$$P(W|S) = P(S|W)P(W)/P(S) = P(S|W)(n)/P(S) = n/P(S). \qquad (1)$$

We want $n/P(S) > d/(d+f)$. Equivalently, $P(S) < n + (nf/d)$. Thus, the maximum proportion of time that the principal agent can send "on" signals without losing credibility is $n + (nf/d)$.

In Formula 1, if $P(S|W) < 1$, then the maximum permissible value of $P(S)$ would be reduced, resulting in fewer "on" signals being sent. The next section shows that agents will only be clean for both "on" and "off" signals if n is high enough that no signaling strategy is needed. Therefore, the assumption that $P(S|W) = 1$ is safe, since failing to signal a monitored transaction would only result in more corrupt transactions.

Consider the example of the police force where officers routinely accept \$50 bribes and face a \$500 fine if caught. Assume only 2% of the interactions are observed (that is, $n = .02$). The principal agent can signal up to $n + (nf/d)$, or 22%, of the interactions and still be believed by the officials. This results in 22% of the interactions being clean instead of only 2%.

3.3 Limit of the Effectiveness of Signaling

We have shown that all officials will choose to be clean when $n > d/(d+f)$, even with no signaling. The following shows that there is no alternative approach that will convince all officials to be clean when $n < d/(d+f)$.

Again, let S be the event that the official is given an "on" signal and, in addition, let R be the event that the official is given an "off" signal. $P(S) + P(R) = 1$ since the light will be either on or off. Based on prior reasoning, officials will choose to be clean as long as they believe that the probability they will be caught is at least $d/(d+f)$. In order to convince officials to be clean all the time, we need the following two inequalities to be true simultaneously:

$$\begin{aligned} P(W|S) &> d/(d+f), \\ P(W|R) &> d/(d+f). \end{aligned}$$

Bayes' Theorem leads to this equality:

$$\begin{aligned} P(W|S) &= P(S|W)P(W)/P(S) \\ &= nP(S|W)/P(S). \end{aligned} \tag{2}$$

Coupled with the first inequality, we obtain an upper limit on how often an "on" signal can be used.

$$\begin{aligned} P(W|S) &> d/(d+f) \\ nP(S|W)/P(S) &> d/(d+f) \\ P(S) &< nP(S|W)(d+f)/d. \end{aligned} \tag{3}$$

By the same logic, $P(R) < nP(R|W)(d+f)/d$. Suppose those two inequalities are true simultaneously. Adding them, we find:

$$\begin{aligned} P(R) + P(S) &< [nP(S|W)(d+f)/d] + [nP(R|W)(d+f)/d] \\ 1 &< (n(d+f)[P(S|W) + P(R|W)])/d \\ 1 &< n(d+f)/d. \end{aligned} \tag{4}$$

Solving for n, we find $n > d/(d+f)$. No signaling algorithm will result in all clean transactions unless $n > d/(d+f)$, and if $n > d/(d+f)$, then using no signaling is sufficient. In other words, signaling algorithms can reduce the number of corrupt transactions but not eliminate them entirely.

3.4 Ineffective Observers

The previous analysis assumes that corrupt transactions that are monitored are always caught and punished. However, due to the corruption or laziness of the observers, or the cleverness of the agents, some corrupt transactions would succeed even if they were monitored. We now consider this possibility.

From the perspective of the clients and officers, everything is the same as in the previous analysis, except that the meaning of n is changed from "the probability that the transaction is monitored" to "the probability that the transaction will be caught (if corrupt)."

The problem changes more significantly for the principal agent. The principal agent cannot choose a strategy that signals "on" only when the transaction will be caught, because the principal agent does not know which corrupt transactions might succeed. The principal agent only knows which transactions are being observed. However, assume the principal agent does know what percentage of monitored and corrupt transactions are ignored or missed by the observers. Even in this context, the principal agent must maintain the credibility of the "on" signals.

Let W be the event that a given transaction is monitored, and C be the event that a transaction is monitored in such a way that any corruption would be caught. In the previous analysis, it was assumed that $P(W) = P(C)$, but now $P(W) \geq P(C)$. Based on Formula 4, in order to ensure that officials choose to be clean, the principal agent must ensure that

$$P(C|S) > d/(d+f) \tag{5}$$

We know that $P(C) = P(W)P(C|W) = nP(C|W)$ and that $P(S|C) = 1$ (since every monitored transaction is signaled). Combining those facts with Bayes' Theorem, we have

$$\begin{aligned} P(C|S) &= P(S|C)P(C)/P(S) \\ &= P(S|C)nP(C|W)/P(S) \\ &= nP(C|W)/P(S). \end{aligned} \tag{6}$$

From Forumla 5 and Formula 6, we derive an upper limit on how often the principal may give an "on" signal and still cause officials to choose to be clean.

$$\begin{aligned} P(C|S) &> d/(d+f) \\ nP(C|W)/P(S) &> d/(d+f) \\ P(S) &< [nP(C|W)(d+f)]/d. \end{aligned} \tag{7}$$

If the principal agent signals "on" over and above the probability n of transactions that are observed, but not above the limit given in Formula 7, then the officials will choose to be clean. Above that limit, the "on" light will not provide a sufficient deterrent, and the officials will continue in their corruption equilibrium.

As an example, consider the same situation as before: a police force where officers regularly take \$50 bribes, and the penalty for being caught is \$500. Assume again that the proportion of interactions that are monitored is $n = .02$, and add a modification that only 50% of the watched and corrupt interactions are caught. How often should the principal agent send an "on" signal?

$$\begin{aligned} P(S) &< [nP(C|W)(d+f)]/d \\ P(S) &< [.02 * .50 * 550]/50 \qquad = .11 \end{aligned} \tag{8}$$

In this example, the officials will choose to be corrupt if the "on" signal is used more than 11% of the time. The principal agent is still able to signal more than five times as many interactions as are actually monitored, and all of these will be clean instead of corrupt.

3.5 Alternate Metric: Maximize Number of Agents Caught

Throughout this paper, the goal has been to minimize the number of corrupt transactions. A corrupt transaction that is watched and caught still counts as a corrupt transaction. An alternative objective could be to maximize the number of corrupt transactions that are caught. This may make sense if the idea is to reduce corruption by removing the corrupt agents from their positions. On the other hand, if bribery is pervasive, then that approach could lead to simply removing anyone at random.

If the goal is to maximize the number of corrupt transactions that are caught, then the strategy is simple: do not signal at all. Signaling indicates to the corrupt agents when they should not be corrupt, so it would reduce the number of corrupt transactions and reduce the number of corrupt transactions that are caught.

4 Conclusion and Future Work

We have shown that in some circumstances monitoring plus signaling can be used to reduce the number of corrupt transactions more than monitoring alone. In the model presented in this paper, truthful signaling can reduce bribery more than monitoring without signaling if the fraction, n, of transactions that can be monitored is less than $d/(d+f)$, where d is the standard amount of a bribe and f is the amount of the fine if caught. If untruthful signaling is used, then the principal agent can signal additional transactions over and above the monitored transactions, as long as the total number of signaled transactions does not exceed $n + (nf/d)$. If, due to ineffective observers, some corrupt transactions succeed even when being monitored, and $0 \leq P(C|W) < 1$ represents the probability that a corrupt transaction is caught when it is being watched, then the principal agent can signal additional transactions over and above the monitored transactions, as long as the total probability of a signaled transaction does not exceed $nP(C|W)(d + f)/d$.

We considered two different criteria for reducing corruption: reducing the number of corrupt transactions and maximizing the number of corrupt transactions that are caught. As future work, a third criterion could be considered: minimizing the number of corrupt transactions that are not caught. Also, all three of these could be explored in a more realistic model. The models that we have analyzed included simplifications. These models assumed a uniform population, but in the real world, individuals will have different tendencies toward bribery and different risk tolerances for getting caught.

References

[1] Matteo Castiglioni, Andrea Celli, and Nicola Gatti. Persuading voters: It's easy to whisper, it's hard to speak loud. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1870–1877, 2020.

[2] Matteo Castiglioni, Andrea Celli, and Nicola Gatti. Public Bayesian persuasion: Being almost optimal and almost persuasive. *arXiv preprint arXiv:2002.05156*, 2020.

[3] Shaddin Dughmi. Algorithmic information structure design: A survey. *SIGecom Exch.*, 15(2):2–24, Feb. 2017.

[4] Shaddin Dughmi. On the hardness of designing public signals. *Games and Economic Behavior*, 118:609–625, 2019.

[5] David Lingenbrink and Krishnamurthy Iyer. Optimal signaling mechanisms in unobservable queues. *Operations research*, 67(5):1397–1416, 2019.

[6] Philip M. Nichols. Corruption as an assurance problem. *American University International Law Review*, 19:1307, 2003.

[7] Gedion Onyango. The art of bribery: Analysis of police corruption at traffic checkpoints and roadblocks in Kenya. *International Review of Sociology*, 2022.

An Early Measure of Women-Focused Initiatives in Gender-Imbalanced Computing Programs*

Mary Villani[1], Ilknur Aydin[1], Lisa Cullington[2]
[1]Computer Systems
[2]RAM (Research Aligned Mentorship) Program
Farmingdale State College, SUNY
Farmingdale, NY 11735, USA
{villanmv,aydini,cullinlm}@farmingdale.edu

Abstract

Gender disparity in student enrollment is consistent with many other national and global computer degree programs, however, as student enrollment at Farmingdale State College (FSC) computing degrees has doubled, the percentage of women remained hovering at 8-15% over the last decade. Recently, initiatives have been underway to address the gender imbalance that exists at FSC computing degree programs. With a limited budget, and some local funding, multiple initiatives to reduce the gender disparity have been taken recently. This paper describes the details of the initiatives in progress and presents early survey results to report a measure on how students identifying as women are responding to these activities. In addition, early survey results provide promising preliminary read on the experience of women while pursuing their computing degrees and indicate how to focus the future initiatives.

1 Introduction

Farmingdale State College (FSC) is a four-year college within the large state college system of SUNY (State University of New York) which offers affordable tuition and location in densely populated areas resulting in a primarily commuter campus. With a student population of over 10,000, only 10% reside on campus and 79% work at off-campus jobs. Computing degrees student enrollment at FSC is dominated by male students which is consistent with many other national and global computing programs according to National Center of Education Statistics (`https://nces.ed.gov`). Although women make up most college graduates, they earned only 19% of the computer science degrees awarded in 2016. Even though enrollment in FSC computing degree programs has doubled over the past twelve years, reaching over 600 at its peak, women enrollment has hovered between 8-15% as seen in Figure 1. FSC has offered Computer Programming and Information Systems (CPIS) for the past twenty years and introduced a Computer Science (CS) program in fall 2021. Given this persistent gender disparity in enrollment, a strategic vision was developed to address the issue as follows.

1. achieve 50% women enrollment in the program as inspired by the Anita Borg's 50/50 vision [4].
2. be the college of choice regionally and nationally for students due to the focus on Women and Diversity in Computing.
3. have FSC CPIS and CS graduates be candidates of choice by the computing technology community.

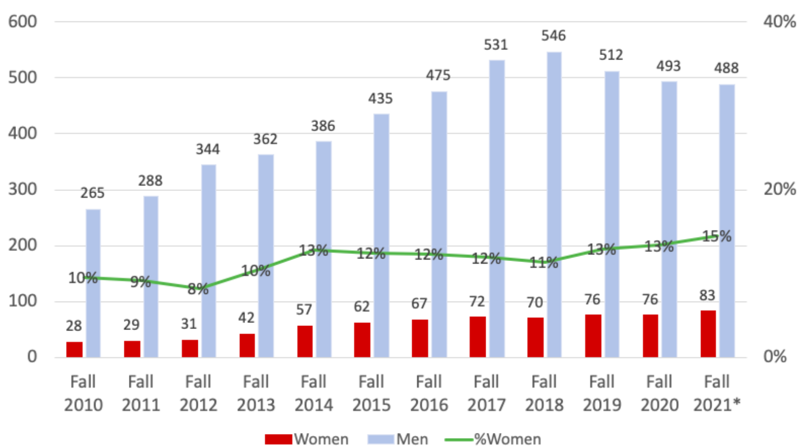


Figure 1: Enrollment of CPIS degree (*fall 2021 combined with CS)

An extensive list of tactical initiatives has been identified to achieve the strategic vision. However, with limited resources and funding, three initiatives were selected for implementation, with the goal of improving women student experience in the short term and women student recruitment and retention in the long term. The purpose of this paper is to document the preliminary outcomes of implementing these initiatives. Section 2 explains each initiative underway. Section 3 provides relevant literature to support programming decisions. Section 4 presents the end of the semester survey design, results, and analysis. Section 5 further discusses implications for the initiatives given the early survey results. Finally, Section 6 concludes the paper with future research agenda.

2 Methodology: Initiatives Underway

The three initiatives include the following and are explained below 1) restarting and maintaining an active Women in Computing club 2) designing and implementing a summer orientation program specifically for women computing students, and 3) planning field trips to women-centric conferences (Figure 2). Recent initiatives since spring 2020 has been led by two, tenured women faculty. While one of the faculty has been teaching first and second year computing and programming courses, the other faculty has taught the capstone senior project class for over a decade. Hence, between the two faculty, they have a direct connection with most of the undergraduate women student population. Note that, there were earlier but non-lasting efforts during 2009-2013. History, details, and analysis of those efforts were presented earlier and has fueled the work and efforts for the current initiatives [13, 15].

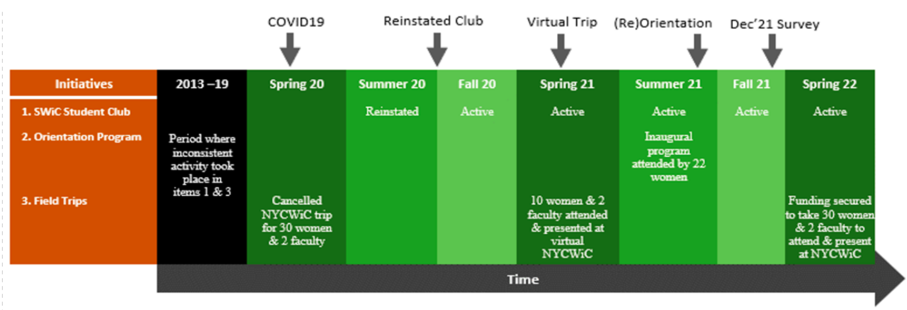


Figure 2: Chronology of women-centric initiatives taken at FSC

2.1 Supporting Women in Computing Club (SWiC)

In FSC, the **Supporting Women in Computing (SWiC)** club was reinstated amid the pandemic in fall 2020. Remote learning caused by the pandemic had a positive impact on club membership such that students seemed eager for connection while SWiC virtual meetings and events filled this gap. The lessons learned from operating remotely during fall 2020 and spring 2021 was applied to create an active calendar, full of hybrid events in fall 2021 [14]. SWiC hosted guest speakers from afar while maintaining active chats and strong in-person and virtual member attendance.

2.2 Summer (Re)Orientation Program for Women Students

In FSC, all women (n=79) enrolled in computing programs at FSC (including first year first time transfer and returning) were invited to attend a one-day inaugural (re)orientation program in August 2021. 32 women students RSVP'd and 22 attended the (re)orientation. It was a productive day including a welcome presentation, icebreaker activities, two technical presentations by other teaching faculty, tours of facilities, and a team building activity ending with raffles and prizes. Findings from pre- and post-orientation student surveys were overwhelmingly positive where 90% of attendees expressed that attending the orientation was time well spent and that they learned a great deal. 80% of the attendees expressed they were excited and motivated to attend classes after attending the incoming student orientation. Upon learning of the gender gap in the enrolled student population in computing programs at FSC, 75% of the students expressed that they wanted to succeed in computing classes to defy the gender gap. About 60% of attendees said they would get involved in the SWiC club because of attending the orientation. Attendees also expressed that they wished that it had been longer in duration and suggested additional ice breaker and team-building activities. The design and implementation details of this 2021 summer (re)orientation and its effectiveness were reported in [2].

2.3 Field Trips to Women-Centric Event and Conferences

In FSC, a field trip to attend ACM New York Celebration of Women in Computing (NYCWiC) Conference in Poughkeepsie, NY in April 2020 was cancelled due to COVID19. The disappointment of not attending is indeed what inspired a few women students to restart the SWiC club in fall 2020. The officers of the reinstated SWiC submitted a diversity poster in the following year that was accepted to present at the virtual NYCWiC'21. Faculty advisors also moderated a Birds of a Feather Session. Virtual attendance to NYCWiC was at a deeply reduced rate, therefore, club funds were used to allow SWiC club members interested to attend the virtual conference. Recognizing that attending virtually is not same as attending in person, still 10 students were able to enjoy the benefits and have the experience of attending a women-centric computing conference.

3 Literature Review

In recent years, K-12 school districts and nonprofits have increased the number of after-school student clubs focused on increasing awareness and interest in computing majors and future careers for girls [10, 16]. Heo et. al [6] found that by participating in an after-school computing club, girls' knowledge and confidence in computing skills grew. Most efforts on understanding the retention of women in STEM at the undergraduate level has focused on increasing the number of women faculty in the department, providing individual tutoring and peer mentoring options, interactive engagement during class and residential learning communities [3]. Less is known about the impact of these informal student clubs on undergraduate students' retention in computing majors and earning a baccalaureate degree at commuter colleges.

Orientation and summer bridge programs have provided significant benefits for historically underrepresented groups in higher education, such as racial and ethnic groups, first generation college students, and students who are classified as low-income [7, 8, 5]. Tomasko et. al found that summer bridge programs increased students' persistence in addressing students' sense of belonging, introducing students to the social life of college, and increasing their ties to the institution [12]. Summer bridge programs have the power to increase students' retention at an institution, but more specifically, retention in STEM majors. Howard and Sharpe conducted a study [7] on the effectiveness of the summer bridge program as part of the Tennessee Louis Stokes Alliance for Minority Partnership (LSAMP) program. LSAMP is a support and access program funded through the National Science Foundation with a goal of increasing the number of students successfully earning STEM baccalaureate degrees. They found that the LSAMP summer bridge model was effective at increasing retention of students at the institution after the first year, retaining the students in their major, and increasing their grade point average in foundational courses such as English and Mathematics. Despite these promising studies, limited research details how these programs, specifically shorter, one-day programs, can be adapted to support women enrolled in computing-related degree programs at commuter colleges.

Many professional and research associations, such as the Anita Borg Institute, ACM, and regional chapters of Grace Hopper, host diversity conferences for students to attend and to present their undergraduate research. Many of these conferences are specifically designed to support women in the computing field. The CRA CERP found that underrepresented students who attended these diversity conferences were more confident in completing their undergraduate degree [11]. Klawe in [9] identified three strategies implemented at Harvey Mudd College (HMC) to increase the number of women in the computer science major. These strategies included modifying the introductory computer science course, providing summer research opportunities, and increasing student attendance at the Grace Hopper Conference (GHC). HMC brought first year women in any major to the annual GHC and increased their women enrollment in computer science from 12% to 40% from 2005 to 2012. Alvarado et. al [1] found that students from HMC who attended GHC were more likely to take additional computer science courses if they were not previously considering the major, and those who were, grew to be more confident in their ability to pursue the major.

4 Results and Analysis

Short term impact of the three initiatives (Figure 2 and Section 2) on women student experience was measured with a survey given to all women computing students of FSC at the end of fall 2021. Out of the 69 women students who received survey via email, 19 responded, yielding a response rate of 27.5%. Average email survey response rate at FSC is 30%, so the survey response rate was consistent with larger institutional survey administrations.

In terms of the demographic of the survey respondents, over 40% (n=8/19) were incoming (first-year students or transfer) while 60% were returning students. This is consistent with the summer'21 (re)orientation attendees' breakdown such that the (re)orientation program taken interest by returning women as well. Most of the incoming first-time students (n=4/5) graduated high school within 6 months. None of the incoming transfers held a bachelor's degree but transferred more than 61 credits (Note that 120+ credits is needed to complete a bachelor's degree at FSC). Majority of the returning student

Table 1: Survey Results of Women Computing Student Experience at FSC

	Academic		**Social**	
	percent	*count*	*percent*	*count*
Excellent	41.18%	7	35.29%	6
Good	52.94%	9	29.41%	5
Acceptable	5.88%	1	23.53%	4
Poor	0%	0	11.76%	2
Total	100%	17	100%	17

Table 2: Impact of Attending Summer'21 Orientation on Fall'21 experience

	percent	*count*
It was nice to see some familiar faces in my classes	27.27%	3
It was helpful to have seen and met some faculty	45.45%	5
I felt more confident attending classes	27.27%	3
There was no impact	0%	0
Total	100%	11

Table 3: Measure of Impact on Orientation Attendees to Join Clubs

	percent	*count*
SWIC (Supporting Women in Computing) club	57.14%	4
ACM Computer Tech club	0%	0
Not inspired by attending the student orientation	42.86%	3
Total	100%	7

respondents (n=8/11 or 73%) were upper-class women.

Table 1 presents women respondents' rating of their academic and social experiences in computing classes at FSC. Almost 95% of respondents rated their academic experience as good or excellent. This is good news that there are no 'poor' ratings, however it may be interpreted that more satisfied students take the time to complete surveys. Women students' social experience ratings is not quite as favorable and need further investigation.

4.1 Impacts of Summer 2021 (Re)Orientation Program and Women in Computing Club

There were 33 women students that attended the summer'21 (re)orientation. Out of the 19 respondents who took the fall 2021 end of semester survey, 8 of the respondents had attended the (re)orientation which suggests that attending the (re)orientation program did not encourage students to complete the end of semester survey. Table 2 gives details about how attending the orientation impacted students. Note that a respondent could select multiple items from the list while taking the survey. This preliminary data shows that attending (re)orientation program has positively impacted the participants even after only one semester.

The impact of restating SWiC and maintaining it as an active student club since fall 2020 is not directly measured in fall 2021 end of the survey but will be in the upcoming semester. During the (re)orientation program, officers of the only two student clubs of the department, including SWiC officers,

Table 4: Measure of Impact from attending Virtual NYCWiC'21

	percent	*count*
Learned from women in the industry	42.86%	3
Learned from researchers at other regional schools	14.29%	1
Learned about career options	28.57%	2
Learned about some innovative technology and tools	14.29%	1
Total	100%	7

made presentations to the attendees as well as helped with the logistics of the (re)orientation program. At the end of fall 2021 survey, respondents who attended (re)orientation were asked if they were inspired to get involved in the SWiC club and most respondents (n=4/7 or 42%) did, as seen Table 3.

4.2 Impact of Virtual Field Trip in Spring 2021

Only three of the 19 survey respondents had attended the virtual NYCWiC conference in spring 2021. Despite the low numbers, the feedback is positive. Table 4 provides the breakdown of the takeaways from attending the spring 2021 virtual NYCWiC conference. Note that a respondent could select multiple items from the list. The takeaway with learning about career opportunities being the second big takeaway and tied for third is learning from researchers at other regional schools and learning about some innovative technology and tools. In addition, all survey participants were asked if given the opportunity to attend the next NYCWiC conference in April'22 in Lake George, NY and 57% (n=11/19) say they would be interested in attending. This is a positive response but is telling those students are not clear on the benefits of attending a women-centric conference and time should be built to enlighten women students, so they attend and reap the benefits.

4.3 Free Form Responses

Survey respondents were given the opportunity to provide free form feedback. There were 6 responses (n=6/19) as presented below in which two made no suggestions. One common theme was to have more fun and interactive events with SWiC rather than information sessions with guest speakers. This can clearly enhance the student experience. Another item of mention in the responses was that there should be elections for SWiC officers. This is a great suggestion as to date club officers have been volunteers from a small pool of women students. The club membership has grown over the past year to include members from multiple related computing degree programs; therefore, it has been added to the calendar late in spring 2022 semester to solicit volunteers for officer positions and to hold elections. One item worth mentioning however, is that because of most FSC students are commuters (90%) and many work off campus (79%) in addition to attending school, it is often hard to get volunteers to take on leadership roles. The experience of the club advisors has also been that those who take on the leadership roles need to be mentored and cultivated. Another dilemma is that traditionally half of the incoming students are transfers, therefore there are only two years for them to take on a leadership role and once cultivated, they leave. There has been a significant effort to have a leadership group (President, VP, Secretary, Treasurer) in all stages of their studies so there is a succession plan upon leaders graduating. The following lists the free form responses.

- *"May be some more fun activities to do in the SWiC meetings besides informational stuff."*
- *"I think that the orientation was a great program and as someone who had no on campus experience before fall, it really gave me confidence and encouragement to do well this semester. I do think the SWiC club can have more opportunities and events. One suggestion I would like to make is to involve more students and have some sort of elections when it comes to choosing the officer members. Other than that, I did enjoy being a part of the club and its events."*
- *"I have not attended or gone to any of these events, so I am not sure if I am in a position to make any suggestions. I would love to attend, in the future."*

5 Discussion

The preliminary survey results reviewed in this paper raise some interesting questions and provide insights to the experience of women students in the program. Two groups of implications about initiatives and survey administration are identified and listed below.

- *Implications for the Initiatives*
  - A general observation was that student academic experience was rated positive in which social experience rated mediocre. This is a flag that needs to be addressed.
  - Suggestions and feedback to have SWiC meetings more interactive and fun versus organized speakers is to be taken into serious consideration and implementation.

    - Respondents were inquiring about SWiC officer elections which haven't been public and should be modified going forward.

- *Implication for the Survey Administration*
    - Timing and delivery of survey administration should be reviewed for a higher response rate.
    - Demographics of survey participants could include male students to explore if academic and social experiences differ by gender.
    - Race, ethnicity, and disability status of respondents should also be collected.
    - Additional questions about club experience, attending conferences and (re)orientation should be added.
    - Many respondents who attended the (re)orientation were inspired to get involved in SWiC, but not the co-gender ACM Computer Tech Club, a deeper analysis into the reasons for this should be undertaken.
    - The response to the interest level of attending a women-centric conference was moderate; however, the expectation was it would be higher. This is interpreted that potential student attendees need to learn the benefits of attending women-centric conferences.

6 Conclusion and Future Work

This paper addresses the three initiatives taken at FSC to reduce the long-lasting gender disparity in its computing degree programs. The early survey results from fall 2021 provide a promising preliminary measurement on how the initiatives improve the experience for women students while pursuing their computing degrees. This valuable feedback provides the researchers with the knowledge to modify initiatives for maximum effectiveness to share with the broader community. The research plan is to conduct surveys at the end of each semester to continue to monitor student feedback to measure the response to the initiatives (Student Club, Orient Program, and Field Trips) and to assess the women student experience. In addition, the researchers will continue to pursue external sources of funding to further fund the existing initiatives and take on additional ones to improve the gender disparity and women experience at FSC.

References

[1] Christine Alvarado and Eugene Judson. "Using targeted conferences to recruit women into computer science". In: *Communications of the ACM* 57.3 (2014), pp. 70–77.

[2] Ilknur Aydin, Mary Villani, and Lisa Cullington. "Designing a (Re)Orientation program for Women Computing Students at a Commuter College and Measuring Its Effectiveness". In: *Proceedings of the 129th ASEE (American Society for Engineering Education) annual conference*. Minneapolis, Minnesota, June 2022.

[3] Heidi Blackburn. "The status of women in STEM in higher education: A review of the literature 2007–2017". In: *Science & Technology Libraries* 36.3 (2017), pp. 235–273.

[4] *Dr. Anita Borg: 50/50 by 2020*. Youtube. `https://www.youtube.com/watch?v=3nImg8vPUe4`. Accessed: 2022-07-30.

[5] Angelia D Gibson, Maria Siopsis, and Karen Beale. "Improving Persistence of STEM Majors at a Liberal Arts College: Evaluation of the Scots Science Scholars Program: Evaluation of the STEM Retention Program at Maryville College". In: *Journal of STEM Education: Innovations and Research* 20.2 (2020).

[6] Misook Heo and L Monique Myrick. "The girls' Computing club: Making positive changes in gender inequity in Computer Science with an informal, female learning community". In: *International Journal of Information and Communication Technology Education (IJICTE)* 5.4 (2009), pp. 44–56.

[7] Barabra L Howard and Lonnie Sharpe Jr. "The summer bridge program: An effective agent in college students' retention". In: *Journal of Interdisciplinary Studies in Education* 7.2 (2019), pp. 20–30.

[8] Joseph A Kitchen, Philip Sadler, and Gerhard Sonnert. "The impact of summer bridge programs on college students' STEM career aspirations". In: *Journal of College Student Development* 59.6 (2018), pp. 698–715.

[9] Maria Klawe. "Increasing female participation in computing: The Harvey Mudd College story". In: *Computer* 46.3 (2013), pp. 56–58.

[10] Dee Michell et al. "Broadening participation not border protection: How universities can support women in computer science". In: *Journal of Higher Education Policy and Management* 39.4 (2017), pp. 406–422.

[11] Burcin Tamer. “Students who Participated in Diversity Conferences are More Confident in Their Ability to Complete Their Undergraduate Degree in Computing”. In: *Computing Research News* 30.7 (Aug. 2018). URL: `https://cra.org/crn/2018/08/diversity-conferences`.

[12] David L Tomasko et al. “Association of Summer Bridge Program Outcomes With STEM Retention of Targeted Demographic Groups.” In: *Journal of College Science Teaching* 45.4 (2016).

[13] Mary Villani and Ilknur Aydin. “Learning from the Journey: A Decade of Supporting Women in Computing at a Commuter State College”. In: *Proceedings of the 6th IEEE RESPECT (Research in Equitable and Sustained Participation in Engineering, Computing, and Technology)*. Virtual, May 2021, pp. 1–2. DOI: `10.1109/RESPECT51740.2021.9620593`.

[14] Mary Villani and Ilknur Aydin. “Mentoring a Women in Computing Student Club: The Good, The Bad, and The Ugly”. In: *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2*. Birds of a Feather Session. Virtual, Mar. 2022, pp. 1191–1191. DOI: `https://doi.org/10.1145/3478432.3499186`.

[15] Mary Villani and Ilknur Aydin. “Solving the gender disparity puzzle in computing disciplines at a commuter state college”. In: *Proceedings of the ISECON (Information Systems Education Conference)*. Vol. 2021-October. Virtual, 2021.

[16] Tina Vrieler, Aletta Nylén, and Åsa Cajander. “Computer science club for girls and boys–a survey study on gender differences”. In: *Computer Science Education* 31.4 (2021), pp. 431–461.

On Teaching and Testing Recursive Programming*

Gulesh Shukla[1] and David G. Wonnacott[2]
[1]Haverford College
Haverford, PA 19041
gulesh.shukla.7@gmail.com
[2]Computer Science Department
Haverford College
Haverford, PA 19041
davew@cs.haverford.edu

Abstract

Novice programmers face many challenges when learning to program. Recursion is often considered a particularly difficult challenge. Considerable research has been done to explore the ways in which new students understand recursive code. Our survey of this research suggests that much of the education research focuses on the use of *tracing* to teach recursion, and to test student understanding thereof.

However, a fundamentally different approach is suggested by our informal conversations with faculty who emphasize recursion in the first or second semester. Specifically, many faculty try to develop students' ability to think *abstractly* about functions they call, and then leverage this thinking to help students think about recursive functions without getting lost in a morass of detail.

In this paper, we consider some of the challenges of teaching recursion via abstraction, propose experiments to measure the success of this approach, and report on the results of a preliminary experiment this past spring. We close with our thoughts about pedagogy, the classification of *notional machines*, and tool support for various types of visualization.

1 Introduction

A significant amount of effort has gone into categorizing the intellectual steps involved in understanding recursion (as well as other code), measuring how well students understand recursion, and identifying the pitfalls that lead to incorrect understanding among students. Here, we review important terminology and relevant concepts.

1.1 Active Flow, Termination, and Passive Flow

Active flow refers to the forward passing of the control when a function calls another function [5]. Thus, each recursive call is an example of active flow. *Passive flow* refers to the backward flow, where control is passed back to the calling function. In recursive code, passive flow begins when the base-case code passes control to the calling process, i.e., when each branch of active flow *terminates*.

So, in a classic recursive factorial function (below), active flow occurs as `fact` calls itself at `fact(val-1)`, termination occurs at the base case's `return 1`, and passive flow brings the sub-answers back to the level of recursion "above", where it is multiplied by `val`.

```
public int fact(int val){
    if( n < 2)
        return 1;
    else{
        return val * fact(val - 1);
    }
}
```

1.2 *Loop* and *Copies* Models of Recursion

Kahney investigated student understanding of recursion [6] and identified two common models. Many students see recursion as, essentially, a strange notation for loops, in which the recursive code is executed repeatedly as the function calls itself (Kahney refers to this as the "loop model"). Novices using the loop model seem to think that, like a loop, a recursive process *entirely* terminates when it reaches the base case. In other words, they seem to ignore the impact of passive flow.

Other students use what Kahney calls the "copies model", in which each recursive call is thought of as calling a *copy* of the function's body. Kahney suggests that expert programmers tend to possess this model (though possession by a novice does not, by itself, guarantee success in reasoning about recursion).

1.3 Notional Machines

A *notional machine* [2] is an abstract computer that is used as a black box to explain and understand the execution of programs. The properties of a notional machine "are implied by the constructs in the programming language employed" [2]. Stack-based notional machine are a popular tool to teach program execution. More recent work that focuses on student understanding sometimes employs a very different framework, e.g., the *substitution notional machine* used by the Tunnel-Wilson et al. [10]. We refer to the classic "stack of variable-value or variable-object-value associations" system as a *dictionary notional machine*, when needing to identify this approach specifically.

1.4 Teaching and Testing Recursion

Scholtz and Sanders conducted a study to test the effectiveness of tracing a tool to teach recursion [8]. They found that tracing is helpful to find output of the simple recursive functions. But, it becomes a liability when applied to complex recursive functions. Additionally, they claim that "trace methods are mechanical processes and their effective application does not necessarily imply real understanding" [8], further raising questions about its value.

Note that tracing is widely used as an approach to teaching and testing the understanding of recursion, even by authors using different notional machines [10]. We see tracing as an idea that transcends the use of the loop model, copies model, and other conceptual models such as the step model discussed by Götschi, Sanders, and Galpin [5],[7]. With each of these models, the students are tracing the steps of *some code* in *some manner*; the differences are in *which* parts of the code and *how* they move between steps.

We believe that *abstraction* could provide an approach that is superior to tracing, in teaching recursion to novice programmers. In other words, treat the recursive calls as black boxes, in the same way that we would treat calls to library functions or functions that the student has written and thoroughly tested. The idea of abstraction is, of course, already central to much CS pedagogy. It is often used to introduce design via "wishful thinking" [1, Sec. 2.3 Symbolic Data], and is connected to the teaching of recursion in both course materials [3] and research [4],[9].

Unfortunately, despite the resources mentioned in the previous paragraph (and considerable informal evidence in the form of statements by CS faculty who discuss using abstraction to teach recursion), we have not found published research that *tests* this approach. Thus, we have undertaken this task of experimental verification.

2 Experiments

We originally planned to compare two groups of students in a 2nd-semester computer science introduction, with some students seeing a review/introduction to recursion in terms of tracing, and others seeing an abstraction-based presentation. However, we were not able to get enough participants to conduct our original experiment (other experimenters doing work with optional outside-of-class activities are encouraged to make a plan for very low attendance). We thus formulated a fall-back experiment, and conducted that.

For our Fall-Back experiment, we could only manage to get 2 participants. But, we still think that we got useful insight from the question asked by students during the teaching session. Furthermore, the after test questionnaire filled by students gave us useful student perspective (summarized below in the Results section).

2.1 Original Experiment

Our original plan was to teach recursion to two groups of students, using abstraction (in one) and tracing (in the other); then, give students recursive problems to practice; and finally, test their understanding.

For abstraction based teaching, we, first, use the base case and general case categorization to explain how original problem depends on sub problem and how base case is an essential part of recursion. We explain how each smaller problem is similar to the original problem. Thus, if we get a bigger problem, our current problem becomes a sub problem for the new problem. Next, we raise an important question, "suppose we are given a solution to the sub problem, how can we use the solution to solve our original problem?" This step helps us build our solution bottom up. For instance, if we want to reverse a singly linked list, we need to change the pointers to merge solution of sub problem with the head to get the fully reverse linked list. We can ask this question as this question applies to every recursive call when we are building our solution, which we start building when we hit base case.

Furthermore, it is crucial that we ask the right question to our recursive call. For instance, we need to know what the recursive call will return and what input it needs. If the recursive call for reversing a singly linked list problem returns anything other than the head of the reversed linked-list, then we are not solving the problem correctly. Similarly, for input, the recursive call should take the head of sub linked-list.

For tracing based teaching, we, still, use the base case and general case structure. However, we become more mechanical. We trace the recursive call until we reach base case. Then, we, like abstraction, focus on how to include the solution from the last recursive call to solve our original problem. We use the previous idea to build the solution as we engaged in passive flow.

2.2 Fall-back Experiment

In Fall-back experiment, we first, asked the students their familiarity with recursion on the scale of 5 and gave them a problem to solve. The problem had three code samples. Students had to find what problem the codes samples are trying to solve (all the code samples were trying to solve the same problem) and which of the code samples do it correctly. We, then, taught recursion to students using both methods: abstraction and tracing. For teaching, we used the power function. We started with an iterative version of the power function, which students had experienced with.

We started our session with a discussion of recursion. We, then, wrote a recursive version of power function. We used the abstract technique that we explained in the 'original experiment' section. For tracing, we used the debugger. After basic program, we engaged in a discussion of how to write a faster version which will reduce our complexity, a reason we use recursion. After the teaching session, we asked the students to solve the same problem given to them in the start of the session plus one additional problem. Additionally, we asked students students the following questions to understand what they think of abstraction:

1. Please share your thoughts about the relative importance of these two ways of thinking. Based on your prior programming experience and your work on the post-lesson test, how useful do you think each will be, as you do recursive programming?

2. Please also share your thoughts about the use of each in education. Based on this and prior learning experiences, which approach should be emphasized in the first few lessons about recursion? Do you have any other thoughts about how to help beginners?

3 Results

Before the teaching session, students were not confident in recursion as the average response to the pre-test survey, where we asked them how comfortable they were with recursion, was 1. Further, both the participants failed to solve the initial problem given to them. During the teaching session, students were able to understand the execution of the program to calculate basic power as we traced through the input using debugger. When we used abstraction to write a faster version of power function, students seemed to understand the main idea: ask your helpful friend the answer to sub-problem who can do the same until we reach a friend who knows the answer to the smallest instance of the problem (base case) and use the result to calculate the result for the input given to current function call.

However, students were confused as we were writing the faster version of power function. In particular, they were not able to understand how we get 10^5 from the recursive call, required to find 10^6. In other words, they were not able to imagine the abstract calls and how are we building the result. We used debugger to trace the program and used our abstract explanation along the way. We saw that the students were able to understand the execution of the program with the help of debugger. This, further, enforces the point that tracing helps in understanding the execution of the programs.

Our post test survey saw an increase of 2 in the confidence level of the students. This is, further, supported by the fact that students were able to describe the two problems given to them. One of the students was able to find correct code samples for the two problems on post-lesson test. The other student was able to find the correct code samples for the first problem.

Below is the response of one of the participant to the post test survey:

1. "I thought that tracing via debugger makes my understanding more concrete while using the big picture method (abstraction) can help me design the recursive model from the start, which is more useful in terms of writing my own recursive method and do the post-lesson test."

2. "I personally prefer method b (abstraction), for I found that method b actually helps me to design the recursive model by letting me understand the basic structure: base case and method (recurrence)."

We can see that the response found abstraction more useful than tracing. The student found that tracing helped him with the execution of the program but abstraction helps him design recursive programs. The ability to design programs indicate that they are able to understand recursion as student with trace mental model were struggling with writing recursive functions [8]. Thus, we think it is a great starting place for abstraction as a tool to teach recursion.

4 Conclusions

Although the small number of respondents in our survey doesn't allow any statistically significant conclusions, our experience and the survey responses have enriched our thinking in several ways.

4.1 Pedagogy

Our original experiment focused on the scientifically-appealing "pure A/B test", with some students learning via tracing, and others via recursion. However, part-way through the session, it became clear (probably to all involved) that *blending* the two may be best. As the above-cited student pointed out, abstraction is great for *design*, and tracing helps to understand what's going on.

As we designed the original experiment, we felt that the exercises and tests can bias ones thinking toward one approach or the other. Specifically, if one focuses on "mystery" functions that produce some unspecified result, and ask questions like "what is *f(7)*", the student has few options other than tracing. However, students may be better prepared for real programming activities by questions for which the purpose of the code is known. Thus, we focused on simple algorithm design questions, and questions that asked students to reason about the relationship between an abstract goal and several proposed recursive functions.

4.2 Notional Machines and Tool Support

Notional machines are often distinguished by their model of the machine state and mechanism for updating that state. The dictionary notional machine is visualized by most debuggers, with a stack of name-value associations, updated by each line of the program in turn. The substitution notional machine can be visualized in, e.g., the "stepper" tool for student languages in the Dr. Racket environment. This shows a series of rewrites of the function call, until finally it is rewritten into the result.

We have been discussion a different aspect of notional machine design: Instead of focusing on that execution model, we also wish to consider how readily it lets us control the level of abstraction. Debuggers in professional development environments typically give the choice, at a given function call, of "step over" or "step into", letting the user choose the appropriate level of abstraction to help them understand their code. However, many educational tools, such as the popular `pythontutor.org`, lack the "step over" option. This creates a challenge when using the excellent visualizations produced by such tools when one wants to skip the details within a call.

4.3 Future Work

We find ourselves torn between approaches to future teaching. Our students might benefit if we immediately adopt the pedagogy we feel is superior, teaching design and high-level understanding via abstraction, but supplementing it with occasional detailed examples of tracing. However, something like our original A/B comparison could provide better evidence for our feeling about this approach.

We will have several opportunities to perform experiments in *required* elements of at least two spring courses in 2023: The instructor for our CS2 course has confirmed that we can hold a required "recursion review" next spring, instead of the optional review sessions discussed in this paper; we will also have an opportunity to teach recursion in several sections of one or more courses that cover programming in the context of another field (e.g., Economics or Physics).

We are also keenly aware of the challenges presented by lack of tools, e.g., when trying to teach abstraction-based reasoning about function calls in a visualizer that lacks "step over". We are currently exploring various work-arounds, considering the level of challenge presented by forking and editing PythonTutor, and considering teaching using only and standard debugger.

We invite feedback and suggestions from CCSC participants, regarding these questions.

References

[1] Harold Abelson and Gerald J. Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, MA, USA, 2nd edition, 1996.

[2] Benedict Du Boulay. Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1):57–73, 1986.

[3] Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, and Shriram Krishnamurthi. *How to design programs: an introduction to programming and computing*. MIT Press, 2018.

[4] Gary Ford. An implementation-independent approach to teaching recursion. In *Proceedings of the Fifteenth SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '84, page 213–216, New York, NY, USA, 1984. Association for Computing Machinery.

[5] Tina Götschi, Ian Sanders, and Vashti Galpin. Mental models of recursion. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '03, page 346–350, New York, NY, USA, 2003. Association for Computing Machinery.

[6] Hank Kahney. What do novice programmers know about recursion. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '83, page 235–239, New York, NY, USA, 1983. Association for Computing Machinery.

[7] Ian Sanders, Vashti Galpin, and Tina Götschi. Mental models of recursion revisited. In *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, ITICSE '06, page 138–142, New York, NY, USA, 2006. Association for Computing Machinery.

[8] Tamarisk Lurlyn Scholtz and Ian Sanders. Mental models of recursion: Investigating students' understanding of recursion. In *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '10, page 103–107, New York, NY, USA, 2010. Association for Computing Machinery.

[9] Raja Sooriamurthi. Problems in comprehending recursion and suggested solutions. In *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '01, page 25–28, New York, NY, USA, 2001. Association for Computing Machinery.

[10] Preston Tunnell Wilson, Kathi Fisler, and Shriram Krishnamurthi. Evaluating the tracing of recursion in the substitution notional machine. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, page 1023–1028, New York, NY, USA, 2018. Association for Computing Machinery.

Experience in Teaching Cloud Computing with a Project-Based Approach*

Linh B. Ngo
Computer Science
West Chester University of Pennsylvania
West Chester, PA, 19380
lngo@wcupa.edu

Abstract

This work presents the development and evolution of a cloud computing course that focuses on infrastructure and orchestration aspects of cloud ecosystems. Over the period of four semesters (once per year), course materials and teaching approaches have been revised to adapt to the advances in cloud computing technologies and other indirect changes that can impact students' learning environments. The course's outcomes are discussed using students' project completion rate and quality.

1 Introduction

Over the years, cloud computing has gradually become integral to the computer science curriculum. Different aspects of cloud computing can be taught as a single topic course, a module within another topic, or a hands-on tool. The pervasiveness of cloud computing has resulted in an extensive set of knowledge areas (KAs) and learning objectives (LOs) [1]. Several developments of cloud computing topics focus on fundamental cloud concepts in virtualization of computing resources, including networking and storage [7, 17, 25]. Others study the *-as-a-service* aspect of cloud computing [12, 24]. In the early 2010s, educational computing platforms for these courses are often on-site clusters running virtual machines.

The availability of cloud resources for education from both industry (e.g., Amazon AWS, Google GCP, and Microsoft Azure) and academic (e.g., CloudLab [21] and Chameleon [18]) facilitates the development of more complex cloud computing courses. In [20], a course covering both fundamental cloud concepts and cloud-based computing models were developed. The course includes both infrastructure and service-related projects, and students carry out work using Amazon AWS resources. Similarly, [8] combines concepts on cloud computing fundamentals and big data infrastructures. Google Cloud Platform enables the integration of cloud-based storage into a traditional database management system course [9]. Microsoft Azure and Amazon AWS ecosystems are incorporated in a IT/CS course that uses activity-based learning [13]. The work of [4] provides a road-map for a course that can be adapted for either beginners or advanced students. The syllabus template leverages Amazon AWS, Google Cloud Platform, and Chameleon Cloud. To streamline the interaction between on-site, industry-based, and academic-based cloud resources, a management toolkit called EasyCloud has been developed [3].

Our work contributes to the development of cloud computing curriculum via the integration of Cloud Orchestration (CO) and Service-Oriented Architecture (SOA) KAs into our course. In addition to the core Fundamentals of Cloud Computing (FCC) KA, concepts of CO and SOA are taught via project-based learning.

The remainder of this paper is organized as follows. Section 2 presents our basic learning objectives, available computing environments for educational activities, and the evolution of the course's learning materials over time. We also discuss the motivation behind these changes. Section 3 analyzes students' feedback regarding the latest course offering. Section 4 concludes the paper and discusses future work.

2 Course Development

The course has been taught over a period of four years during Spring semesters. The course description is generic and has remained unchanged: *"... provides an introductory overview to the technologies that enable cloud computing. Topics covered include basic concepts about cloud computing, and advanced technical concepts regarding virtualization, containerization, and orchestration."*. Course assessments include quizzes, exams, one warm-up assignment, and one major project. For hands-on learning activities, students use CloudLab, a federal research and educational computing infrastructure. The course's materials are maintained on a public GitHub repository [19]. Changes made over time can be accessed from the repository's commit history.

Quizzes and exams are multiple-choice and assess students' understanding of key theoretical concepts and important hands-on technical aspects. The ratio between the two areas in the question sets are 60/40: 60% of the questions are theory-based, and the remaining 40% assess whether students understand the hands-on activities rather than just simply copy and paste commands. The warm-up assignment serves as a brief introduction to Linux and the command line interface. Hands-on activities accompany key technical concepts and help prepare students for the course project.

2.1 CloudLab

Funded by the National Science Foundation in 2014, CloudLab has been built on the successes of the Global Environment for Network Innovations (GENI) [5] in order to provide researchers with a robust cloud-based environment for next generation computing research [21]. These resources are distributed across several U.S. institutions. As of Summer 2022, CloudLab boasts an impressive collection of hardware. At the Utah site, there is a total of 785 nodes, including 315 with ARMv8, 270 with Intel Xeon-D, and 200 with Intel Broadwell. The compute nodes at Wisconsin include 270 Intel Haswell nodes with memory ranging between 120GB and 160GB and 260 Intel Skylake nodes with memory ranging between 128GB and 192GB. At Clemson University, there are 100 nodes running Intel Ivy Bridges, 88 nodes running Intel Haswell, and 72 nodes running Intel Skylake. All of Clemson's compute nodes have large memory (between 256GB and 384GB), and there are also two additional storage-intensive nodes that have a total of 270TB of storage available [22].

In order to provision resources using CloudLab, a researcher needs to describe the necessary computers, network topology, startup commands, and how they all fit together in a resource description document. CloudLab provides a graphical interface, as shown in Figure 1, inside a web browser that allows users to visually design this document through drag-and-drop actions. For large and complex profiles, this document can also be automatically generated using Python code. CloudLab supports direct integration between public GitHub repositories containing the infrastructure code and CloudLab profile storage infrastructure.

2.2 Spring 2019

In this first offering, the course focused primarily on virtualization and containerization. Cloud orchestration was discussed and hands-on activities are carried out, but the topic was not as integrated in the project. The primary learning environments was CloudLab. In addition, students were provided instructions to set up VirtualBox [26] on their personal computer to create a

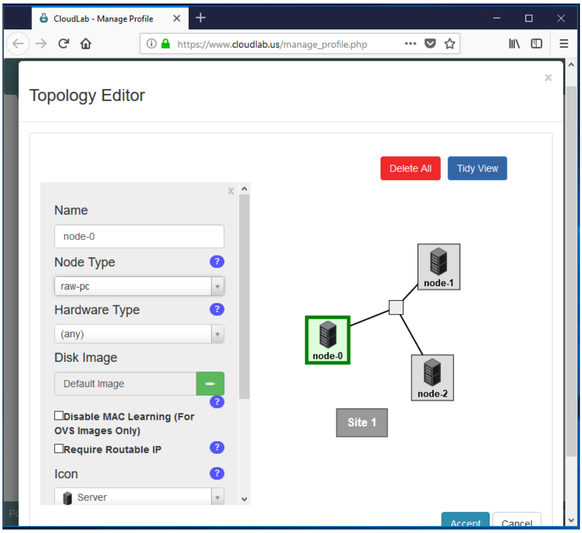


Figure 1: CloudLab's GUI for designing experimental profiles

virtualization environment. For the warm-up assignment, students installed and configured the slimmed-down version of Alpine Linux [2].They were also to setup and test a web server.

The key theoretical concepts in virtualization, containerization, and orchestration were demonstrated through lectures about KVM, Docker, Docker Swarm, OpenStack, and Kubernetes. While there was no required textbook, papers on these subjects were presented and discussed [15, 23, 6]. Hands-on activities were built upon the warm-up assignment. They included:

- Launch an OpenStack infrastructure based on a profile on CloudLab. Upload the virtual image created from the warm-up assignment. Enable ingress/egress security and map public IP address so that that the web server can be accessed externally.
- Recreate the web server installation in a container environment and test the port mapping capabilities of Docker.
- Deploy the web server container on Docker Swarm.

There were three different categories for project selection. The categories were created based on the perspective of cloud consumer, cloud provider, and cloud researcher. In the first category, the goal was to install and deploy computing services in the cloud. The computing services were required to have at least two services (e.g., one web server and one back end database) and the services were to be virtualized/containerized and deployed on OpenStack or Docker Swarm. In the second category, the task was to install and configure a container engine for the default OpenStack profile on CloudLab, which only supports KVM. For the last category, the task was to rerun the benchmarks for CPU and memory access time based on the evaluation described in [10] for KVM and Docker and to update the evaluation with Singularity's performance [16].

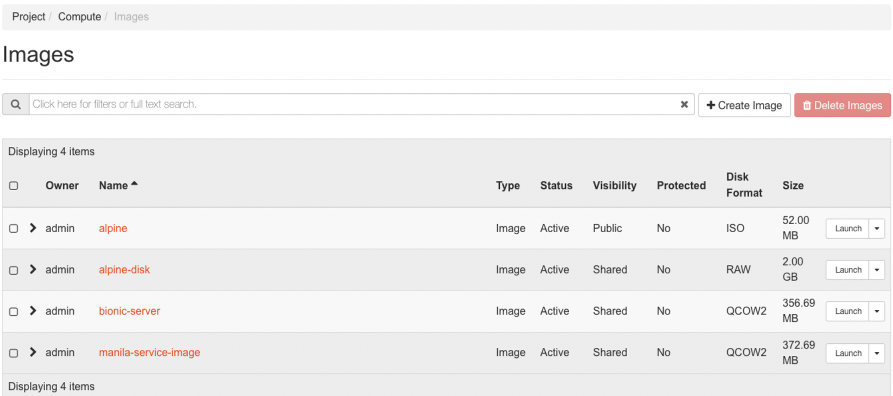
2.3 Spring 2020

The Spring 2020 was impacted with COVID as all courses are moved online after Spring Break. Students' performance on assignment and projects were reduced. All contents of the course remained the same as Spring 2019.

2.4 Spring 2021

Several changes in software and hardware ecosystems led to changes in the course's contents. The first change came from MacOS' introduction of new Apple M1 CPUs. This made VirtualBox not working on the new Mac laptops. Similar issues arrived with the latest updates for Windows 10 and Windows 11. To overcome these issues, the warm-up assignment had to be done completely on the OpenStack deployment of CloudLab. The steps were changed accordingly. The Alpine Linux's installation image needed to be uploaded to OpenStack, and computing instance and volume needed to be allocated and launched. After installation and configuration was completed, the updated volume was saved as a persistent image, as shown in Figure 2. For testing purpose, students were to launch a new computing instance using this persistent image, attach a public IP address, and observe the resulting web server via this public IP address (see Figure 3).

The second change came from the rising popularity of Kubernetes. More specifically, we introduced the concept of continuous development and continuous integration (CD/CI) into the settings of cloud computing through these activities using Jenkins [14] and Kubernetes. The hands-on activities allowed students to observe how Jenkins containers are setup and launched on a Kubernetes cluster. Through the deployment of Jenkins and its relatively more complex components, the concepts of *everything as a services* and *infrastructure as code* were further highlighted. Additional hands-on activities demonstrated how Jenkins can also leverage Kubernetes to build a fully operational CD/CI pipeline where containerized services are built, tested, and deployed to support live updates. An example pipeline is shown in Figure 4.


Project / Compute / Images

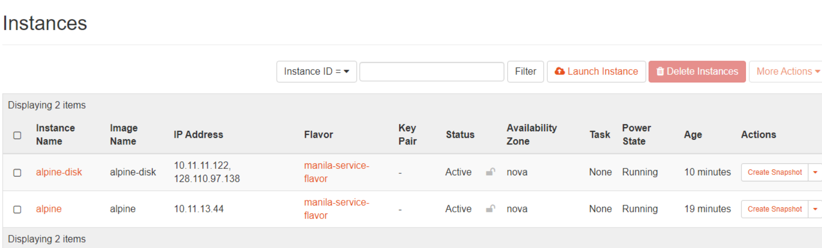
Images

Displaying 4 items

Owner	Name	Type	Status	Visibility	Protected	Disk Format	Size
admin	alpine	Image	Active	Public	No	ISO	52.00 MB
admin	alpine-disk	Image	Active	Shared	No	RAW	2.00 GB
admin	bionic-server	Image	Active	Shared	No	QCOW2	356.69 MB
admin	manila-service-image	Image	Active	Shared	No	QCOW2	372.69 MB

Displaying 4 items

Figure 2: Alpine image (before installation) and volume (after installation)


Instances

Displaying 2 items

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
alpine-disk	alpine-disk	10.11.11.122, 128.110.97.138	manila-service-flavor	-	Active	nova	None	Running	10 minutes	Create Snapshot
alpine	alpine	10.11.13.44	manila-service-flavor	-	Active	nova	None	Running	19 minutes	Create Snapshot

Displaying 2 items

Figure 3: Alpine instance with attached public IP

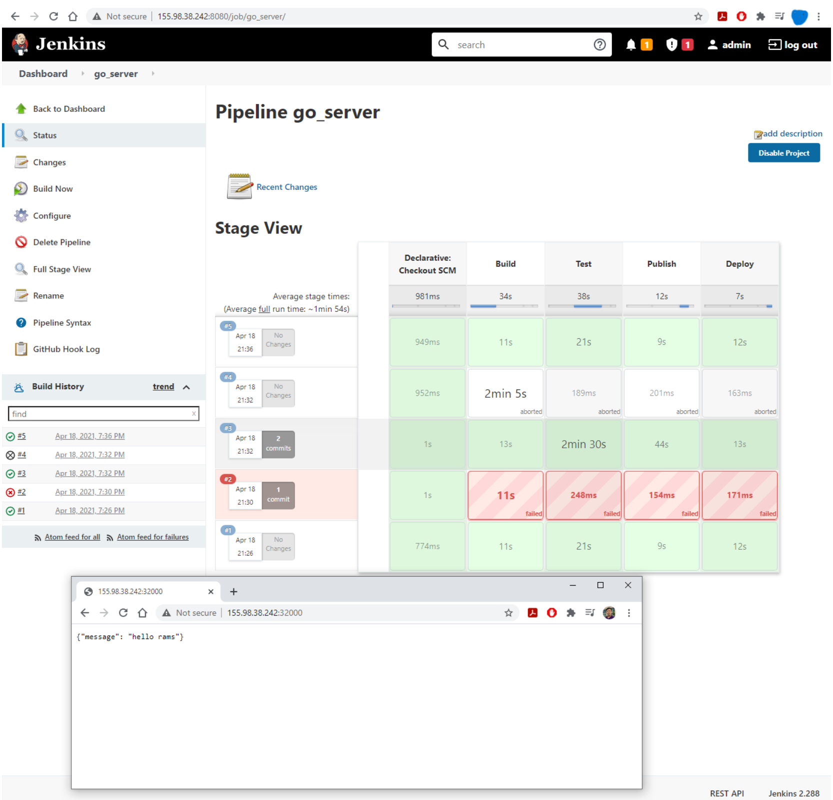


Figure 4: Jenkins pipeline supporting deployment on Kubernetes cluster)

The project categories remained the same. However, the scope for the performance evaluation (cloud research perspective) was reduced. The requirements for performance evaluation were now to only select one benchmark (CPU or memory) rather than two. For cloud customer, student projects were encouraged to use Kubernetes instead of Docker Swarm to deploy containers.

2.5 Spring 2022

There were two major changes made to the contents of the course this semester. First, Docker Swarm materials were archived and no longer taught as a regular topic. Second, the project categories were reduced to just one that combine both cloud provider and cloud consumer. This project required student teams to:

- Setup a supporting Kubernetes cluster in CloudLab
- Deploy Jenkins
- Design and implement a cloud service including at least three components that are containerized separately.
- Design a Jenkins pipeline that integrate the building, testing, and deployment of the containerized components to ensure continuous integration of the cloud services.

In previous semesters, projects were introduced halfway through the semester. This semester, the project was introduced during the first week of class. The requirements of the project were discussed in details. Various stages of the project and the in-class lectures were designed such that they are intertwined. In one such example, the project motivations, deploying services in cloud, were framed from the perspective needs of new startups that are interested in developing and providing online services but lack funding to support physical computing servers. This provided a path to the introduction of cloud computing and its relevant characteristics. In another example, the Docker-based hands-on activities accompanying the containerization lecture helped students in preparing the Docker image for their proposed service components. As the prototype service components were integrated, students would need to leverage the cloud-based CD/CI process with Kubernetes and Jenkins to streamline their testing/building/deployment process.

3 Discussion

Cloud computing has remained a topic of interests for students during all semesters that the course was offered. The course workload is demanding, as reflected in students' written feedback at the end of each semester. On the other hand, even as a technical elective, the course is able to maintain both high enrollment and high retention rate. This is reflected through the students' numerical scoring on the course' contents, organization, delivery, and several other factors. Over the semesters, there are two major challenges in teaching this course.

The first challenge has to do with the rapid pace of new technical advances in cloud computing. While fundamental cloud concepts remain consistent, underlying technologies are updated and deprecated at more frequent rate. As a result, hands-on activities and project ideas need to be changed to ensure that the course can keep up with real-world technologies. Examples of adaptations include the phasing out of Docker Swarm and increasing in emphasis for Kubernetes. Indirect non-cloud changes also impact the class. For example, hardware changes for Mac and operating system updates for Windows disable VirtualBox-based activities, facilitating modifications to make these activities completely cloud-based rather than desktop/laptop-based. There is no obvious solution to this challenge except for instructors to try to stay on top of new materials. Sharing course materials and leveraging federal academic infrastructures like CloudLab will help avoid duplicated efforts between different institutions.

The second challenge arises from the comprehensive technical demands for working with cloud infrastructure. With the focus on system and infrastructure, students are required to pick up skills in Linux administration, Linux-based software installation and configuration, scripts. At the same time, cloud computing topics such as virtualization, containerization, and orchestration draw from topics in computer systems, operating systems, and networking. Many students expressed concerns in being overwhelmed with new materials in their verbal feedback. It is interesting to note that despite the complaints, students understand the needs to include both theoretical contents and hands-on activities for the course. Moving the project to the beginning of the semester and integrate the project's recommended progress to the course's lectures is an approach following project-based learning [11]. This helps students to immediately apply in-class contents to their project progression.

Having different project categories in the earlier semesters sounded fine on paper. In reality, few groups want to go with either cloud provider or cloud research categories, and even fewer are successful. While the amount of acquired knowledge is significant even with incomplete work, the inability to see the project through had unnecessarily dampened the students' spirit. Focusing only on one project category allows for better alignment with project-based learning, leading to a higher chance of success. It is interesting to note that while concern for heavy workload remains, students have become more understanding and appreciative of the complexity of cloud computing knowledge through seeing how these concepts can be tie-in to the project. A number of constructive suggestions were given to the instructor, including providing short snippets of video recordings for hands-on activities, making hands-on activities graded submission to provide incentive, and giving future students examples of successful projects.

Overall quality rating of students' projects over the semesters can be summarized as follows:

- Spring 2019: With 43 students, there were 14 groups in total. Two groups attempted the cloud provider category, two groups attempted the cloud research, and the remaining ten attempted the cloud consumer category. For cloud provider, both groups chose to augment the default OpenStack profile of CloudLab so that it supports Docker containers instead of just KVM. This required installing and configuring additional packages for OpenStack. The cloud consumer groups selected varieties of webserver/-database combination for different user-case scenarios. The differences among the groups lay in the selection of web server engine (Apache, Nginx ...) and databases (MySQL, Postgres, MariaDB ...) and the design of web pages and data tables. At the end of the semester, all cloud provider and cloud researcher projects were considered satisfactory as they were able to demonstrate significant inclusion of cloud technologies even though no group fully complete all requirements of the project deliverable. For cloud consumer groups, only one group was able to deploy their components inside OpenStack. The remaining groups were able to partially complete the tasks, meaning that they were able to configure the individual components and set up the local web page and data tables. However, they were not able to to *connect* these components inside the cloud via container services.

- Spring 2020: The semester was interrupted halfway through due to covid, and the quality of projects suffered. Projects of this semester are ommitted from this discussion.

- Spring 2021: There are 8 groups for 33 students. Two groups selected the cloud research category and the remainders selected cloud consumer category. There was no interest in the cloud provider category. The scope of individual project design were expanded comparing to the previous semesters with several groups attempting additional components such as LDAP servers for authentication and external computing service for API calls. This semester, four projects were considered satisfactory and three partially complete. One group was not able to carry out the work but this was due to group dynamic.

- Spring 2022: The surge post-covid led to two sections of the course being offered in this semester. There were a total of 16 groups across both sections. The project topics varied and resembled the real world apps better than the previous semesters. For example, there were chess playing app, automatic calendar integration app, online store services, coin mining services, and many others. It was interesting to observe that two sections have different approaches to carrying out the projects. Groups in one section tended to focus on setting up the Kubernetes cluster and the CD/CI pipeline first, while groups in the other developed the individual component containers first before working on the cloud cluster. The former had a slower start comparing to the latter, as indicated in the mid-semester report, but in the end, the former had a better completion rate than the latter. Among all groups four were able to demonstrate fully end-to-end final products, ten had the Kubernetes cluster up and running with either the CD/CI pipeline workings or the individual components working, and two groups only had the individual containers developed but no cloud infrastructure.

4 Conclusion and Future Work

Through four semesters/years of teaching Cloud Computing, we have become appreciative of the complexity and challenges in teaching this subject. This work provides another perspective in teaching cloud computing with emphasis on infrastructure and end-to-end cloud service. Future potential augmentation and modification for this course includes:

- *Less OpenStack more Kubernetes:* While both OpenStack and Kubernetes represent different key concepts in the entire cloud ecosystem, it is not beneficial to try to include every concepts in hands-on/warm-up activities. As latest version of Docker Desktop supports both Mac and Windows, and as Windows' Linux Kernel Subsystem become part of the regular release, a warm-up assignment using Docker/Kubernetes on local computers will be more useful.

- *More graded assignments/labs:* Variety of in-class hands-on activities will be created as labs to help assessing students' understanding of hands-on activities.

- *Sample projects:* Highly successful projects from previous semester will be highlighted to demonstrate progression and expectation.

References

[1] Joshua Adams et al. "Cloud computing curriculum: Developing exemplar modules for general course inclusion". In: *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*. 2020, pp. 151–172.

[2] *Alpine Linux*. https://www.alpinelinux.org/. 2022.

[3] Cosimo Anglano, Massimo Canonico, and Marco Guazzone. "Easycloud: a rule based toolkit for multi-platform cloud/edge service management". In: *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE. 2020, pp. 188–195.

[4] Cosimo Anglano, Massimo Canonico, and Marco Guazzone. "Teaching Cloud Computing: Motivations, Challenges and Tools". In: *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE. 2020, pp. 300–306.

[5] Mark Berman et al. "GENI: A federated testbed for innovative network experiments". In: *Computer Networks* 61 (2014), pp. 5–23.

[6] Brendan Burns et al. "Borg, omega, and kubernetes". In: *Communications of the ACM* 59.5 (2016), pp. 50–57.

[7] Ling Chen et al. "Introducing cloud computing topics in curricula". In: *Journal of Information Systems Education* 23.3 (2012), p. 315.

[8] Debzani Deb, Muztaba Fuad, and Keith Irwin. "A module-based approach to teaching big data and cloud computing topics at cs undergraduate level". In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 2019, pp. 2–8.

[9] Aijuan Dong. "Integrating cloud computing into computer science curriculum: a pilot study with teaching introductory database courses". In: *Journal of Computing Sciences in Colleges* 37.3 (2021), pp. 120–131.

[10] Wes Felter et al. "An updated performance comparison of virtual machines and linux containers". In: *2015 IEEE international symposium on performance analysis of systems and software (ISPASS)*. IEEE. 2015, pp. 171–172.

[11] Kevin Gary. "Project-based learning". In: *Computer* 48.9 (2015), pp. 98–100.

[12] Lee Gillam, Bin Li, and John O'Loughlin. "Teaching clouds: Lessons taught and lessons learnt". In: *Cloud computing for teaching and learning: Strategies for design and implementation*. IGI Global, 2012, pp. 82–94.

[13] Michalina Hendon and Loreen Powell. "Activity based learning for cloud computing". In: *Journal of Computing Sciences in Colleges* 35.8 (2020), pp. 176–185.

[14] *Jenkins: build great things at any scale*. https://www.jenkins.io/. 2014.

[15] Avi Kivity et al. “kvm: the Linux virtual machine monitor”. In: *Proceedings of the Linux symposium.* Vol. 1. 8. Dttawa, Dntorio, Canada. 2007, pp. 225–230.

[16] Gregory M Kurtzer, Vanessa Sochat, and Michael W Bauer. “Singularity: Scientific containers for mobility of compute”. In: *PloS one* 12.5 (2017), e0177459.

[17] Peng Li and Lee W Toderick. “Cloud in cloud: approaches and implementations”. In: *Proceedings of the 2010 ACM conference on Information technology education.* 2010, pp. 105–110.

[18] Joe Mambretti, Jim Chen, and Fei Yeh. “Next generation clouds, the chameleon cloud testbed, and software defined networking (sdn)”. In: *2015 International Conference on Cloud Computing Research and Innovation (ICCCRI).* IEEE. 2015, pp. 73–79.

[19] Linh Bao Ngo. *Introduction to Cloud Computing.* https://github.com/class-master/csc468. 2022.

[20] M Suhail Rehman et al. “A cloud computing course: from systems to services”. In: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education.* 2015, pp. 338–343.

[21] Robert Ricci, Eric Eide, and CloudLab Team. “Introducing CloudLab: Scientific infrastructure for advancing cloud architectures and applications”. In: *; login:: the magazine of USENIX & SAGE* 39.6 (2014), pp. 36–38.

[22] Robert Ricci et al. *CloudLab Phase II: Community Infrastructure To Expand the Frontiers of Cloud Computing Research.* https://www.nsf.gov/awardsearch/showAward?AWD_ID=1743363. 2017.

[23] Tiago Rosado and Jorge Bernardino. “An overview of openstack architecture”. In: *Proceedings of the 18th International Database Engineering & Applications Symposium.* 2014, pp. 366–367.

[24] Ian Sommerville. “Teaching cloud computing: a software engineering perspective”. In: *Journal of Systems and Software* 86.9 (2013), pp. 2330–2332.

[25] Xinli Wang et al. “Introducing cloud computing with a senior design project in undergraduate education of computer system and network administration”. In: *Proceedings of the 2011 conference on Information technology education.* 2011, pp. 177–182.

[26] Jon Watson. “Virtualbox: bits and bytes masquerading as machines”. In: *Linux Journal* 2008.166 (2008), p. 1.

Teaching Parallel Programming in Introductory CS Classes Using Java Patternlets in Interactive Notebooks*

Ruth Kurniawati
Computer and Information Science Department
Westfield State University
Westfield, MA 01086
rkurniawati@westfield.ma.edu

Abstract

There has been a recent push towards the early introduction of parallel programming in introductory CS courses, with classes using both plugged and unplugged activities. For the plugged activity, teaching material using OpenMP and simplified parallel programming patterns (patternlets) has been gaining popularity, especially in introductory classes that use C or C++. However, in Java-based introductory CS courses, introducing parallel programming using patternlets is challenging because there is no native support for OpenMP-like directives in the Java programming language. This paper reports my experiences in introducing parallel programming in introductory CS courses at Westfield State University. I describe the use of interactive notebooks containing Java patternlets using both Pyjama OpenMP-like directives and Java parallel streams. Students' perceptions of their learning experience with these notebooks was generally positive or neutral, however the Java parallel stream notebook may be more effective in helping students complete parallel programming assignments.

1 Introduction

A shift to multiple cores began about 20 years ago when chip manufacturers hit the "power wall" [9]. Today, the presence of multiple cores is standard in both modern computers and mobile devices. Most personal computers and laptops now have multiple cores, and cloud-based computing resources with multiple cores are readily available. In this new environment, parallel and distributed computing (PDC) has become the norm rather than the exception in modern software applications. It is therefore imperative for computer science (CS) students to gain an understanding and mastery of parallel and distributed computing.

In 2019, the Accreditation Board for Engineering and Technology (ABET) introduced requirements for exposure to PDC in Computer Science in version 2.0 of their accreditation criteria [5]. Over this the same time, there has also been a push towards the early introduction of PDC in introductory CS courses [12, 14], and materials have been developed for teaching PDC in introductory CS courses [2, 12, 14]. Most of these new materials are based on the MPI and OpenMP standards that continue to predominate.

In this paper, I report my experience in teaching PDC in two Java-based introductory CS courses at Westfield State University. The materials developed focused on shared-memory parallel programs in order to allow students to get started with PDC concepts without the complexities related to distributed programs. Parallel programming patternlets [1], small and minimalist example programs, were used, allowing the students to immediately experiment with the programs while at the same time can use the working example programs for future reference. Two interactive Java patternlet notebooks were created to present the introductory material to students, one with examples in Pyjama OpenMP-like directives and the other using Java parallel streams. The advantages and disadvantages of each approach are discussed.

2 Parallel Programming Patternlets

A design pattern describes a recurring problem in a domain and provides a solution to the problem that can be reused effectively [6]. The use of software design patterns has helped advance software development into a more rigorous and repeatable engineering endeavor. Mastering common patterns in software development allows new programmers to think about a problem at a higher level of abstraction while reusing solutions that others with more experience have developed.

In parallel programming, there have been a couple of attempts towards identifying and cataloguing patterns [11, 9]. Both these approaches use a hierarchy of patterns that will allow the programmer to approach the problem starting from a high-level (design level), to the mid-level (algorithm structure and supporting structural levels), and then to the low-level (implementation mechanism level) constructs.

It can be quite difficult to explain these patterns to students who are new to parallel programming, and the subject can often be overwhelming. To address this, *patternlets*, simplified minimalist patterns, have been developed to introduce students to parallel programming [1]. These patternlets are presented as collections of example programs that illustrate a parallel pattern. Each example program can be compiled and run independently, so that students can experiment and use these small examples as a working model. To ensure that students can easily understand the pattern, the example programs are minimalist. However, these examples are also scalable so that students can appreciate the changes (such as speed-up or problems due to race conditions) that happen when there are multiple threads present. The CSinParallel project [2] repository contains collections of patternlets in several programming languages and application programming interfaces (APIs), including C (with OpenMP, Open MPI, and pthread), Python (with Open MPI), and Java (with Open MPI). The OpenMP and pthread patternlets are examples of parallel shared memory programs. The MPI-based patternlets are, however, examples of distributed programs which use message passing to communicate, making them unsuitable for teaching introduction to parallel programming.

3 Parallel Programming in Java

In early versions of Java, parallel programs are implemented using the functionalities provided through the `Thread` and `Runnable` classes in the `java.lang` package. Creating a multithreaded program requires that the student has mastered object-oriented programming so that they can create an implementation of the `Runnable` interface. The implementation of this interface is passed to the `Thread.start` API so that a thread can be launched. Once launched, one can manage thread execution using other APIs, such as `join`, `sleep`, or `interrupt`. Higher level concurrency constructs were added in later versions of Java, including thread pools, executor services, a fork/join framework and parallel streams. Currently there is an ongoing effort to add support for light-weight virtual threads and structured concurrency in Java [13, 4].

4 Teaching Parallel Programming with Java

Teaching PDC in Java-based introductory CS courses is a challenging task. Not only are the concepts difficult to grasp, but programming activities are even more challenging for students in introductory CS courses who are still developing foundational programming skills.

There are some teaching materials available to teach shared-memory parallel programming in Java using the fork-join construct [8]. This is suitable for more advanced students who have mastered both object-oriented programming in Java as well as more advanced concepts such as divide-and-conquer and recursion. However, these constructs are much more verbose compared to OpenMP and may be distracting to the students in introductory courses.

For introductory parallel programming, OpenMP is a widely used API because it allows the development of portable shared-memory parallel programs using concise and simple directives. This allows instructors to focus on teaching higher level concepts and issues related to parallel programming, rather than the details of adding parallelism. There is, unfortunately, no native support for OpenMP in the Java programming language. A Java compiler and runtime tool called Pyjama [15], however, provides support for OpenMP-like directives in the form of a Java comment (see Listing 1). Pyjama translates Java code with the directive to code that calls the appropriate Java concurrency constructs. Open source teaching materials that use Pyjama are available from CDER [7]. Additionally, I have previously developed open source Pyjama materials that use a Jupiter notebook patternlet-based teaching approach [10] which are used in the courses described in this paper.

Listing 1: Parallel loop with Pyjama example

```
class ParallelLoopEqualChunks {

  public static void main(String[] args) {
    if (args.length >= 1) {
      int numThreads = Integer.parseInt(args[0]);
      Pyjama.omp_set_num_threads(numThreads);
    }

    //#omp parallel for
    for (int i = 0; i < 16; i++) {
      int id = Pyjama.omp_get_thread_num();
      System.out.println("Thread " + id + ": iteration " + i);
    }
  }
}
```

4.1 Method 1: Teaching with Pyjama-based Notebooks

Inspired by earlier work on patternlets [1, 3, 2] and drawing on Pyjama's concise OpenMP-like directives, I created an interactive Jupyter notebook containing patternlets implemented in the Java programming language using Pyjama directives. The patternlets in this notebook include the Single Program Multiple Data (SPMD), parallel loop, and reduction patternlets. The notebook also contains a few larger example programs from the CSinParallel project [2] that I have adapted to Pyjama.

An example of the Notebook's parallel loop patternlet is shown in Listing 1. Readers who are familiar with OpenMP will see that this example program looks very similar to the corresponding C program that uses OpenMP.

The use of Pyjama directives requires students to use a slightly different workflow to compile and execute the patternlet examples. They must compile the source code using the Pyjama compiler and then run the resulting class file with the Pyjama runtime (see Figure 1). This change in the workflow would likely be problematic for students in introductory CS classes who do not yet have much experience with the command-line, but it is not problematic when students use the provided interactive notebook because the commands are already provided and students need only execute the cells containing the commands.

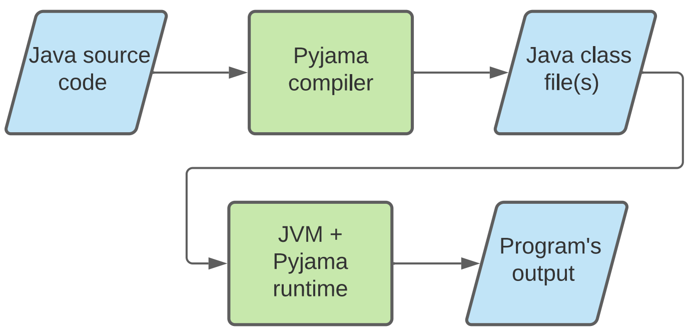
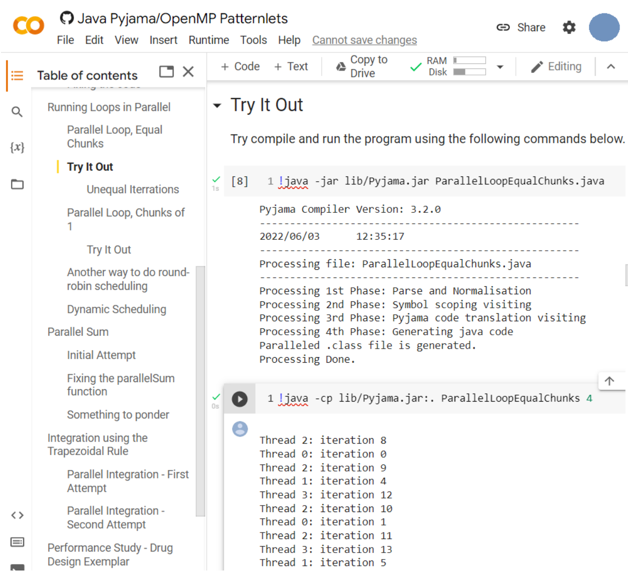


Figure 1: Compiling and running a Java program with Pyjama.

This Pyjama-based notebook was used in the fall 2021 semester to introduce parallel programming in Westfield State University's CS2 and Data Structures (DS) classes. The notebook was deployed in the Google Colab environment which allows students to run and experiment with the example programs just by opening the notebook in a web browser without requiring any setup. When students run examples in the notebook, Google Colab allocates a backend with two virtual CPUs which is sufficient to allow students to conduct hands-on exercises and explore asynchrony, concurrency/dependency, and performance – three out of the four pervasive PDC concepts listed in the TCPP PDC Curriculum [12]. A screenshot of a student's view of the notebook is shown in Figure 2.

In the DS class, the patternlets material in the notebook was accompanied by an assignment in which students write their own program in Java and answer questions related to parallelism on topics such as speed-ups, slow-downs,

Figure 2: Screenshot of the Pyjama Notebook Running in Google Colab.



and race conditions. For the assignment, students were provided with a Maven project template which was set up to perform the Pyjama compilation step on the source code and launch the program in the Pyjama runtime. Because Maven projects are supported in most Java IDEs, it is straightforward for students to open the assignment project in an Integrated Development Environment (IDE) so that they can edit, compile, and run their assignment code in the IDE.

Even though the majority of students, based on class discussions, seem to master the concepts presented in the notebook fairly well, most required a significant amount of support in completing the programming assignment. Students particularly needed help in two areas: understanding Pyjama compiler error messages and debugging issues in their programs.

Pyjama is built on top of an open source Java parser, so the error messages are not user-friendly. This problem was exacerbated by the fact that the Pyjama compiler always returns an error code of zero, so Maven and the IDE did not detect compilation failures. Even after this issue was addressed by modifying the Pyjama source code to return non-zero error codes, it was still very difficult for students to understand and fix syntax errors in their program. The students were not only new to the Pyjama directives, but also confused by error messages from Pyjama. This was further confounded by additional spurious syntax error messages due to Pyjama's lack of support for modern Java syntax. Pyjama's Java parser only supports Java language version 5, so students receive unexpected syntax errors if they use valid Java language constructs from newer versions of the Java language.

Debugging Pyjama programs also raises some new challenges. Even though students are able to use an IDE to open, edit, compile, and run the Maven project, there is no debugging support implemented for Pyjama programs. To debug a Java program with Pyjama directives, one has to use the Pyjama-generated Java source code. This generated code is not, however, intended for human consumption — it is long and hard to understand. An example of Pyjama generated code can be seen in Listing 2. The students in my classes were frustrated by the experience and ended up either asking for help for each error or sometimes giving up altogether.

Listing 2: Generated Pyjama code for the parallel loop example

```java
class ParallelLoopEqualChunks {
  public static void main(String[] args) {
    if (args.length >= 1) {
      int numThreads = Integer.parseInt(args[0]);
      Pyjama.omp_set_num_threads(numThreads);
    }

    /*OpenMP Parallel region (#0) -- START */
    InternalControlVariables icv_previous__OMP_ParallelRegion_0 =
    PjRuntime.getCurrentThreadICV();
    InternalControlVariables icv__OMP_ParallelRegion_0 =
    PjRuntime.inheritICV(icv_previous__OMP_ParallelRegion_0);
    int _threadNum__OMP_ParallelRegion_0 =
    icv__OMP_ParallelRegion_0.nthreads_var.get(
    icv__OMP_ParallelRegion_0.levels_var);
    _OMP_ParallelRegion_0 _OMP_ParallelRegion_0_in =
    new _OMP_ParallelRegion_0(
    _threadNum__OMP_ParallelRegion_0, icv__OMP_ParallelRegion_0);
    _OMP_ParallelRegion_0_in.runParallelCode();
    PjRuntime.recoverParentICV(icv_previous__OMP_ParallelRegion_0);
    RuntimeException OMP_ee_0 = (RuntimeException)
    _OMP_ParallelRegion_0_in
    .OMP_CurrentParallelRegionExceptionSlot.get();
    if (OMP_ee_0 != null) {
      throw OMP_ee_0;
    }
    /*OpenMP Parallel region (#0) -- END */
  }

  // 120 lines of generated inner class code is omitted
}
```

4.2 Method 2: Teaching with Parallel Stream-based Notebooks

Due to these challenges with Pyjama, an alternate notebook was developed covering the same set of patternlets and examples, but using Java parallel streams instead. I initially hesitated to use this material because students were not familiar with functional programming. The students had, however, been introduced to Java lambda expressions when working with GUI and event-driven programming. After some consideration, the new parallel stream notebook was offered as an alternative to the Pyjama notebook in spring 2022. A simple version of the parallel loop example is shown in Listing 3. This simplified example will use the default thread pool, which will by default have the same number of threads as there are cores. To be able to specify the number of threads to use, a custom thread pool is needed. The parallel stream version of a *for*-loop with the custom `ForkJoinPool` is shown in Listing 4.

Listing 3: Parallel stream example

```java
public class ParallelLoop {
  static final int REPS = 16;

  public static void main(String[] args) throws Exception {
    IntStream.range(0, REPS).parallel().forEach(i -> {
      String name = Thread.currentThread().getName();
      System.out.println("Thread " + name + ": iteration " + i);
    });
    System.out.println("Done.");

  }
}
```

Listing 4: Parallel stream with a custom pool example

```java
public class ParallelLoopCustomPool {
  static final int REPS = 16;
  public static void main(String[] args) throws Exception {
    // check and parse arguments
    int numThreads = Runtime.getRuntime().availableProcessors();
    if (args.length > 1) {
      numThreads = Integer.parseInt(args[0]);
    }

    ForkJoinPool customPool = new ForkJoinPool(numThreads);
    customPool.submit(() ->
      IntStream.range(0, REPS).parallel().forEach(i -> {
        String name = Thread.currentThread().getName();
        System.out.println("Thread " + name + ": iteration " + i);
      })
    ).get(); // wait until the stream operations are done.

    System.out.println("Done.");
  }
}
```

In the spring 2022 semester, two sets of material (both the Pyjama and parallel stream notebooks) were used in our DS class to introduce students to parallel programming in Java. In this small class (with only six students), the Pyjama-based material was presented first and the parallel stream material was presented afterwards as an alternative. Half of the students chose to complete their assignment using the parallel stream approach, and the other half chose Pyjama for their assignment. Once again, the students using Pyjama required a lot of support. The students using the parallel stream approach, however, were better able to complete the assignment independently.

4.3 Results and Observations

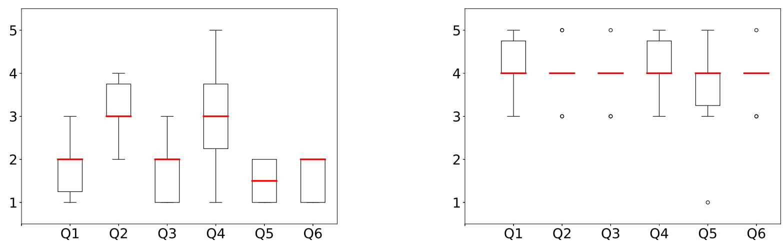
In both the fall 2021 and spring 2022 semesters, the presentation of material using patternlet and interactive Jupyter notebooks in Google Colab allowed students to appreciate the overhead, speed-up, and race-condition that resulted from having multiple threads in the patternlet examples. The asynchrony during the concurrent execution of the parallel patternlets can be easily seen by running the example programs multiple times. Students were able to both modify the examples directly in the notebook and obtain hands-on experience implementing different synchronization mechanisms to eliminate the race condition.

Both formal and informal student feedback on the material was generally positive or neutral. An anonymous survey was done before and after parallel programming was covered in class. The questions in the survey are shown in Figure 3 with responses on a 1–5 Likert-scale.

Figure 3: Survey questions

1. I understand how I can speed up a program using parallelism.
2. I think it is important to be able to easily experiment in order to learn how to speed up a program using parallelism.
3. I understand how I can handle race conditions in a concurrent program.
4. I think it would be useful to see race conditions in practice in order to learn how to correctly handle them.
5. I am able to add parallelism into a Java program and obtain the expected speed up.
6. I am able to add parallelism into a Java program and maintain correctness.

In the DS courses, the majority of students responded to the survey; their responses are shown in Figure 4. Because the parallel stream material was deployed late, the survey did not contain a question to differentiate between students that used the Pyjama-based material and students that used the parallel stream based material.



(a) Before (b) After

Figure 4: Survey questions and results from DS course offerings in the fall 2021 and spring 2022 semesters ($n = 10$).

5 Conclusion and Future Work

Pyjama-based teaching materials can be used to introduce shared-memory parallel programming with Java patternlet examples that use OpenMP-like directives. The Pyjama compiler and runtime allows for the use of simple directives to achieve simple task-level parallelism. Using Pyjama for more complicated tasks, such as a parallel programming assignment, is somewhat difficult due to Pyjama's limitations, including a lack of support for modern java syntax, confusing error messages, and the absence of source-level debugging.

The difference in student performance in the parallel programming assignment after switching teaching materials to a parallel stream based notebook highlighted the importance of using constructs that are integrated in the Java language and have strong community support. Ongoing maintenance will still be required to keep the parallel stream based notebook current with changes in the Java platform. A new set of Java patternlet examples based on the new virtual thread feature [13] (currently available for preview in JDK 19) should also be developed. With upcoming support for structured concurrency in Java [4], it will be possible to have a simple Java patternlet for parallel loop without having to resort to more complicated structures such as creating a thread pool. At this time, Java's support for structured concurrency is still in incubator mode and only available through the Loom project, with a proposed target of JDK 19. With these more modern concurrency structures, teaching parallel programming in Java may well be easier in coming years.

References

[1] Joel C. Adams. “Patternlets - A teaching tool for introducing students to parallel design patterns”. In: *J. Parallel Distributed Comput.* 105 (2017), pp. 31–41. DOI: 10.1016/j.jpdc.2017.01.008. URL: https://doi.org/10.1016/j.jpdc.2017.01.008.

[2] Joel C. Adams et al. *CS in Parallel: Parallel Computing in the Computer Science Curriculum.* https://csinparallel.org/. Accessed: 2023-05-30.

[3] Joel C. Adams et al. “Teaching Parallel and Distributed Computing in the Time of COVID”. In: *SIGCSE ’21: The 52nd ACM Technical Symposium on Computer Science Education, Virtual Event, USA, March 13-20, 2021*. Ed. by Mark Sherriff et al. ACM, 2021, p. 1358. DOI: 10.1145/3408877.3439514. URL: https://doi.org/10.1145/3408877.3439514.

[4] Alan Bateman and Ron Pressler. *JEP 428: Structured Concurrency (Incubator).* https://openjdk.java.net/jeps/428. Accessed: 2022-05-29.

[5] ABET Computing Accreditation Commission. *Criteria for Accrediting Computing Programs.* https://www.abet.org/wp-content/uploads/2018/02/C001-18-19-CAC-Criteria-Version-2.0-updated-02-12-18.pdf. Accessed: 2022-05-27.

[6] E. Gamma et al. *Design patterns: elements of reusable object-oriented software.* Addison-Wesley Professional, 1995.

[7] Sheikh Ghafoor and Mike Rogers. *Plugged Java iPDC Modules.* https://www.csc.tntech.edu/pdcincs/index.php/ipdc-modules/. Accessed: 2022-05-30.

[8] Dan Grossman. “Fork-join Parallelism with a Data-Structures Focus”. In: *Topics in Parallel and Distrib. Comput: Introducing Concurrency in Undergraduate Courses*. Ed. by Prasad et al. M. Kaufmann, 2015.

[9] Kurt Keutzer and Tim Mattson. *Our Pattern Language: A Design Pattern Language for Engineering (Parallel) Software.* https://patterns.eecs.berkeley.edu/?page_id=98. Accessed: 2022-05-30.

[10] Ruth Kurniawati. “Teaching Parallel Programming with Java and Pyjama”. In: *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2*. SIGCSE 2022. Providence, RI, USA: Association for Computing Machinery, 2022, p. 1109. ISBN: 9781450390712. DOI: 10.1145/3478432.3499115. URL: https://doi.org/10.1145/3478432.3499115.

[11] Timothy G. Mattson, Beverly A. Sanders, and Berna Massingill. *Patterns for parallel programming*. Boston: Addison-Wesley, 2005. ISBN: 9780321228116.

[12] Sushil K. Prasad et al. *NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing - Core Topics for Undergraduates (Version II-beta)*. Posted: November 2020, Accessed: 2022-05-27. 2020. URL: %5Curl%7Bhttps://tcpp.cs.gsu.edu/curriculum%7D.

[13] Ron Pressler and Alan Bateman. *JEP 425: Virtual Threads (Preview)*. https://openjdk.java.net/jeps/425. Accessed: 2022-05-29.

[14] Apan Qasem, David Bunde, and Phil Schielke. *ToUCH: Teaching Undergraduates Collaborative and Heterogeneous Computing*. https://touch.cs.txstate.edu/. Accessed: 2022-05-30.

[15] Vikas, Nasser Giacaman, and Oliver Sinnen. "Pyjama: OpenMP-like Implementation for Java, with GUI Extensions". In: *PMAM*. Shenzhen, Guangdong, China: ACM, 2013. ISBN: 9781450319089. DOI: 10.1145/2442992.2442997. URL: https://doi.org/10.1145/2442992.2442997.

Integrating Data Privacy Principles into Product Design: Teaching "Privacy by Design" to Application Developers and Data Scientists*

Susan S. Conrad
School of Technology and Innovation
Marymount University
Arlington, VA 22207
sconrad@marymount.edu

Abstract

Data privacy and data security are terms often used interchangeably but strong data security does **not** guarantee data privacy. Data privacy focuses on protecting the rights of an individual to control access and keep personal data private whereas data security focuses on keeping data safe from threats and vulnerabilities. With the proliferation of online data systems, users often provide their information online and have no idea how the data will be protected or shared. New laws dictate that privacy online begins with integrating privacy measures in the design phase of IT ecosystem products and services however few system developers have the training to understand how this is accomplished. This paper discusses the privacy-by-design approach to system design and offers suggestions about how this data must be protected throughout the data lifecycle. It looks at how the NIST privacy framework and enhanced security techniques can be applied to the collection, processing, disclosure, retention, and deletion of personal data and discusses strategies for educators and developers to add privacy provisions to their products

1 Introduction

Data Privacy is far more than securing the confidentiality, integrity, and availability (the CIA triad) of data. Rather data privacy strives to allow users to control their data, limiting how it can be collected, stored, and used by others. Security is the first step to protecting the privacy of data, but the triad does not always keep it from getting into the wrong person's hands or being misused. Most of us will occasionally share our name and some personal information with someone we meet, after we have vetted them. However, when we provide information online, there is no vetting with people who have access to that data. Once we enter that information, we have no idea who may see the data or how the data will be used. Even with vague, lengthy, and complex online privacy notifications, users are still not sure what will happen with the personal data they are providing. Misuse of this data can lead to a loss of privacy which can have significant consequences, such as financial loss, identity loss, embarrassment, physical loss, reputation ruin, and even personal safety threats.

Privacy is not a right protected by the U.S. Constitution, except possibly by the 4^{th} Amendment, which protects the right of people to be "secure in their persons, houses, papers, and effects, against unreasonable searches and seizures" [14]. There are sector specific privacy laws such as the Health Information Portability Accountability Act (HIPAA, 1996) for health records, Gramm-Leach-Bliley (GLB, 1999) for financial data; Fair Credit Reporting Act (FCRA, 1970) for credit information, Family Education and Rights Privacy Act (FERPA, 1974) for student academic records; and Children's Online Privacy and Protection Act (COPPA, 1998) for children under 13. However, these niche laws seek only to protect specific data and not an individual's personal identity data. To address this privacy protection gap, more than 160 privacy laws were proposed in 2021, encompassing 38 states and numerous federal proposals [15]. This patchwork of state and national laws differs in terms of regulations, penalties and scope making it challenging for organizations to remain privacy compliant.

One provision found in many of the laws being proposed, as well as passed legislation, requires developers (often termed controllers) to follow the principle of "Privacy by Design" during a product inception and development. In essence, both data privacy and data security strategies must be considered when collecting, processing, storing, transporting, and disclosing personally identifiable information (PII). See, for example, the European Union's General Data Protection Regulation (GDPR), California Consumer Protection Act, Colorado Protection Act, Virginia Consumer Data Protection Act and the federally proposed American Privacy and Protection Act have included privacy by design provisions in their legislation [16]. Data privacy strategies must be considered during the product inception and development not only to secure the data but also to ensure privacy of the data and stay legally compliant. This paper will offer educators privacy strategies that protect data during IT ecosystem development lifecycle and incorporates a minimalist approach to data collection.

In addition, special privacy protections are afforded to children under 13 through the Children's Online Privacy Protection Regulations (COPPA, 1998) designed to keep kids safe online. This national law requires verifiable parental consent for children under the age of 13 for online services to include streaming, websites and mobile applications. The law also puts restrictions on the types and methods to market to children under the age of 13. This places a significant obligation on companies who create online games, mobile apps, or social media platforms to be sure that parents have clearly consented to the collection of data for children under 13.

2 Background

The concept of "privacy by design" simply means that with the inception of a product, privacy measures as well as security measures should be integrated into the development of new digital systems and products to ensure privacy of the data [2]. One key practice, that of data minimization, asks the question "what data is absolutely necessary" to collect for processing [3]. Adhering to the concept of data minimization requires developers to collect only what data is necessary for processing, nothing more. This often puts developers at odds with marketing departments looking for as much data as possible to target advertising or even sell the data to 3rd party data aggregators.

According the NIST Cybersecurity Framework, secure coding using the most up-to-date techniques is essential for protecting the data collected, stored, and transported [11]. For example, applying end-to-end encryption by encrypting data when collected, storing the data in an encrypted format, and transporting the data in an encrypted form, will help protect the data in case of a data breach. Utilizing highly sophisticated encryption methods will further ensure data security. Implementing intrusion detection systems and access control measures limits availability of data to unauthorized individuals. Automated audit logs to search for anomalies within the network, written policies and procedures, documented testing methodologies and user training are just a few of the security measures needed to protect data [8]. However, even with all these protections, data still is vulnerable to privacy leaks.

The question facing developers is not when to incorporate data privacy measures into IT products but rather how. Data security and data privacy are often discussed interchangeably but they are not the same and must be treated separately. Data security alone does not guarantee data privacy. Privacy and security are highly dependent upon one another, but each have distinct considerations. Security is concerned with protecting collected data in a secure manner, while privacy is concerned with protecting the rights of individuals to be "secure in their persons." It is not uncommon for data to be secured and yet leave PII data vulnerable to access. Privacy cannot happen without security but security does not guarantee that the data will be used responsibly [1],[2],[9].

Prioritizing privacy and security when developing applications, is a shift in the product development ecosystem, as the data lifecycle must be considered when there is collection, storage, or sharing of data [7]. Critics claim that adding security into products increases development time and cost. However, the demand for new applications, especially Internet of Things (IOT) devices is staggering with more than 14 billion actively connected IOT devices in 2022 and expected to double in the next three years. Developers feeling great pressure to quickly get a product to market often fail to incorporate security and privacy features into their product. This security omission is a significant security risk with 48% of businesses reporting that they cannot detect IOT security breaches [6].

Developers most likely have been trained in agile development techniques, but they probably have not been trained on the data lifecycle which is data-focused as opposed to a system implementation focus. Thus, integrating data privacy strategies into product design is not customary for their trade. This gap in training is where educators must take the lead to instill data privacy requirements into the design and coding of the IT systems since it is not only easier but also more cost effective [3].

3 PII and the Data Lifecycle

To know what personal information most needs protection, students must be made aware of the concept of "Personally Identifiable Information" (PII). Currently, in the absence of a national privacy law, there is no uniform definition of PII, and several states have passed privacy laws with varying definitions of what data is considered private [14]. In addition, the European Union has adopted a privacy framework, the GDPR, while China, India, South Africa, Australia, Brazil, and numerous other countries have enacted their own privacy laws, each with differing definitions of PII. Some definitions of PII include: data with names and identifying numbers (SSN, Driver's License, Passport number, etc.); biometric data (fingerprints, DNA, iris print, vein print, etc.); online identifier (IP address, RFID tags, cookie identifiers); health data (illnesses, lab results, mental health information); financial data (bank account numbers, mortgage data, insurance data), and certain records about someone (criminal, education, purchasing behavior); and combinations of data.

Generally, PII is any datum or combination of data that can identify a unique individual. Within the collection of PII, a subsection of data is classified as "sensitive data" and includes elements such as: SSN, driver's license number, passport number, Alien registration numbers, financial information, criminal history, biometric and medical records. This data requires special handling and security measures such as encryption, pseudonymity, and consent [17][13].

3.1 The Data Lifecycle

To understand the requirements to protect privacy, one must review the data lifecycle from a privacy perspective. From collection to destruction, data has a predefined lifecycle [1]. Often the idea of destroying data is omitted from discussions, but under new laws that require organizations to honor requests from users to delete their personal data, strategies, and policies for when and how to delete data must be integrated into a system's design. The data lifecycle describes the flow of information through an organization. Designers must consider how end-to-end data privacy strategies will be incorporated into each of these phases: collection, processing, retention, disclosure, and destruction.

Collection The first phase of the data lifecycle begins with capturing data from a user. When an application is collecting data online, a user logs onto the internet through an internet service provider (ISP); opens a web browser and connects to a webpage. Once on the webpage the user will enter data, often PII, and that information will flow from the user's computer through the browser, the internet, to an application for processing and storage. The data has encountered five mediums throughout this process. Here data can be stored by both the application requesting the data but also the browser. Many websites use cookies to streamline surfing on the internet, the cookies collect data about the user, the user's device, meta data and online behaviors [10]. Cookies vary in terms of their usage, collection methods, and data storage techniques, and provide security risks. In some countries, users must physically consent to the cookie and if they select not to allow cookies, the user should not be prevented from viewing content on the website. Obtaining user consent during the data collection phases is core to IT system design requirements.

The collection phase of the data lifecycle is the most important process to secure because concerns about malware and cyber intrusions are most evident in this phase due to potential vulnerabilities in the application, the internet connection, the browser or even the user's own device. The GDPR identifies the data collector as the "data controller" and as the most responsible party for protecting privacy and access to an individual's data. For example, if Company X uses a cloud-based payroll system (Company Y) to process payroll, Company X is the data controller and Company Y is the data processor. It is during the data collection phase that the data collector must get the consent of the user. The consent must state the purpose for collecting the data and reveal who will have access to the data. Responsibility for the data, even though the processing is conducted by Company Y, falls on the controller, Company X [2][18].

Processing The processing phase refers to any type of data manipulation, whether manual or through automation. It includes the structuring of collected data. The GDPR refers to providers who perform this activity for a data controller as a "data processor" [18]. For example, if an organization uses a cloud-based service to processes personal data about customers, such as individual order requests, that cloud service provider is the data processor.

Retention Retaining data is a key part of the lifecycle. Storing personal data in a secure facility and ensure the integrity of that data is critical to the security of personal information. Data storage is generally provided by a data processor.

Disclosure The disclosure phase reveals with whom the data will be shared. An organization may desire to share personal data with third parties for advertising or marketing purposes. For states and countries who have passed data privacy laws, personal data collected by a data collector can be shared with third parties for advertising or marketing activities only if the user has consented to this activity. The consent document initially presented to the user must identify who can have access to this personal data.

Destruction Privacy advocates suggest that data should be kept for only the time necessary to complete the process or required by law following a strategy of data minimization. Various state laws and the GDPR are now requiring that consent notifications include how long data will be retained. Users may request that their data be deleted, and that the data be removed from all accessible online websites. This means that data collectors and processors must be aware of who has access to specific PII and how it can be deleted from online access [4].

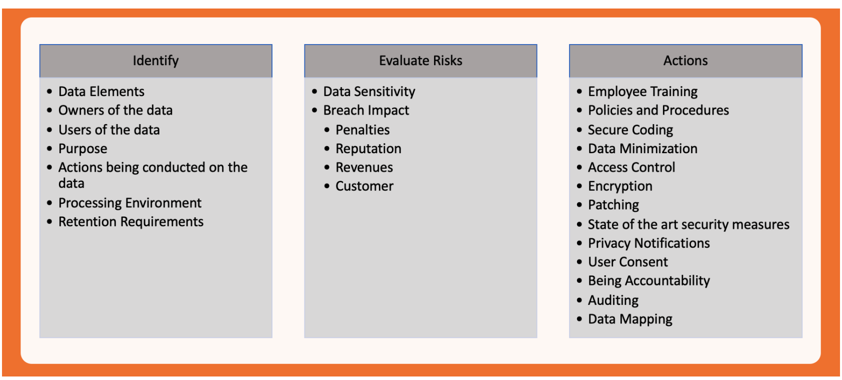
3.2 Data Protection Methods

Since privacy laws with steep penalties require PII and sensitive personal data to be secured and protected, it is essential that the system developer along with a privacy expert design systems which comply with laws and protect personal data. The first step to protecting data is to understand what qualifies as PII and sensitive data according to the prevailing privacy laws, so appropriate security can be integrated into the design.

Large-scale development projects often employ a service-oriented architecture (SOA) or client-server architecture for obtaining data which consists of front-end and back-end systems. A common simplified scenario involving data collection is for a user to enter data through a front-end web browser (client) for processing with a webserver who communicates with the back-end server to process the request and store the data in a database. During this initial contact with the web browser, the user's privacy rights must be considered.

Since the GDPR is a model that other countries and US states have copied, this paper will highlight the privacy notification requirements under GDPR. Per the GDPR, when users go online, the website must tell the user whether the site collects cookies. Users must be given an option to either allow the cookies or not allow them. If the user is asked to provide personal data, even an email address, the site must provide a "privacy notification" to the user. The privacy notification will tell the user: (1) Company privacy policies; (2) What data is being collected; (3) Purpose for the data collection (4) Whom it is shared with; (4) How long it retained; (5) Consent by Opting-in or Opting-out [5]. Figure 1, the Privacy Notification Flow, shows an example how each privacy provision interacts with the front and back end of a web application using SOA software design.

Figure 1: Privacy Notification Flow



3.3

When designing the front-end system, Privacy by Design regulations require developers to consider ways to securely authenticate users in order to verify that only "authorized" people can view the data [1]. If data is collected, the data controller is mandated to provide a privacy notification which details uses of the data and requires the consent of the user, informs the users if the site uses cookies, and allows the user to opt-in or opt-out for any sharing of the data with

3rd parties. Consent must be an action provided by the user to either select "Yes" or "No," Back-end systems must incorporate mechanisms to limit who can access the data and allow access only for the purposes specified in the consent notification. Methods to authenticate users and only allow those authenticated to access the data is central to protecting the privacy of an individual's data. While data security focuses on protecting the data from cyber-attacks such as malware, ransomware, man-in-the-middle, advanced persistent threats, denial of service, and social engineering; data privacy is all about protecting against access of the data by unauthorized individuals/organizations [12]. Privacy demands implementation of all the security methods used to protect data with additional measures for protecting privacy [8].

3.4 Teaching Privacy by Design

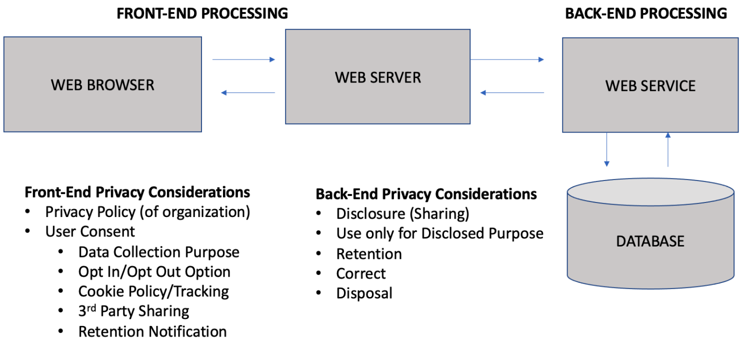
The US National Institute of Standards and Technology (NIST) has long had a well adopted Cybersecurity Framework, but recently in 2020 they added a Privacy Framework: A tool for improving Privacy through Risk Management. The NIST Privacy Framework utilizes a risk management approach to help organizations prioritize potential privacy risks associated with their products and services. The framework is a cross-organizational set of processes to help manage privacy risks and communicate privacy risks with internal and external stakeholders across the data processing ecosystem [3].

The framework encourages a privacy-by-design approach to minimize risks and build privacy protections in the initial design. The NIST privacy framework identifies three privacy-by-design objectives: (1) predictability - privacy protections are measurable; (2) manageability - the system's ability to manage data from collection to deletion; and (3) disassociability - segregating personal data from a user's transactional data. The framework includes a template for entities to map data collected in the design phase of an IT project and rate the privacy risks by identifying: the sensitivity of the data (PII); who has access to the data; data retention requirements; and intended purposes [9]. Figure 2, the Privacy Action Model shows the process of compartmentalizing privacy risk by identifying the data and characteristics, identifying the level of sensitivity and risks in case of a breach and lastly the actions that can be taken to protect data privacy.

As the model shows there are specific actions that can be taken to reduce privacy risks within both the system development and the IT lifecycles. These actions include measurable security and privacy procedures that can significantly protect an individual's data and help preserve the individual's control of their own data.

Developing a curriculum that will teach new generations of IT and Data Scientist professionals best privacy practices begins with building awareness and

Figure 2: Privacy Action Model



technical proficiencies for protecting data privacy. The following list provides educational objectives for adopting a privacy/security approach to data.

Privacy Top Teaching Tips Mitigating risks begins with teaching about PII and sensitive data emphasizing the importance of protecting data from unauthorized use and corruption. Educators can modify how they teach the system develop lifecycle to include privacy protection in the design and development phases of the projects involving data by integrating some the action steps defined in Figure 2. The following text provides some specific suggestions for educators and developers as to how privacy protection techniques can be incorporated in the development and management of IT ecosystems.

(1) Code with up-to-date coding protocols to ensure no backdoors are open for malware to penetrate the system. (2) Test code for vulnerabilities and document test results to create an auditable record that the code is the latest version and error-free. (3) Incorporate end-to-end encryption when collecting, transferring, and storing data (4) Embed privacy notifications and user consent functionality in the design and development of front-end web services or mobile apps (5) Educate students about privacy laws, NIST privacy framework and system development methodologies which include privacy provisions. (6) Use pre-coded modules and APIs from standard coding libraries and frameworks. (7) Code opt-in and opt-out options with the user consent and data sharing defaulting to opt-out. (8) Integrate secure user authentication code or plug-ins when creating accounts for users. (9) Analyze APIs to verify they use encryption to transfer data and that they are coded with the most current versions of the specific coding language. (10) Teach students about organizational privacy policies and the privacy notification and consent process. (11) Adopt a data

minimization approach to system design. (12) Develop privacy training for all people within an organization. (13) Teach students what data is considered PII and sensitive data.

4 Conclusion

Protecting privacy begins during the design phase of a digital product. The ramifications of not considering privacy can result in losses to both individual users and organizations. Data has many uses and collecting it, whether to target marketing to consumers or to fuel machine learning algorithms, has financial value. A review of the current laws protecting privacy, noting that the legal landscaping is quickly changing, indicates that national privacy regulations are inevitable with the GDPR lighting the way.

Furthermore, protecting privacy should be more than a legal requirement but an ethical requirement as failure to do so leaves people vulnerable to unforeseen consequences. A discussion of what personal data should be protected and its data lifecycle provides insight to the phases through which data flows and how failure to protect this data can lead to exposure and undesired consequences. The interrelationship between data security and data privacy requires identifying where data security alone is insufficient to protect data privacy. It is important for educators to integrate into existing computer science and information technology classes, lessons that will inform students about privacy issues and offer ways to mitigate privacy risks. Given the importance that privacy will continue to play in the development and management of IT ecosystems, academic institutions must incorporate courses dedicated to the highly complex topic of privacy and the rights of the individual to protect and control their own data.

References

[1] Travis Breaux. *An Introduction to Privacy for Technology Professionals.* International Association of Privacy Professionals, 2020.

[2] Susan S. Conrad. Protecting personal information and data privacy: what students need to know. *J. Computer Sci*, 35(3):77–86, October 2019.

[3] R. Jason Cronk. *IAPP-Strategic Privacy by Design.* International Association of Privacy Professionals, 2022.

[4] Deloitte. Data Destruction Survey Report, 2020.

[5] GDPR. What are the gdpr consent requirements? 2019. Retrieved June 12, 2022, `https://gdpr.eu/gdpr-consent-requirements/`.

[6] Bojan Jovanovic. Internet of things statistics for 2022 - taking things apart. *Dataprot.*, 2022. Retrieved June 11, 2022, `https://dataprot.net/statistics/iot-statistics/`.

[7] Rebecca Kern. The genesis of "privacy by design.". *POLITICO*. Retrieved June 10, 2022, `https://politi.co/3NzchId`.

[8] Richard Kissel. Security considerations in the system development life cycle. *NIST Special Publication 800-64 Revision 2*, page 68.

[9] Nist.gov. Privacy framework. NIST. 2020. Retrieved June 10, 2022, `https://www.nist.gov/privacy-framework/privacy-framework`.

[10] Karen Lawrence Oqvist and Filip Johnssen. *Hands On Guide to GDPR Compliance - Privacy by Design*. International Association of Privacy Professionals, 2018.

[11] Ron Ross, Victoria Pillitteri, and Kelley Dempsey. Assessing enhanced security requirements for controlled unclassified information. *National Institute of Standards and Technology*, 2022. `https://www.doi.org/10.6028/NIST.SP.800-172A`.

[12] Secure coding. *WhiteHat Security Glossary*. Retrieved June 11, 2022. `https://www.whitehatsec.com/glossary/content/secure-coding`.

[13] Homeland Security. What is personally identifiable information? | homeland security.

[14] Daniel Solove and Paul M. Schwartz. *Privacy Law Fundamentals 6th Edition*. International Association of Privacy Professionals, 2022.

[15] National Conference of State Legislators. 2021 consumer data privacy legislation. 2021. Retrieved June 10, 2022, `https://www.ncsl.org/research/telecommunications-and-information-technology/2021-consumer-data-privacy-legislation.aspx`.

[16] European Union. Privacy by design. 2018. Retrieved June 10, 2022, `https://gdpr-info.eu/issues/privacy-by-design/`.

[17] European Union. Processing. *General Data Protection Regulation (GDPR).*, 2018. Retrieved June 9, 2022, `https://gdpr-info.eu/issues/processing/`.

[18] European Union. What constitutes data processing? *European Commission - European Commission*, 2022. Retrieved June 9, 2022, `https://ec.europa.eu/info/law/law-topic/data-protection/reform/what-constitutes-data-processing_en`.

Introducing Computer Science to Education Faculty – Teach the Teacher Educators*

Jiang Li, Marisel Torres-Crespo, Jennifer Cuddapah, Ann Stewart
Department of Computer Science and Information Technology
Department of Education
Hood College
Frederick, MD 21701
lij@hood.edu

Abstract

This paper describes a collaborative project that was conducted to promote the Computer Science (CS) and Computation Thinking (CT) Education among Education Faculties. 9 Education Faculties from three local and regional schools participated in a learning experience designed to address the K-12 Computer Science Framework[1] and Maryland's K-12 Computer Science Standards. The collaboration was designed to develop CS/CT knowledge, understanding, skills and application among Education Faculties who work to prepare teachers. The plan is to build the capacity of faculty to integrate CS and CT into several preservice courses as well as build into the coursework sequence meaningful experiences that expose future teachers to CS and CT knowledge and skills so that they can in turn incorporate these into their own lesson plans. Project evaluation included formative and summative assessments to examine changes in content and pedagogical knowledge. Phase one feedback from the participants shows positive result.

1 Introduction

Computer Science develops students' computational and critical thinking skills and shows them how to create, not simply use, new technologies. The knowledge and skills learned from studying computer science prepare students for careers in a variety of sectors. According to a study conducted by code.org and Computer Science Teachers Association (CSTA)[2], in the past few years, more than 40 states have enacted one or more policies to make computer science education fundamental.

Systemic change, teacher engagement and development of teaching resources are required to bring CS to K-12[3]. Recommendations have been made to create a successful CS education program, such as ensuring teachers are prepared and supported, create continuity and coherence around learning progressions, make participation equitable etc.[5].

Hood College's Computational Thinking Partnership: Teach the Teacher Educators (TTE) is the third round of a collaborative project involving the Departments of Computer Science and Information Technology and Education, Frederick County Public Schools (FCPS) and Frederick Community College (FCC). The aim of the third phase is to develop Hood College's as well as local community college teacher educators in their knowledge, understanding, skills, and application of the K-12 Computer Science Framework and Maryland's K-12 Computer Science Standards. Teacher education faculty participants from Hood as well as some education faculty from two community colleges from which Hood draws many transfer students (Frederick Community College and Montgomery Community College) studied foundational principles of computer science and computational thinking that will be incorporated into their preservice courses to prepare future teachers.

2 Implementation

2.1 Background

The collaboration, Hood College Computational Thinking Partnership (CTP) project, was initiated in late spring/early summer of 2019 with a vision to facilitate participants' learning about and application of foundational principles of computer science and computational thinking into K-8 STEM curriculum and teaching. In the first round of the project in summer/fall 2019, 23 diverse educators participated (12 Hood preservice students, 1 Hood faculty member, 5 FCC preservice students, and 5 FCPS classroom teachers) in workshops and mini-grants. The second round of project began with a similar focus in Spring 2020 where we enrolled 15 more educator participants (10 Hood preservice students and 5 FCPS classroom teachers) and delivered a similar professional learning experience both face to face and then virtually when COVID necessitated quarantine. Robot kits and curriculum were built collaboratively with 14 additional participants (2 Hood preservice students and 12 FCPS classroom teachers). The results about initial learnings, activities, accomplishments from the first two rounds of project were excellent and were reported[7].

Hood College is committed to preparing teachers to work effectively with diverse student populations. Although females make up over half of the college graduates in this country, and the vast majority of teacher educators and teaching population, only 20% of computing field graduates are women. This is about the same percentage of underrepresented minority students according to National Science Foundation, National Center for Science and Engineering Statistics 2017. Authors have written about the lower confidence levels among women about their computing abilities[4, 6, 8, 9]. Since the teacher educators at Hood are all female and have little to no background in computing fields and computational thinking, they are likely to feel inadequate about working with their preservice teachers (most of whom are also female) on topics and skills about which they themselves do not know or understand. In turn, the students with whom our graduates will teach are a diverse group, entering a work world filled with opportunities which necessitate computer science and computational thinking understanding and application. Education faculty should learn about CS and CT and incorporate it into the preservice education courses so that future teachers integrate this knowledge and skill development into the classes they teach.

2.2 Goals and Objectives

To build on the experiences from CTP 2019-2021, round three of the project was proposed to develop the capacity of our teacher education faculty who work to prepare teachers. It is all about "Teach the Teacher Educators." (TTE) To be more specific, the following are the goals and objectives of the proposed project, which also reflects the two phases of the project.

To facilitate the preservice education faculty participants' learning about the principles and practices of computer science and computational thinking.

To equip participants to incorporate computational thinking, CS standards-based activities and assessments into their preservice courses.

2.3 Recruiting

We continued our partnership with Frederick Community College and added connection with another local community college, Montgomery College, to invite their teacher education faculty to participate.

The CTP/TTE project utilized multiple recruitment means and incentives. The project director actively worked with each partner and reached out to the potential participants in several ways.

Marketing and recruitment materials highlighted the importance of computer science in education and this CTP/TTE educational opportunity. Participants were offered a stipend for participation. Additionally, participants had the opportunity to receive mini-grants to implement the projects they designed in the CTP in their classrooms. The online Blackboard forum kept participants connected to one another and the faculty and consultants in between sessions.

2.4 Format and Content

In phase one of the project, participants received professional development addressing the K-12 Computer Science Framework and Maryland's K-12 Computer Science Standards through ten sessions that were offered throughout the fall semester of 2021, eight in face-to-face format, and two in online format. Each session was held during Hood's Common Hour, a designated time when all faculty are available and no classes are scheduled.

Five Hood CS and the two Education faculty who have been involved in the first two rounds of the project designed and delivered the professional development around the 5 core concepts from the framework – Computing Systems, Networks and the Internet, Data and Analysis, Algorithms and Programming, and Impacts of Computing. During each professional development (PD) session, the content specifics of CS and CT were conveyed and there were times for hands-on application and QA and discussion around misconceptions. Then, the Education Specialist facilitated discussion and application of the ideas and skills learned to the preservice curriculum and existing courses. Participants explored hard/software platforms for CS content delivery, unplugged activities and use open-source sites such as Scratch, Code.org and Khan Academy. At the end of the semester, a planning retreat session was held, where faculty participants made plans to integrate their learning into an upcoming class they teach.

In phase two of the project, participants developed or adapted an assignment/assessment to integrate the CT and CS standards, with the peer and content specialists' support. They identified where in their syllabus they need to make changes to incorporate the assignment/assessment to make explicit the connections to the standards. Participants have been implementing their ideas into their spring or fall class. Mini-grants were rewarded to support the participants carrying out the changes.

A BlackBoard site was created to host resources, module material, discussion forum, and evaluation links. In Fall 2022, there will be a Gallery Walk and Reflection Session where faculty will share what they did and what they applied in their classes. They will reflect about what they intend to change and incorporate into the next iteration of their class planning.

2.5 Sustainability

To obtain the sustainable goal, the following approaches were conducted: (1) Evaluating development and implementation of CTP through quantitative and qualitative analysis of surveys, observations, and documents; (2) Revising existing pre-service teacher preparation curriculum to integrate CS standards; (3) Maintaining collaboration between FCPS and the CS and Education Departments; and (4) Securing future funding for CS education.

Hood seeks to sustain Hood TTE outcomes by incorporating the learning into the preservice classes. Each class will contain a project/assessment and instructional activities. These will be designed and incorporated initially in the grant year. In subsequent years, the initial experiences will be reflected on, and TTE project will be improved, and adjusted.

3 Results

The Hood TTE Project Evaluator has assessed and reported progress and attainment of the goals and objectives. Project evaluation includes formative and summative assessments to examine acquisition of content and pedagogical knowledge. All faculty participants completed surveys at the beginning and end of the program as well as feedback after each workshop session.

Table 1 shows the pre-survey result. The values in the table indicate the percentage of the faculty numbers on how strongly agree or disagree with the corresponding survey statements.

Table 1: Pre-survey Result

Statement	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
1	78%	11%	11%	0%	0%
2	67%	0%	33%	0%	0%
3	78%	0%	11%	11%	0%
4	56%	11%	11%	11%	11%
5	22%	11%	33%	11%	22%
6	44%	0%	22%	22%	11%
7	33%	0%	33%	11%	22%

Table 1 statement list:

1. I have the knowledge I need to teach CS effectively.
2. I have the skills I need to teach CS effectively.
3. I have the curricular tools and resources I need to teach CS effectively.
4. I have a social network that enables me to teach CS effectively.
5. I can interest my preservice teachers in CS.
6. I can effectively teach my preservice teachers CS.
7. I can assess my preservice teachers' learning and performance with regard to CS.

As mentioned in previous section, during the phase one of the project, 10 professional development (PD) sessions were offered to the participants to facilitate their learning about the principles and practices of computer science and computational thinking. Table 2 shows the average result of the surveys conducted after each session.

Table 2: PD Session Result

Statement	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree
1	2%	4%	12%	48%	34%
2	2%	6%	14%	47%	31%
3	2%	14%	27%	35%	22%
4	2%	4%	12%	31%	52%
5	4%	2%	10%	18%	44%

Table 2 statement list:

1. I have a good understanding of the new concepts presented.
2. I feel comfortable with the technical terminology used in the new concepts presented.
3. I feel comfortable describing the new concepts using grade level appropriate language.
4. I understand why the new concepts are important to know.
5. The hands-on activity(s) helped me to understand the new concepts.

4 Conclusion and Future Work

From the positive data shown in the previous section, the phase one of our project was concluded as a success. Our participants gained domain knowledge and were equipped with curriculum content to teach CS in their classrooms. Our participants learned about the principles and practices of computer science and computational thinking.

In phase two of our project, the participants will incorporate computational thinking, CS standards-based activities and assessments into their preservice courses. Additionally, participants' course products will be shown in their syllabi, directions and rubrics which will be the qualitative data to be evaluated. Project evaluation results will be written in a final report for us to indicate outcomes and analysis. Products will be shared in the peer gallery walk events.

Acknowledgement

This project was sponsored by Maryland Center for Computing Education (MCCE) Maryland Pre-service Computer Science Teacher Education Program.

References

[1] K–12 computer science framework, 2016. Retrieved from `http://www.k12cs.org/`.

[2] 2018 state of computer science education, 2018. Retrieved from `https://advocacy.code.org/`.

[3] Valerie Barr and Chris Stephenson. Bringing computational thinking to k-12: what is involved and what is the role of the computer science education community? *ACM Inroads*, 2:48–54, March 2011.

[4] S. Beyer, K. Rynes, and S. Haller. Deterrents to women taking computer science courses. *IEEE Technology and Society Magazine*, 23(1):21–28, 2004.

[5] P. Blikstein. Pre-college computer science education: A survey of the field. *Google, LLC*, 2018.

[6] S. Cheryan, S. A. Ziegler, A. K. Montoya, and L. Jiang. Why are some stem fields more gender balanced than others? *Psychological Bulletin*, 143(1), 2016.

[7] Jiang Li, Paulette Shockey, Jennifer Cuddapah, Christy Graybeal, Marisel Torres-Crespo, and Anthony Williams. Introducing computational thinking to pre-service teachers. *The Journal of Computing Sciences in Colleges*, October 2018.

[8] L. J. Sax, J. M. Blaney, K. J. Lehman, S. L. Rodriguez, K. L. George, and C. Zavala. Sense of belonging in computing: The role of introductory courses for women and underrepresented minority students. *Social Sciences*, 7(8):122–145, 2018.

[9] B. C. Wilson. A study of factors promoting success in computer science including gender differences. *Computer Science Education*, 12(1-2):141–164, 2002.

Assessing Student Outcomes Related to Design for ETAC-ABET Accreditation*

Lili Ma and Benito Mendoza
Department of Computer Engineering Technology
CUNY-New York City College of Technology
Brooklyn, NY 11201, USA
LMa@citytech.cuny.edu, BMendoza@citytech.cuny.edu

Abstract

This paper describes our assessment model that evaluates both Program Criteria Indicators and Student Outcome (2) which is related to engineering design, in a sophomore-level laboratory course for ETAC-ABET Accreditation. The Program Criteria Indicators are evaluated using two laboratory assignments. The Student Outcome (2) on students' ability to design systems, components, or processes for well-defined engineering technology problems is evaluated using the final design project. Due to the nature of this course which involves 3D modeling, analysis, and design, both the Program Criteria and Student Outcome (2) fit consistently with the course content and objectives. By making the assessment instruments faculty-driven and incorporating assessments as regular course activities, continuous improvement of students' learning experiences is achieved. The assessment cycles span from traditional in-person to half in-person and half online due to COVID-19 disruption and then to online teaching. The presented assessment results contributed to the recent re-accreditation of our programs.

Keywords: ABET accreditation, assessment, program criteria, student outcomes, CAD

1 Introduction

The Department of Computer Engineering Technology (CET) at CUNY-New York City College of Technology (City Tech) offers a four-year BTech program in CET, whose first two years are an AAS program in Electromechanical Engineering Technology (EMT). Both programs are ABET-accredited and thus abide by the accreditation criteria established by the Engineering Technology Accreditation Commission (ETAC).

In the field of assessment in higher education, defining a sustainable assessment model is essential for accreditation as an element of an institution's process for continuous improvement. Practitioners regularly emphasize two features as key to success in implementing a sustainable assessment model: 1) it must be faculty-driven and 2) it must become part of the curriculum [5]. In this paper, we present a sustainable assessment model that incorporates these two elements.

The ABET criteria for accreditation have two sections: the General Criteria and the Program Criteria. The former applies to all programs accredited by an ABET commission, while the latter is a discipline-specific accreditation criterion. Initially, our model was developed for assessing the Program Criteria, which remain roughly the same for more than six years. Though the 2019-2020 ETAC-ABET requirements changed by no longer requiring direct assessment of Program Criteria, we choose to continue these assessments that serve as evidence for achieving the Program Criteria.

The newly-published General Criteria presents a new Criterion 3 (Student Outcomes) that changed from ten to five Student Outcomes (SOs), among which the Student Outcome (SO) (2) requires evaluating competencies not previously demanded in the EMT program. This newly-added SO (2) evaluates the students' ability to design systems, components, or processes for well-defined engineering technology problems appropriate to the discipline. We believe that direct assessment by the faculty is necessary to provide an objective measure of students' achievement of educational outcomes [23]. Our assessment strategy of SO (2) is also faculty-driven and incorporated into the course activities. Specifically, it relies on a direct assessment method called performance appraisal [20] using a scoring rubric.

In computer engineering, design can be related to specializations such as software systems and computer architecture. However, computer engineering is related to other fields such as robotics, mechatronics, embedded systems, and cyber-physical systems, where physical components and computers are integrated to form complex computer-controlled systems. Computer engineers should be able to design, simulate, implement, and evaluate a complete computing system [9]. Computer-Aided Design (CAD) is a computer tool that engineers use to develop models on computers as mock-ups of real-world physical models. CAD design skills are thus important for computer engineers [3, 11].

Our most recent assessment cycle started in 2014 and concluded in Spring 2021. During this time frame, we used two laboratory exercises and the corresponding student work samples as evidence for the Program Criteria. Assessment of SO (2) is via the final design project, which was recently added in response to this new requirement. Corresponding assessment results and analyses were presented during the recent visit in Spring 2021, demonstrating achievement of the requirements as stipulated in the Program Criteria as well as the recently-added SO (2) on students' design capabilities. Our proposed easy-to-deploy assessment model along with its implementation incorporated assessment as regular course activity. These activities help to achieve the ultimate objective of education, i.e., continuous and constant improvement of students' competencies and learning experiences [6].

The paper is organized into the following sections: assessment model, assessment implementation, assessment results & analyses, online teaching, and conclusions & future work.

2 Assessment Model

A curriculum map [25] is used to reveal the course where the assessment should take place. The curriculum mapping, which maps each SO with each course, provides a visualization mechanism to find where SOs and courses or educational strategies intersect. After examining the curriculum mapping, a sophomore-level, one-credit laboratory course titled "EMT 2480L: Electromechanical Systems Design Laboratory" was selected to carry out the assessment.

The course is structured to have 3 contact hours dedicated to laboratory exercises. It introduces 2D/3D Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE). Industry-leading software such as SOLIDWORKS, AutoCAD, Inventor, and VFEA are used to build mechanical parts and assemblies [17, 12, 19, 27]. CAE techniques are utilized to introduce concepts of mechanics of materials. The course starts from the introduction of fundamental 2D/3D modeling, moves on to simulation using CAE techniques to analyze and test components/products, and ends with a focus on the application of these skills to provide design solutions. Since this course involves computer-aided modeling, simulation, analysis, and design, it serves as a good candidate for assessing both the Program Criteria and SO (2) in one course.

In our assessment model, two lab assignments are used to assess the Program Criteria. These two assignments are referred to as I-Beam and Hub. They appear around the middle of the semester, after introducing fundamental skill-building modules (Fig. 1). The final design project for assessing the SO (2) [4, 26] occurs at the end of the semester.

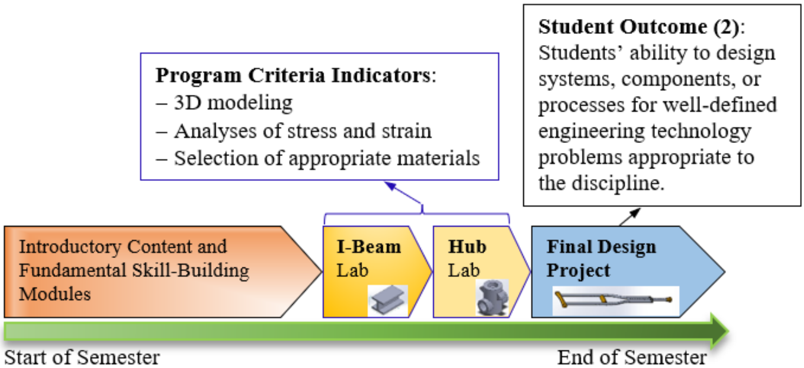


Figure 1: Timeline of three assessment assignments in one course.

2.1 Assessment of Program Criterion Indicators via Two Laboratory Assignments

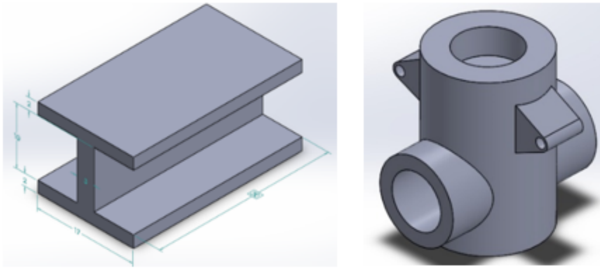
Three indicators are used to evaluate the Program Criteria:

1) Constructing a 3D model of a mechanical part.
2) Using knowledge of statics and strength of materials to determine stresses in a component of an electromechanical system.
3) Using knowledge of engineering materials to select appropriate materials for the construction of a prototype electromechanical system.

Two lab assignments, i.e., linear static analyses of I-Beam and Hub as shown in Fig. 2, were developed/used to evaluate the three indicators mentioned above within one lab assignment. In each lab, students are first asked to create the 3D mechanical part and then use different materials to compute, simulate, and evaluate stress, strain, the factor of safety (FOS), and moments of inertia.

2.2 Assessment of ETAC-ABET SO (2) on Design Capabilities via Final Design Project

CAD skills are usually considered design skills [1, 7]. Thus, the evaluation of students' design capabilities fits well in this course. The SO (2) is evaluated using the final design project, where students are asked to design a product, explore different types of materials while analyzing their properties of stress failure and strength of materials, and practice some simple time management skills. The following rubrics (Table 1) were developed and used [15]:



(a) I-Beam (b) Hub

Figure 2: The two lab assignments for assessing Program Criteria Indicators.

Table 1: Our rubrics for assessment of ETAC-ABET SO (2)

Performance Indicator (PI)	**Excellent 4**	**Good 3**	**Acceptable 2**	**Unacceptable 1**
1. Understand the Design Problem and the Requirements	Clear and complete understanding of design goals and constraints.	Overall sound understanding of the problem and constraints. Does not significantly impair the solution.	Some understanding of the problem. Major deficiencies that will impact the quality of the solution.	Little or no grasp of the problem. Incapable of producing a successful solution.
2. Use Project Management Techniques for Completion	The timeline is clearly defined and developed. The details are comprehensive; it represents a plan with a high probability of project completion.	The timeline illustrates an understanding of individual task requirements, potential bottlenecks identified, and reasonable potential for project success.	The timeline is loosely defined and lacks a clear understanding of time requirements for tasks, and risks incomplete projects.	Lacking a defined timeline.
3. Evaluate Alternative Designs and Options	Final design achieved after reviewing reasonable alternatives. This includes different materials and/or economic advantages.	Alternative approaches are identified to some degree. For example, different materials	Serious deficiencies in exploring and identifying alternative designs.	Only one design is presented.
4. Complete Implementation of Design Process	Quality and focused design process implemented, fully documented, clear qualitative/quantitative criteria for making decisions.	Rational, documented process, measurable criteria for making design decisions.	A vague process was implemented for the design, little record of the process, and poorly defined criteria.	No design process was implemented.

After the definition of the rubrics and performance indicators (PIs), the following documents were developed, adopting well-known practices to define the assessment instruments:

- Assignment Worksheet. The form briefly describes the assignment to use for assessment and selects the PIs to cover.

- Assignment Description. It describes details of goals, requirements, content, and instructions for the students. Most of the time, this document becomes the handout for the students. In addition to the common expectation of "understanding the design requirements" and "finishing the design in time", emphases were placed on highlighting "Project Management Techniques" as well as "Conducting Alternative Designs" to cover all four PIs in the rubrics.

- Content Validity Form. It is a mapping between the PIs and the assignment. It describes specifically how each PI listed in the rubric will be evaluated (e.g., via a question, a calculation, or a drawing) and how the instructor will verify attainment. The Content Validity form helps the assessment process in two ways. At the design stage, it ensures the assignment covers each PI. At the assessment stage, it identifies which part of the assignment to evaluate each PI.

Instructors are allowed to use different projects as long as all PIs can be effectively assessed. A variety of projects were brought into this course (Fig. 3), greatly enriching the course activities as well as challenging the students to handle real-world problems [8, 14].

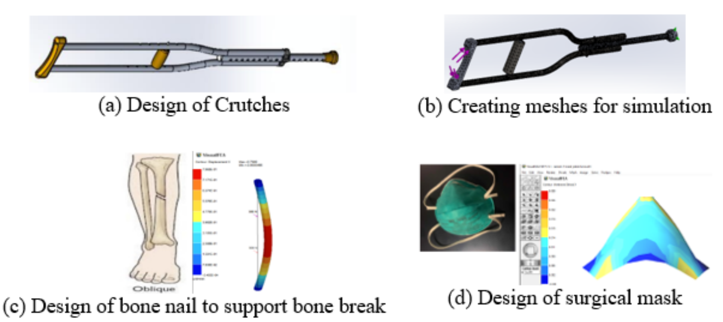


Figure 3: Examples of final design project for assessing ETAC-ABET SO (2).

3 Assessment Implementation

Assessment implementation involves collaboration among several members. The Chair of the department's assessment committee provides continuous guidance about expectations and timely feedback. The course instructors constantly improve course materials, adding well-designed labs and projects. The course coordinator ensures that the assessment is conducted every semester. Three to four sessions of the assessed course are regularly offered in both the Fall and Spring semesters, with a cap of twenty-two students in each session. Over years, we have formed a team of instructors, consisting of both full-time and adjunct professors, who become experienced in performing the assessment tasks. After some pilot practices, assessment of both the Program Criteria Indicators and the SO (2) becomes a routine since Fall 2019. Incorporating assessment as regular course components helps to maintain the standard.

Assessment data, results, and students' reports are collected and analyzed after each semester and then maintained by the course coordinator and Chair of the department's assessment committee [21]. The instructor of each session usually performs evaluations. The course coordinator will confirm the assessment results of each session, ensuring that a similar standard is used across all sessions. Analyses of the assessment results are disseminated and discussed with the department faculty before the beginning of each following semester (in the departmental meeting or the curriculum meeting). If one or more PIs are not achieved, the department will discuss necessary remedial actions at the course level and/or at the program level [16]. These suggested corrective measures are implemented in the upcoming semester immediately. In this way, an effective and sustainable assessment routine is formed. We believe this is the key to continuous improvement and maintaining accreditation [10].

Table 2 shows the timeline for assessing both the Program Criteria Indicators and the SO (2). After some pilot practices, assessments using the three assignments were performed regularly across all sessions every time the course was offered, starting Fall 2019. Specifically,

- Assessment using the I-Beam lab for the Program Criteria Indicators started in Spring 2018 across all sessions. It has continued every semester across all sessions since Spring 2018.

- Assessment using the Hub lab for the Program Criteria Indicators started in Spring 2019 with one participating session. It has continued every semester across all sessions since Fall 2019.

- Assessment using the final design project for the SO (2) started in Spring 2019 with two participating sessions. It has continued every semester across all sessions since Fall 2019.

Table 2: Assessment timeline

In-Person	Spring 2018			Spring 2019			Fall 2019			
Session #	1	2	3	1	2	3	1	2	3	4
I-Beam Lab	x	x	x	x	x	x	x	x	x	x
HUB Lab				x			x	x	x	x
Final Project				x		x	x	x	x	x

Online	Spring 2020				Fall 2020				Spring 2021			
Session #	1	2	3	4	1	2	3	4	1	2	3	4
I-Beam Lab	x	x	x	x	x	x	x	x	x	x	x	x
HUB Lab	x	x	x	x	x	x	x	x	x	x	x	x
Final Project	x	x	x	x	x	x	x	x	x	x	x	x

The global pandemic COVID-19, which occurred in the middle of Spring 2020, transitioned this course from in-person teaching to online. A preliminary study of the effect of this change was performed and presented in Sec. 5.

4 Assessment Results and Analyses

Assessment results of the three Program Criteria Indicators are shown in Figs. 4 and 5, using the I-Beam and the Hub lab, respectively. The SO (2) assessment results using the final design project are given in Fig. 6. A target of at least 70% of students meeting/exceeding the competence criteria is used in all assessments presented in this paper.

4.1 Assessment Results of Program Criterion Indicators

Assessment of the Program Criteria Indicators using the I-Beam lab is shown in Fig. 4. Data collected are across all sessions over six semesters, including three in-person offerings (Spring 2018, Spring 2019, Fall 2019) and three semesters under e-learning (Spring 2020, Fall 2020, Spring 2021). Note that the Spring 2018 semester is the pilot practice, where two areas are below the target of 70%, on calculations/analyses of stress, strain, and FOS and engineering professionalism of selecting the appropriate materials. These two areas were communicated with all course instructors at the beginning of the following semester. Students' performance in these two areas was improved starting Spring 2019. Students' performance for all three indicators met the target steadily in the recent three offerings, after some initial oscillations.

A pilot trial was run in Spring 2019 to assess the Program Criteria Indicators using the Hub lab with one participating session (out of three sessions). The performance of the participating students met the target (of 70%) for all

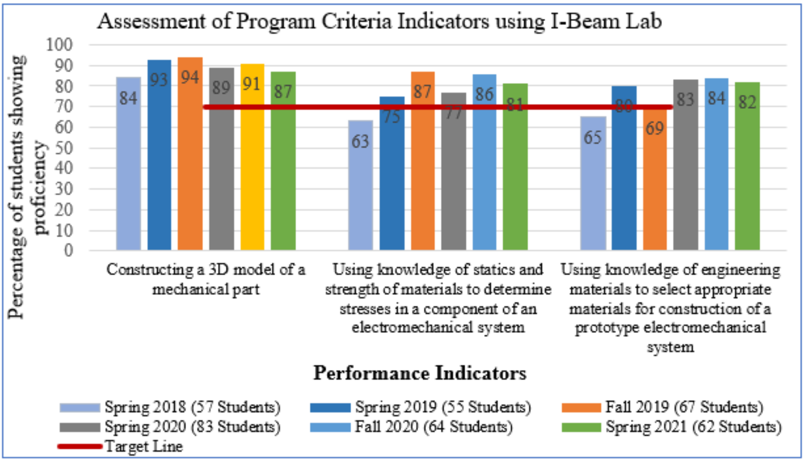


Figure 4: Assessment of Program Criteria Indicators using the I-Beam lab.

three indicators. The assessment was extended to all sessions in Fall 2019, where the percentage of students meeting/exceeding the expectation dropped below the target in two areas. These two areas are noted to be the same as previously identified by the I-Beam lab. The assessment results together confirm that these two areas still need improvement. By reviewing the requirements regarding analysis of stress, strain, and FOS and emphasizing the importance of choosing different engineering materials for performance comparison, students' performance in these two areas was improved to meet the target in the next three offerings. Despite the increased complexity in the Hub lab, the results in Figs. 4 and 5 show, via enhanced evidence, that satisfactory performance was achieved across all sessions starting Spring 2020.

During Spring 2019 and Spring 2021, both the I-Beam lab and the Hub lab were used to assess the three Program Criteria indicators. At that time, we were not sure which assignment would be more appropriate. So, both were used for double verification. It can be seen from Figs. 4 and 5 that consistent assessment results were obtained in the last three semesters (i.e., Spring 2020, Fall 2020, Spring 2021) when using both lab assignments. This indicates that using only one lab, i.e., the Hub lab that is relatively more challenging, will be sufficient for the assessment purpose. Later on, the I-Beam lab will be used as a preparation lab.

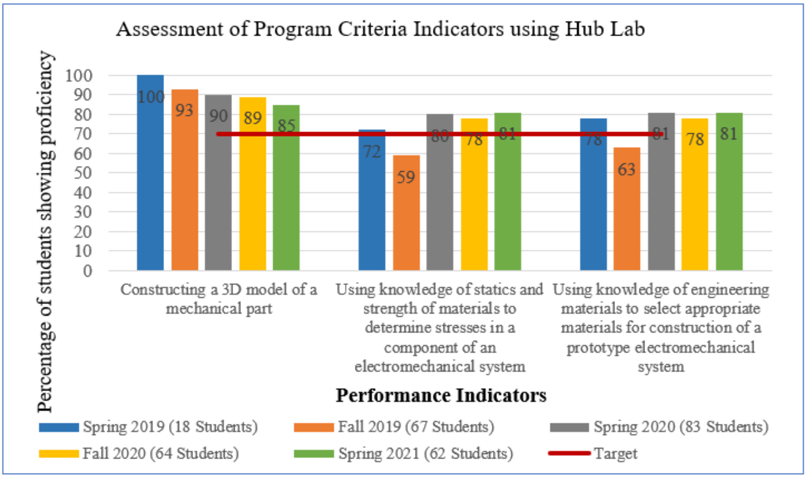


Figure 5: Assessment of Program Criteria Indicators using the Hub lab.

4.2 Assessment Results of Student Outcome (2)

Assessment of the ETAC-ABET SO (2) on design capabilities using the final design project is shown in Fig. 6. Starting with a pilot run in Spring 2019 with two out of three participating sessions, assessment results show that all areas were satisfactory except for the project management. In the next assessment cycle (Fall 2019), all sessions were assessed and satisfactory performance was achieved for all four indicators.

Assessment results in Spring 2020 show noticeable degradation in two areas. One is regarding project management skills and the other is on alternative designs. We believe this is largely due to the COVID-19 pandemic, which occurred in the middle of Spring 2020. Teaching and learning were first interrupted and then transitioned from in-person teaching to online. Before lessons resume remotely, many arrangements need to be made to ensure all students have the proper electronic devices (i.e., computers, laptops), Internet access, and software licence. Students' performance caught up in Fall 2020 in three areas, except for the project management skills. Finally, in Spring 2021, student's performance in all four indicators met the 70% target.

The assessment described in this section has been incorporated as regular course activities starting Fall 2019, across all sessions each time the course is offered. Establishing this routine helps to achieve the ultimate objective of education, i.e., continuous and constant improvement of students' competencies and learning experiences.

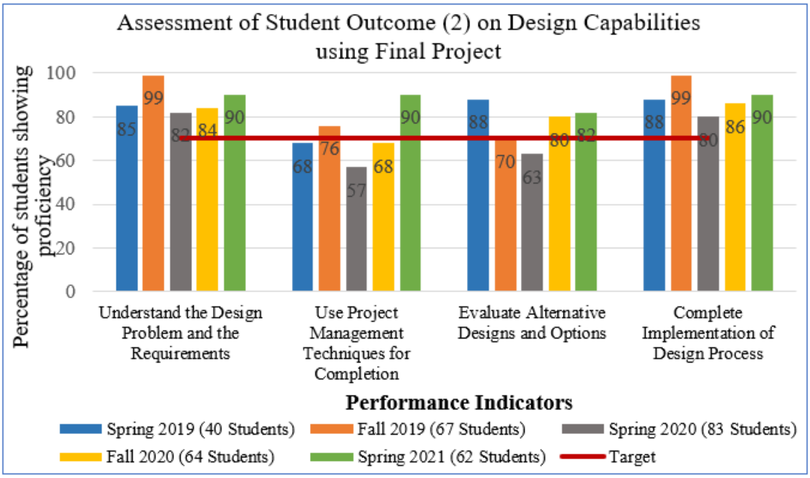


Figure 6: Assessment of SO (2) using the final design project.

5 Online Teaching

In Spring 2020, the global COVID-19 pandemic caused our institution to shift its courses from in-person teaching to online. To teach this course online, the CET department purchased a software license for SOLIDWORKS, allowing all instructors and students to install the software on their computers. Students got more access to the software this way since otherwise the software is only installed in the lab and computer center. Under e-learning, lessons were delivered via synchronous virtual meetings to closely mimic the face-to-face format, and to maintain interactions between the instructor and the students [24]. Some instructors post prerecorded lessons, allowing students to review the lab procedures outside of the class. Many instructors graded lab reports more frequently, providing timely feedback [18]. Due to these arrangements, degradation in students' performance due to the pandemic, should there be any, was not obvious in Fall 2020 and Spring 2021, as indicated in Fig. 6.

In a preliminary study of the effect of teaching modes (in-person vs. online) on students' learning performance, students' assessment scores (on a scale of 1 to 4) of the four indicators used to assess the SO (2) were analyzed using the One-Way analysis of variance (ANOVA) crossing four semesters, with two in-person (Spring 2019, Fall 2019) and two online (Fall 2020, Spring 2021). Data was processed via MATLAB's Statistics and Machine Learning Toolbox, particularly, the anova1() function. The One-Way ANOVA provides results to determine any statistically significant differences between the means of two or more independent samples [13, 2]. In this study, the independent variable is the teaching mode having two levels: in-person and online. The response variable is students' assessment scores of the final design project, as a reflection of students' learning achievement.

Table 3 shows the ANOVA analyses results by comparing the four performance indicators of the group of in-person students with those of the online group. Since all p-values are greater than 0.05, the results of the ANOVA tests indicate there is not sufficient evidence that these two groups of students under comparison perform differently. This preliminary study provides some guidance in determining the teaching format for future semesters. Offering some session(s) online seems to be a reasonable option.

Table 3: ANOVA analyses of students' assessment scores for final design project

#	Performance Indicator	p-value
1	Understand Design Specifications	0.0982
2	Time Management Skills	0.8101
3	Alternative Design	0.3652
4	Design Completion	0.2654

6 Conclusions and Future Work

This paper describes our methods of assessing both the Program Criteria Indicators and the newly-added ETAC-ABET SO (2) in a Design Laboratory, using two lab assignments and one final project, respectively. These assessment models, instruments, projects, and rubrics have been adopted as standard course components since Fall 2019 across all sessions each semester. Results presented in this paper contributed to the successful ABET visit in February 2021 and the recent re-accreditation of our programs.

In terms of Program Criteria Indicators, the target of having 70% or more of students demonstrating proficiency in all three assessed areas was successfully obtained in three consecutive semesters since Spring 2020 (Spring 2020, Fall 2020, Spring 2021). For the ETAC-ABET SO (2), all criteria were successfully met in Fall 2019, exhibited some degradation in Spring and Fall semesters of 2020 (likely due to the global COVID-19 pandemic), and exceeded the target in Spring 2021. These results demonstrate the effectiveness of our actions to improve areas of concern. Overall speaking, progress and improvement were made both at the course level, by adjusting and adding new and better-designed lab/project exercises, and at the program level, by aligning the sequence of courses related to Program Criteria and SO (2) to better prepare the students.

The assessment work helps to identify areas that deserve attention. Among the three Program Criteria Indicators, "using knowledge of statics...", which is assessed by students' abilities to perform hand computations, is found to be weaker than the other two. For future improvement, we will reinforce students' analysis and computation skills. Regarding students' design capabilities, the indicator of "using project management techniques..." is weaker than the rest. For future improvement, we will emphasize more on planning, communication methods, and project control [22]. Students are also recommended to use free software, such as Project Schedule for Android, to document their progress.

This paper also exemplifies an assessment model that includes a sequence of course activities of increasing complexity and/or new components for integrative educational and assessment purposes, covering both engineering analyses and designs. This philosophy can be readily applied to other courses in the STEM filed. For instance, a robotic course can have one project on mobile robot, one on robotic manipulator, and the third one on advanced robotic control. A control course can have one lab on analyses, one on controller design, and the third on hardware-in-the-loop design.

Results presented in this paper include both in-person (pre-pandemic) and online offerings (during/post-pandemic). A preliminary study via the ANOVA analysis shows that there is no significant difference between the performance of these two groups of participating students. In the future, we will keep collecting data and investigate suitable combinations of in-person, online, and hybrid sessions to teach the course effectively while meeting students' needs. We will also collect students' opinions of these course activities for their effectiveness in improving students' learning experiences.

References

[1] M. Abdallah, D. Jones, and I. Gherasoiu. Student outcomes assessment and evaluation for ETAC/ABET. In *ASEE Annual Conference and Exposition*, 2015.

[2] G. A. Boysen. Uses and misuses of student evaluations of teaching: The interpretation of differences in teaching evaluation means irrespective of statistical information. *Teaching of Psychology*, 42(2):109–118, 2015.

[3] J. Bullingham. Computer-aided design (CAD) in an electrical and electronic engineering degree course. In *CADCAM: Training and Education through the '80s*, pages 69–75, Dordrecht, 1985. Springer Netherlands.

[4] D. Che and S. Feazel. Engagement in practice: CAD education via service learning. In *ASEE Annual Conference and Exposition*, 2019.

[5] M. David Miller Tammie Cumming. Academic assessment: Best practices for successful outcomes with accreditation evaluation teams. *New Directions for Community Colleges*, 2019(186):81–93, 2019.

[6] I. Damaj, A. Zaher, and J. Yousafzai. Assessment and evaluation framework with successful application in ABET accreditation. *International Journal of Engineering Pedagogy*, 7(3):73–91, 2017.

[7] T. DeNucci and E. Garcia. Assessing ABET student outcomes with enhanced evidence during accreditation changes. *Journal of Higher Education Theory and Practice*, 10(11):102–111, 2020.

[8] M. A. Dettman. Developing a sustainable assessment plan for EAC-ABET accreditation. In *Latin American and Caribbean Conference for Engineering and Technology*, 2013.

[9] J. Hamblen, H. Owen, S. Yalamanchili, and B. Dao. An undergraduate computer engineering rapid systems prototyping design laboratory. *IEEE Transactions on Education*, 42(1):8–14, 1999.

[10] M. Hossieny and M. Torres. Modifying CAD courses to improve proficiency in interpretation of engineering drawings using modified constructivism approach. In *ASEE Annual Conference and Exposition*, 2017.

[11] M. Jimenez and N. G. Santiago. The supporting role of CAD/CAM tools in undergraduate research education in electrical and computer engineering. In *International Conference on Information Technology Based Higher Education and Training*, 2005.

[12] U. Kahangamage and R. Leung. Remodelling an engineering design subject to enhance students' learning outcomes. *International Journal of Technology and Design Education*, 30:799–814, 2020.

[13] K. Kumpass-Lenk, E. Eisenschmidt, and A. Veispak. Does the design of learning outcomes matter from students' perspective? *Studies in Educational Evaluation*, 59:179–186, 2018.

[14] K. Leahy. Promoting best practice design intent in 3D CAD modelling: Feedback as a constructivist paradigm of teaching and learning. *American Journal of Engineering Research*, 2(5):71–77, 2013.

[15] L. Ma, B. Mendoza, L. Hamilton, L. Aponte, S. Drabkin, and M. Cardenas. Development and implementation of an assessment model in a sophomore electromechanical systems design laboratory for the ETAC-ABET accreditation. In *ASEE Annual Conference and Exposition*, 2021.

[16] S. Mazzetto. Multidisciplinary collaboration: an integrated & practical approach to the teaching of project management. *International Journal of Continuing Engineering Education and Life-long Learning*, 30(1):52–67, 2020.

[17] Kala Meah, Donald Hake, and S. Wilkerson. A multidisciplinary capstone design project to satisfy ABET student outcomes. *Education Research International*, 5:1–17, 2020.

[18] E. Osakue. Teaching solid modeling with AutoCAD. In *ASEE Annual Conference and Exposition*, 2015.

[19] H. Ozturk and D. Raubenheimer. A program assessment tool for engineering programs. In *ASEE Annual Conference and Exposition*, 2011.

[20] S. Pidugu, M. Bakr, S. Midturi, and G. Tebbetts. Assessment process using the first year computer graphics course at university of arkansas at Little Rock. In *ASEE Annual Conference and Exposition*, 2013.

[21] M. Prince, R. Felder, and R. Prent. Active student engagement in online stem classes: Approaches and recommendations. *Advances in Engineering Education*, 8(4), 2020.

[22] R. Rabb, P. Bass, M. Bubacz, and K. Skenes. A CAD course revision: Active learning in and out of the classroom. In *ASEE Annual Conference and Exposition*, 2017.

[23] J. Shaeiwitz and D. Briedis. Direct assessment measures. In *ASEE Annual Conference and Exposition*, 2007.

[24] R. Toto, D. Frederick, and W. Pung. Instructional strategies to promote student strategic thinking when using Solidworks. *Advances in Engineering Education*, 4(1), 2014.

[25] K. Uchiyama and J. Radin. Curriculum mapping in higher education: A vehicle for collaboration. *Innovative Higher Education*, 33:271–280, 2008.

[26] X. Wang and Z. Bi. New CAD/CAM course framework in digital manufacturing. *Computer Applications in Engineering Education*, 27:128–144, 2019.

[27] R. Whalen, C. Patel, D. Tortoriello, and J. Baldacci. Augmenting a first-year design course with an undergraduate student administered Solidworks module. In *ASEE Annual Conference and Exposition*, 2013.

Sketches of Some CS1 Programming Assignments*

Paul Jackowitz[1] and Robert McCloskey[2]
Computing Sciences Department
University of Scranton
Scranton, PA 18510
[1]paul.jackowitz@scranton.edu
[2]robert.mccloskey@scranton.edu

Abstract

Among the most important aims of CS 1 is to provide students with a set of programming assignments that allows them to develop fundamental software development skills. Here we give sketches of CS 1 assignments involving the drawing of "ASCII figures" and the printing of cumulative song lyrics, inspired by our use of the textbook by Reges & Stepp (*Building Java Programs — A Back to Basics Approach* [1]). These assignments could be adapted to courses that utilize other textbooks or employ languages other than Java.

1 Introduction

The central goal of CS 1, at least for students majoring in computing, is to begin to develop fundamental software development skills. As the old adage goes, you "learn by doing". Hence, having a good set of programming problems to work on is key in attaining that goal. Here we provide sketches of several programming assignments that we have employed in teaching CS 1 for the past decade, in the hope that others who teach the course will find them, or adaptations thereof, to be useful. (Complete descriptions are available from the authors.) The assignments reflect our use of *Building Java Programs — A Back to Basics Approach*, by Reges & Stepp [1], which takes the view that objects can wait until after a novice programmer has gained some skill in using the basic procedural programming constructs, including variable declaration and assignment, arithmetic expressions, conditional (i.e., if-else) statements, loops, and method calls/declarations. However, they could be adapted to courses that use other textbooks or other popular languages, including Python, C++, or C#.

2 Procedural Decomposition

Among the difficulties in teaching CS 1 is devising meaningful programming exercises early in the semester, when students have been exposed to only a small subset of a programming language's constructs. Part of the solution lies in making a good choice regarding the order in which those constructs are introduced. In this respect, [1] is rather unconventional.

The first programming principle presented in [1] is procedural decomposition, which suggests that a complex task should be divided into a set of simpler subtasks. In terms of programming in Java, this translates into an appropriate use of methods. In particular, if a subtask is performed multiple times, and in different contexts, that would be a strong indicator that it "deserves" to be implemented in its own method.

```
   /\         \    /
  /  \         \  /
 /    \         \/
 \    /         /\
  \  /         /  \
   \/         /    \
```

Figure 1

To illustrate this principle, all that is needed are "print" statements and (parameterless) methods, which together can be employed to produce primitive "drawings", such as those in Figure 1 (which are taken from [1]). The idea is that a well-modularized program that draws both figures would reflect the fact that the top half of the diamond shape (a cone) is produced in the same way as the bottom half of the 'X', and vice versa (a V-shape), so that the methods that draw a diamond and an 'X', respectively, would rely upon lower-level methods (containing only statements that print string literals) that draw a cone and a 'V':

```
public static void drawDiamond() { drawCone(); drawV(); }
public static void drawX()       { drawV(); drawCone(); }
```

After presenting this example —and perhaps a few others with slightly more complexity— we typically ask students to make extensive use of procedural decomposition in developing a program that "draws" one or more figures having repeated parts.

```
           ###
+-----------------------+
|     |  |     |  |     |
|     |  +-----+  |     |
|     |           |     |
|     |           |     |
|     +-----------+     |
|                       |
|                       |
|                       |
|                       |
+-----------------------+
|                       |
|                       |
|                       |
|                       |
|     +-----------+     |
|     |           |     |
|     |           |     |
|     |  +-----+  |     |
|     |  |     |  |     |
+-----------------------+
           ###
```

Figure 2

One example is the rendering of the soccer field shown in Figure 2. Notice that there are seven distinct lines of text, each of which appears two or more times in the drawing. Figure 3 shows excerpts from a solution. (There and elsewhere in figures that present source code, we use ellipses to indicate missing parts, omit the prefix "`System.out`" from calls to methods that print strings, and compromise formatting convention by placing multiple statements per line along with other accommodations, so as to save space and provide emphasis.)

Among the figures that we have used as the basis of assignments are a football field, "boxes" of various shapes stacked upon one another, and others that are rather contrived. Students are expected to recognize the portions of the figures that appear multiple times and, for each one, to formulate and make use of a method that draws it. To be successful, a student must have a clear understanding of how method calls affect a program's flow of control.

The more essential point has to do with motivating students to recognize the potential merit of refactoring a highly redundant sequence of print statements into an equivalent modularized program. We emphasize the potential problems redundant statements may present in software. We ask, what if a statement, used in multiple places, is subsequently determined to be incorrect, or what if such a statement is desired to be modified (for example, to change the width of the soccer field)? How certain might one be that they have found and modified all of the appropriate statements in need of change? Developing an appreciation of such issues, when they can readily see the results in the output, helps to prepare students to apply similar thinking later on when working with much more involved tasks where abstraction will be vitally important. Even recognizing that there may be multiple worthwhile ways to decompose a task has merit, as it may, and should, lead to discussions regarding the relative merits of different approaches.

A quite different kind of programming exercise by which to teach procedural decomposition — again needing only print statements and parameter-less

```
public class SoccerField {

  static void main(String[] args)
  {
    goal(); widthLine();
    northPenaltyBox();
    midField();
    midfieldLine();
    midField();
    southPenaltyBox();
    widthLine(); goal();
  }

  static void northPenaltyBox()
  {
    goalBoxSide();
    goalBoxTop();
    penaltyBoxSide();
    penaltyBoxSide();
    penaltyBoxTop();
  }

  static void midField() {
    midFieldSide();
    midFieldSide();
    midFieldSide();
    midFieldSide();
  }

  static void midfieldLine()
    { widthLine(); }

  static void goal()
    { println(" ### "); }

  static void widthLine() {
    println("+-----------------------+");
  }

  static void goalBoxSide() {
    println("|     |  |     |  |     |");
  }

  static void goalBoxTop() {
    println("|     |  +-----+  |     |");
  }

  static void penaltyBoxSide() {
    println("|     |           |     |");
  }

  ...

}
```

Figure 3

methods— is to have students develop a program that prints the lyrics of a cumulative song (i.e., one in which each verse extends the previous one) [2].

The classic example is *The Twelve Days of Christmas*. We have used *Old MacDonald Had a Farm*, *There Was an Old Lady Who Swallowed a Fly*, and the Irish folksong *Rattlin' Bog*. To illustrate, Figure 4 shows excerpts of a Java program that prints the lyrics of one version of *Old MacDonald* and that demonstrates a significant use of procedural decomposition, along with (admittedly) method chaining.

3 Loops

Once students have been introduced to variables, assignment, and arithmetic expressions (which [1] does in Chapter 2), more figure-drawing exercises can be used as a vehicle for exploring iteration and generalization using the for-

```
public class OldMacDonald {

  public static void main(String[] args) {
    preamble(); print("a cow, "); eieio(); verseMoo();
    preamble(); print("a duck, "); eieio(); verseQuack();
    preamble(); print("a pig, "); eieio(); verseOink();
    preamble(); print("a dog, "); eieio(); verseWoof();
  }

  ...

  private static void verseQuack() {
    println("With a quack-quack here and a quack-quack there");
    println("Here a quack, there a quack, everywhere a quack-quack");
    verseMoo();
  }

  private static void verseOink() {
    println("With an oink-oink here and an oink-oink there");
    println("Here an oink, there an oink, everywhere an oink-oink");
    verseQuack();
  }

  ...

  private static void preamble()
    { oldMacHadAFarm(); andOnThisFarm(); }

  private static void oldMacHadAFarm()
    { print("Old MacDonald had a farm, "); eieio(); }

  private static void andOnThisFarm()
    { print("And on this farm he had "); }

  private static void eieio() { println "E-I-E-I-O"; }

}
```

Figure 4

loop construct. As an example, consider once again the diamond and X-shapes discussed earlier. Figure 5 shows most of a program that prints the two shapes. An easy-to-change global constant, SIZE, determines the size of each one. Other figure-drawing programs (e.g., for the soccer field) can be refactored in a similar way so that the sizes of the produced figures are determined by global constants.

The point here is for students to learn how to generalize their programs so that straightforward differences in output can be effected without having to make wholesale modifications to the source code. Of course, once students learn how to assign values to variables through input at runtime, those inputs can be used in place of constants, thereby yielding programs that are sensitive to user (or file) input.

```
public class drawDiamondAndX {
  private static final int SIZE = 6;

  public static void main(String[] args)
    { drawDiamond(); drawX(); }

  public static void drawDiamond()
    { drawCone(); drawV(); }

  public static void drawX()
    { drawV(); drawCone(); }

  public static void drawCone() {
    for (int i=1; i <= SIZE; i++) {
      for (int j=0; j != SIZE-i; j++) {
        { print(" "); }
      print("/");
      for (int j=0; j != 2*(i-1); j++)
        { print(" "); }
      println("\\");
    }
  }
  ...
}
```

Figure 5

4 Parameterization

Parameter passing is introduced in Chapter 3 of [1], providing the opportunity to further improve the figure-drawing (and lyric-producing) programs by making use of parameters —in place of global constants or variables— to dictate the behavior of methods.

This is our next step in teaching students when, how, and why to use methods. By utilizing parameters, methods become more self-contained, insulating them from being coupled with external data and thereby increasing their potential for reuse.

The task of having a program draw a checkerboard-like pattern, as in Figure 6, is one that we have employed at this point in the course. This pattern can be described by multiple parameters, including ones indicating the height and width of each cell and the number of rows and columns on the board. Additionally, the characters employed in printing the alternating “light” and “dark” cells can be parameters.

```
public class CheckerBoard {
  static final char DARK_CHAR = '#';
  static final char LIGHT_CHAR = '-';

  static void main(String[] args) {
    int boardSize = 3;
    int cellSize = 4;
    drawBoard(boardSize,cellSize);
  }

  static char otherOf(char c) {
    if(c == DARK_CHAR)
      { return LIGHT_CHAR; }
    else
      { return DARK_CHAR; }
  }

  static void drawBoard(int boardN,
                        int cellN) {
    char start = DARK_CHAR;
    for(int i=0; i != boardN; i++) {
      drawRow(start,boardN,cellN);
      start = otherOf(start);
    }
  }

  static void drawRow(char start,
                      int columns,
                      int cellSize) {
    for(int i=0; i != cellSize; i++) {
      drawLine(start,columns,cellSize);
      println();
    }
  }

  static void drawLine(char start,
                       int columns,
                       int cellWidth) {
    for(int i=0; i != columns; i++) {
      print(start,cellWidth);
      start = otherOf(start);
    }
  }

  static void print(char c, int n) {
    for(int i=0; i != n; i++)
      { print(c); }
  }
}
```

Figure 7

Figure 7 presents the source code for a program that prints such a checkerboard-like pattern. The character variables are global and the board and cells are square, so as to limit the number of parameters to two, but the essential idea is illustrated. Indeed, one can provide this program to students and ask them to generalize it so that the `drawBoard()` method receives all six potential parameters rather than only two.

```
####----####
####----####
####----####
####----####
----####----
----####----
----####----
----####----
####----####
####----####
####----####
####----####
```

Figure 6

Figure 8 presents the source code for a program that is a refactoring of the Old MacDonald program to make use of parameters, in this case strings that represent the animal names and utterances. This program also presents an opportunity for if-else statements to be used for the purpose of choosing the correct article (i.e., "a" versus "an") to precede those strings.

```
public class OldMacDonald {

  public static void main(String[] args) {
    preamble("cow"); verseMoo();
    preamble("duck"); verseQuack();
    preamble("pig"); verseOink();
    preamble("dog"); verseWoof();
  }

  static void preamble(String animal)
    { oldMacHadAFarm(); andOnThisFarm(animal); eieio(); }

  static void verseQuack()
    { withA("quack"); verseMoo(); }

  static void verseOink()
    { withA("oink"); verseQuack(); }

  static void oldMacHadAFarm()
    { print("Old MacDonald had a farm, "); eieio(); }

  static void withA(String noise) {
    println("With a " + noise + "-" + noise + " here and a " +
            noise + "-" + noise + " there");
    println("Here a " + noise + ", there a " + noise +
            ", everywhere a " + noise + "-" + noise);
  }

  static void andOnThisFarm(String animal) {
    print("And on this farm he had a " + animal + ", ");
  }
}
```

Figure 8

5 Arrays

Later in the course, when arrays have been introduced, it is possible to employ them in programs that produce cumulative songs, resulting in code that is very easy to modify so as to produce alternative lyrics. This continues the process of teaching students to strive for generality and flexibility in the programs they develop. Figure 9 illustrates this with excerpts from yet another version of the Old MacDonald program. Here, a pair of parallel arrays is used to specify the various animal species' names, and their corresponding utterances, making it easy to add or modify verses.

```
public class OldMacDonald {

  static String[] animals = { "cow", "duck", "pig", "dog" };
  static String[] utterances = { "moo", "quack", "oink", "woof" };

  public static void main(String[] args) {
    for (int i=0; i != animals.length; i++) { verse(i); }
  }

  static void verse(int k) {
    oldMacHadAFarm();
    andOnThisFarm(animals[k]);
    restOfVerse(k);
  }

  static void restOfVerse(int k) {
    for (int j = k; j != -1; j--) {
      withA(utterances[j]);
    }
    oldMacHadAFarm(); println();
  }

  static void andOnThisFarm(String animal) {
    print("And on this farm he had a " + animal + ", "); eieio();
  }
   ...
}
```

Figure 9

6 Objects

Figure 10 presents a further refactoring of the Old MacDonald program that uses a single array of Animal objects, rather than a pair of parallel arrays of strings. The Animal class is straightforward and is thus omitted here. This mature version makes even better use of abstraction and language features. Note the ease and reliability with which verses may be added and modified. It also decouples the `verse()` and `restOfVerse()` methods from any global variables.

7 Conclusion

When teaching CS 1, one must decide in what order to introduce various programming language constructs, and once that order is decided upon a corresponding set of constraints is imposed. Reges & Stepp [1], like every textbook,

```
public class OldMacDonald {

  public static void main(String[] args) {
    Animal[] animals = { new Animal("cow", "moo"),
                         new Animal("duck", "quack"),
                         new Animal("pig", "oink" ),
                         new Animal("dog", "woof" )
                       };
    for (int i=0; i != animals.length; i++)
      { verse(animals,i); }
  }

  static void verse(Animal[] animals, int k) {
    oldMacHadAFarm(); andOnThisFarm(animals[k].getName());
    restOfVerse(animals,k);
  }

  static void restOfVerse(Animal[] animals, int k) {
    for (int j = k; j != -1; j = j-1) {
      withA(animals[j].getUtterance());
    }
    oldMacHadAFarm(); println();
  }
  ...
}
```

Figure 10

follows one such ordering. Debates about the relative merits of one versus another (e.g., objects-early vs. objects-late) have gone on for decades and will undoubtedly continue. The purpose of this brief paper has not been to necessarily advocate for one particular ordering, but rather to share sketches of assignment ideas we have employed that may be used under some of these constraints. If they inspire related ideas and sketches usable under other sets of constraints, then all the better.

References

[1] Stuart Reges and Marty Stepp. *Building Java Programs: A Back To Basics Approach.* Pearson, Hoboken, NJ, 2019.

[2] Wikipedia. Cumulative_song. https://en.wikipedia.org/wiki/Cumulative_song.

A Lecture That Guides Students to Build A Simple Blockchain*

Tutorial

Penn Wu
DeVry University
Chino Hills, CA 91709
pwu@devry.edu

Blockchain is a commonly defined as a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a non-centralized business network. It is the technology that stores data in the form of a well-defined block and is now commonly used to store data for cryptography and cryptocurrencies such as Bitcoin.

In today's fast-changing market, Blockchain has be recognized as a practical application for Cryptocurrency, Supply Chain Management, Digital Identity, Asset Tokenization, Voting, Notary, college transcript, and so on. It is also a practical topic for college instructors to engage students of computer science with the development of career-related skills.

This workshop will share with audiences how to prepare instructional materials to guide students through a set of hands-on coding activities to learn the core technologies and key components of Blockchain. It will provide the audiences an opportunity to build a simple blockchain with limited coding efforts. The objective is to help the audiences to: (1) understand what it takes to build a blockchain, (2) guide students to write codes to build simple blockchains, (3) brainstorm pedagogies to teach blockchain technologies, and (4) develop instructional materials to encourage students to take actions to explore, discover, understand, and develop their blockchains.

The Blockchain Art Simulation (BARTS) and Experiential Exercises*

Tutorial

George Sanders and Sean Sanders
University at Buffalo
The State University of New York
Buffalo, NY 14260

This workshop introduces the online Blockchain ART Simulation (BARTS). The simulation is used to illustrate the interrelationship of blockchain mining, cryptocurrency, non-fungible tokens (NFT) and hashing concepts. We developed the simulation as a way to gamify the learning process where students participate in a simulation for buying and selling NFT drawings. A set of and a set of experiential online exercises are also presented illustrating hashing concepts for validating blockchain transactions. The workshop participants engage in the mining process using their phones. We have used this material successfully with over 400 students ranging from high school students to graduate students.

Git-Keeper: Streamlined Software for Automated Assessment Workflows*

Tutorial

Ben Coleman
Moravian University
Bethlehem, PA 18018
colemanb@moravian.edu

Nathan Sommer
Xavier University
Cincinnati, OH 45207
sommern1@xavier.edu

Git-keeper is a tool that allows students to obtain and submit assignments using Git. Students receive immediate feedback on their submission based on faculty-created assessment, which allows them to make corrections and resubmit. Faculty can utilize any language or tool that executes via the command line on a Linux system. During the workshop, participants will first act as students to complete a sample assignment. Once they understand the simplicity of git-keeper from the student perspective, the bulk of the workshop will focus on how to create an assignment, publish it to a class, and fetch student submissions.

Reflective Curriculum Review for Liberal Arts Computing Programs*

Tutorial

Jakob Barnard[1], Grant Braught[2], Janet Davis[3],
Amanda Holland-Minkley[4], David Reed[5], Karl Schmitt[6],
Andrea Tartaro[7], James Teresco[8]
[1]University of Jamestown, Jamestown, ND 58405
Jakob.Barnard@uj.edu
[2]Dickinson College, Carlisle, PA 17013
braught@dickinson.edu
[3]Whitman College, Walla Walla, WA 99362
davisj@whitman.edu
[4]Washington & Jefferson College, Washington, PA 15317
ahollandminkley@washjeff.edu
[5]Creighton University, Omaha, NE 68178
DaveReed@creighton.edu
[6]Trinity Christian College, Palos Heights, IL 60463
Karl.Schmitt@trnty.edu
[7]Furman University, Greenville, SC 29690
andrea.tartaro@furman.edu
[8]Siena College, Loudonville, NY 12211
jteresco@siena.edu

The ACM/IEEE-CS/AAAI curricula task force is currently developing an updated set of Computer Science Curricula guidelines, referred to as CS202X (since the release date is not yet determined). Information about the task force and preliminary drafts of the Knowledge Areas that will be included in the guidelines can be found online at http://csed.acm.org. To assist institutions in applying the new guidelines, CS202X will also publish a Curricular Practices

Volume. This volume will include an article by the SIGCSE Committee on Computing Education in Liberal Arts Colleges that will focus on designing or revising CS curricula in liberal arts contexts. Liberal arts colleges, and smaller colleges in general, face unique challenges when designing curricula. Small faculty sizes, limits on the number of courses that can be required for a major and the need for flexibility in student programs of study constrain designs. However, these environments also provide the opportunity to craft distinctive curricula fitted to institutional mission, departmental strengths, locale, student populations and unique academic experiences. These challenges and opportunities, combined with the size of prior curricular recommendations, have often forced smaller programs to assess trade-offs between achieving full coverage of curricular recommendations and their other priorities.

The SIGCSE Committee on Computing Education in Liberal Arts Colleges has heard from many faculty that their institutional and departmental contexts have indeed complicated the adoption of prior curricular guidelines. While the CS2013 and upcoming CS202X recommendations provide some flexibility for curriculum designers by dividing content into core and supplemental categories, smaller colleges still face challenges selecting content and packaging it into coherent curricula. To assist in this process, the committee is developing guidance for effectively integrating CS202X as a part of the design, evaluation and revision of computer science and related programs in the liberal arts. This guidance will encourage faculty to reflect on their programs and the role of CS202X, beginning with their institutional and departmental priorities, opportunities and constraints. Ultimately, this guidance will be presented in the committee's article in the CS202X Curricular Practices volume.

This session will open with an overview and brief discussion of the current CS202X draft. Participants will then begin working through a preliminary version of the committees' reflective assessment process. This process is framed by a series of scaffolding questions that begin from institutional and departmental missions, identities, contexts, priorities, initiatives, opportunities, and constraints. From there, participants will be led to identify design principles for guiding their curricular choices including the CS202X recommendations. Participants will leave the session with a better understanding of how CS202X can impact their programs and a jumpstart on the reflective assessment process. Feedback on the process and this session are welcome and will be used to refine the committee's guidance prior to its publication in the CS202X Curricular Practices volume.

Using METAL's Expanded Map-based Algorithm Visualizations in Computer Science Courses*

Tutorial

James Teresco
Siena College
Loudonville, NY 12211

The Map-Based Educational Tools for Algorithm Learning (METAL) project (https://courses.teresco.org/metal/) provides a set of graphs based on real-world highway data, and map-based algorithm visualization (AV) tools that use this data. This tutorial will introduce attendees to METAL's data, AV capabilities, and course assignments, all of which have expanded greatly in recent years.

Highway Graph Data

Data is derived from the hobbyist "Travel Mapping" project (https://travelmapping.net). Each graph vertex is a "waypoint" (e.g., major intersection), represented by a place name and a latitude-longitude pair, and each edge is a "connection", which links together two waypoints that are directly connected by a road. Granularity is fine enough for sufficiently accurate mapping, but coarse enough that meaningful graphs are not overwhelmingly large.

METAL's data has characteristics that make it especially useful academically. It has a direct, motivating real-world connection. Students can work with graphs representing familiar regions and roads. The variety of sizes and other characteristics, such as connectedness, can be useful and instructive. The smallest graphs have very few (<10) vertices and edges, useful for initial development and debugging. Many graphs contain a few hundred to a few thousand vertices and edges, useful as larger, but still traceable, examples. Large graphs, good for stress testing or empirical timing studies, range in size up to 750,000+ vertices and 900,000+ edges. Currently, only undirected graphs are supported.

Highway Data Examiner

The Highway Data Examiner (HDX) allows users to plot highway graph data, sets of points, and paths (any of which can be the inputs or outputs of programs that use METAL data) right on an in-browser map, and provides a rich set of interactive AVs. AVs display a pseudocode representation of the algorithm, and as lines are "executed", the graph data and values of important variables and data structures are shown in both tabular and map form. Colors help show the progress and results of AVs. Execution speeds can be controlled to focus on low-level details of the algorithm at slow speeds or to gain a valuable higher-level view of an algorithm's progress at fast speeds. AVs for vertex-only algorithms include extremes search, closest pairs, space-filling curve traversal orders, traveling salesman, geometric partitioners, and convex hulls. AVs that use the full graph structure include graph traversals (breadth-first, depth-first), Dijkstra's algorithm for single-source shortest paths, and minimum spanning trees with Prim's algorithm or Kruskal's algorithm.

Tutorial Overview

After a brief project introduction and overview, the tutorial will focus on how to bring METAL data and AVs into classes with a low barrier to entry for instructors. Participants will work with a selection of examples and assignments that have been used successfully to help students learn about algorithms more quickly or to gain a deeper understanding of the algorithms they are studying.

Do you know your cookies?*

Tutorial

Susan Conrad and Soliana Hailemichael
Marymount University
Arlington, VA 22207
{sconrad,sgh34601}@marymount.edu

It is most likely that each of us when surfing the web online has been informed by a website that the site collects cookies. But do most of us understand what data is being collected and what happens with that data? This workshop will introduce participants about the use of cookies in online environments. It will give provide participants the opportunity to analyze the various types of data cookies, discuss how the cookies collect and store data and show how cookies are deleted, if they can be. The workshop will also discuss privacy notifications and what information is required to be disclosed in the notification. There will be a discussion about why IT professionals need to understand cookies and privacy notification. Participants will work on a computer to find information and about notifications and cookies and share what they find with the other participants.

Using Single-Board Miniature Computers (SBCs) to Create Basic Artificial Intelligence (AI) and Machine Learning (ML) projects*

Tutorial

Natalia Bell
School of Technology and Innovation
Marymount University
Arlington, Virginia 22207
nbell@marymount.edu

The adoption of Artificial Intelligence (AI) and Machine Learning (ML) technologies is widely expanding in our society and academia is gradually considering developing more programs and courses that teach students what Artificial Intelligence (AI) is, explore use cases and applications of AI and understand AI concepts through active learning methods. In preparation for developing a new Bachelor's of Science in Artificial Intelligence Program, the author is preparing course material for an introductory-level Principles of Artificial Intelligence course. This tutorial will demonstrate how to use low-cost, credit card-sized single-board miniature computers (SBCs) like Raspberry Pi, Banana Pi, BeagleBone® AI-64, etc. to create a basic AI/ML project that can be used in an introductory course for students of all abilities. In the tutorial, the author will present various SBCs, discuss open-source materials for instructors to create such projects, and demonstrate a basic AI/ML Project. The SBCs hardware systems can also be used for advanced Machine Vision, Deep Learning, and Robotics projects.

Incremental Picross Puzzle Development*

Nifty Ideas

Carole Pe Rosiene[1] and Joel A. Rosiene[2]
[1]Department of Computing Sciences
University of Hartford, Colchester, CT 06117
rosiene@hartford.edu
[2]Department of Computer Science
Eastern Connecticut State University, Windham, CT 06226
rosienej@easternct.edu

In a second CS1 course, we introduce separate assignments and laboratories gradually; these related code developments are based on a picture logic puzzle in which cells in a grid must be selected or left blank according to clues provided at the side of the grid to reveal a hidden picture. The puzzle is called Picross or Nonogram among many other names. The collective, assigned work incrementally builds the final version of a functioning puzzle. Since this is a second course in programming, the first assessed work focuses on the manipulation of a one-dimensional array to start. In succeeding assignments and labs, the intended outcomes include the incorporation of two-dimensional arrays, polymorphism, error handling, recursion, and a graphical user interface. This course is taught in Java and JavaFX, a toolkit for developing rich client applications. The approach of incrementally introducing different programming concepts around a main theme is definitely not novel. However, the use of a commonly available puzzle (available on websites and paperback) and how programming concepts are incorporated in this particular instance make up a series of effective, practical and nifty assignments and labs. While the end product is being incrementally developed, students have an idea, from the start, what the eventual product should be. Students are also aware that in the end, they will have developed a useful product which can be played as opposed to a creation that students might have trouble relating. Preliminary survey results around the use of the same theme throughout the course have been positive, as is the use of a familiarly-known puzzle.

Interesting Exercise to Demonstrate Self-balancing Trees*

Nifty Ideas

Samah Ahmed Senbel
Computer Science
Sacred Heart University, Fairfield, CT 06825
senbels@sacredheart.edu

Data structures is a significant part of the foundation of computer science for computer programming students. It introduces students to the different data structures and their uses, properties and implementation. The course starts by covering the simpler linear data structures: arrays, linked lists, stacks, queues, and hashing, then trees of all types; binary search trees (BST), AVL trees or red-black trees, heaps, and B-trees, and finally graphs are briefly introduced. Students often have no problem understanding the concepts, but struggle with the analysis, comparison, and implementation of the different structures. Therefore, it is important to introduce as many demonstrations, examples, memorable activities and interesting assignments as possible.

Trees data structures and their performance is one of the main topics to teach in a data structures course. Appreciating the importance of tree structure and tree height in software performance is an important concept to teach. In this talk, a simple and amusing activity is presented. It demonstrates to students the importance of a well-balanced tree by comparing the height of a binary search tree to a balanced (AVL) tree build upon some personal data to find the "prettiest" tree (minimum height). The activity highlights the fact that, irrelevant of your data sequence, a balanced tree guarantees a height of O(log n) and everyone "wins" the beauty contest.

The exercise goes as follows: Each student gets to answer 16 personal questions with an integer reply, thus generating a unique data sequence for each student. Students then build a regular BST with this data and the student with the minimum height would be declared the winner of the BST beauty contest. Then, the same data is used to build an AVL balanced tree, and to everyone's delight, the whole class gets the minimum possible height, and everyone wins a small gift.

Adding Interactive Content To Dive Into Systems, a Free Online Textbook for Introducing Students to Computer Systems*

Lightning Talk

Suzanne Matthews[1] and Tia Newhall[2] and Kevin Webb[2]
[1]U.S. Military Academy
West Point, NY 10996
suzanne.j.matthews@gmail.com
[2]Swarthmore College
Swarthmore, PA 19081
{newhall,kwebb}@cs.swarthmore.edu

This lightning talk presents our experiences, motivations, and goals for further developing Dive into Systems (diveintosystems.org), a free, online textbook that introduces computer systems, computer organization, and parallel computing. Our book's topic coverage is designed to give readers a gentle and broad introduction to these important topics. It teaches the fundamentals of computer systems and architecture, introduces skills for writing efficient programs, and provides necessary background to prepare students for advanced study in computer systems topics. Our book assumes only a CS1 background of the reader and is designed to be useful to a range of courses as a primary textbook for courses that introduce computer systems topics or as an auxiliary textbook to provide systems background in other courses. Results of an evaluation from students and faculty at 18 institutions who used a beta release of our book show overwhelmingly strong support for its coverage of computer systems topics, its readability, and its availability. Chapters are reviewed and edited by external volunteers from the CS education community. Their feedback, as well as that of student and faculty users, is continuously incorporated into its online content.

Our new NSF-funded effort involves adding free interactive exercises to our on-line textbook and add instructor materials to supplement our textbook. Over the course of our 3-year grant, we will be assembling teams of exercise developers to create interactive content for specific book chapters. Each exercise developer will receive a $1,000 stipend for their work with the team and will be expected to attend our pre-symposium event at future SIGCSE conferences (additional travel funds will be provided). As part of this effort, we also seek other auxiliary content from the community to add to our instructor portal.

Version 1 release is freely available at https://diveintosystems.org.

Leveraging SOCMINT: Extrapolating Cyber Threat Intelligence From Russia-Ukraine Conflict*

Student Paper Abstract

Bipun Thapa
Marymount University, Arlington, VA 22207
bst77492@marymount.edu

The paper aims to derive Cyber Threat Intelligence (CTI) from the Russia-Ukraine conflict with the help of Social Media Intelligence (SOCMINT), a framework that emanates reasoning from voluntarily available public information—using open-source tools and APIs, datasets created is assessed through topic modeling, thematic analysis (word cloud), Logit function, and neural network classification. The topic modeling and word cloud failed to provide consequential intelligence due to weak datasets - censorship and integrity remain big concerns. Logit function supplied statistically significant features that were influential in the outcome of the tweets, and MLP, a neural network classifier, yielded 91% accuracy when identifying tweet alliances (Russia or Ukraine).

Simplified Decision Model for Free VPN Services Selection in Light of the COVID-19 Pandemic*

Student Paper Abstract

Jonathan Fitzgerald, Matthew Green,
Jack Bluebaugh, Xinliang Zheng
Department of Computer Science & Information Technologies
Frostburg State University, Frostburg, MD 21532
{jjfitzgerald01, mdgreen0, jrbluebaugh0, xzheng}@frostburg.edu

The Covid-19 Pandemic had and continues to have a profound impact on network security in a variety of facets. E-commerce and communications exploded during lock down and quarantines having a dynamic effect on large businesses and small alike. While most business models would dictate that maximizing profit is reliant on cost cutting measures, the pandemic forced business to adjust the business models away from traditional in person shopping towards a virtual meetings and sales – calling into question the existing security models. As a result, small businesses without a specialized technology personnel need to pull different IT technologies together to survive the new environment. For many small businesses free VPN is one of the IT tools to provide certain degree of cross network communication protection. Our paper works to create a simplified decision-making model that allows for decision makers to best determine their needs for their specific business and use the model to choose the right third-party free VPN program.

The Human Factor in Computer Security: Email Security*

Student Paper Abstract

Phillip Behrns
College of Business, Innovation, Leadership, and Technology
Marymount University, Arlington, VA 21532

This essay will focus on email security and as an example of the human factor in computer security. Three most common types of business compromise email BCE are phishing, spoofing and impersonation. Although simple in construction and overall strategy business compromise emails are surprisingly effective. A worrying number of data breaches are traced back to human error related to one of these three types of compromising emails. The goal is to raise awareness of how common and successful business compromise emails can be. Then provide an idea of how to safely deal with business compromise emails when you discover them in your inbox. In conclusion reducing unintentional victims of business compromise emails will reduce data breaches and potentially save a lot of money for related entities.

Password Manager in an Enterprise Environment*

Student Paper Abstract

Jennifer Ferreras-Perez
School of Technology and Innovation
Marymount University, Arlington, VA 21532

The integration of Password Managers (PMs) in enterprise environments continues to rise due to numerous security capabilities they provide. The purpose of this study is to identify the benefits brought on by the implementation of password managers in an enterprise environment. Through a systematic literature review it was revealed that the use of PMs could help improve secure password practice, reduce password-related calls to helpdesks, facilitate password management, and enhance productivity. Consequently, reducing password-related stress, and improving password hygiene in various types of organizations. Summarily, the overall conclusions of this research study emphasize that the implementation of PMs at an enterprise level could be utilized to mitigate various password-related security breaches. Henceforth, the enterprise-level PMs must be encouraged in any industry, primarily those working in IT-related jobs.

Evaluating the Quality of Art Generated by CLIP+VQGAN*

Student Paper Abstract

Aaron Marker
Computer Science Department
Randolph-Macon College, Ashland, VA 23005

Generative Adversarial Networks have improved greatly since their introduction in 2014. An area where this is especially true is with art generation. Earlier models such as ArtGAN and the Creative Adversarial Network (CAN) modified the original GAN architecture with the intention of creating the best art possible. More modern deep learning models such as StyleGAN and CLIP+VQGAN, though a little more general in their applications, can be employed to generate even more realistic art. There is a problem that arises as GANs get better at creating art. There are widely used evaluation metrics for general data accuracy such as Inception Score and Fréchet Inception Distance. These evaluation metrics cannot evaluate an idea as abstract or subjective as art. In trying to match or exceed the ability of a human to create art, it is necessary to use some metric by which different artificial intelligence architectures can be evaluated both independently, to find their weaknesses, and in comparison with other architectures, to determine the architecture best equipped to generate art. This paper explores how to define art as wholistically and concretely as possible so that the quality of art can be evaluated in GANs. In attempt to best target this definition, a survey was designed and conducted evaluating the quality of the CLIP+VQGAN architecture. This paper proposes a survey to evaluate artistic quality of the images produced by a GAN in the most objective way possible.

A Pedagogical Approach to Leveraging Gamification for Cybersecurity Awareness in Higher Education*

Student Paper Abstract

Jeremy Gilmore
Computer Science and Information Technology Department
Kutztown University of Pennsylvania, Kutztown, PA 19530
jgilm180@live.kutztown.edu

In a time when the Internet is a dominant tool used by billions of people, cybersecurity becomes an important topic that needs to be treated with the utmost importance. "Bad actors" have been known to come up with crafty and sophisticated attack vectors or approaches to catch unsuspecting victims and hijack their hardware, compromise their system, or steal personally identifiable information (PII), among other things. One such group of victims is college students. Between time spent on homework, research, extracurriculars and personal activities, students are among those who spend the largest amount of time on the Internet. Not only that, but information systems used by universities contain troves of sensitive data, such as social security numbers or bank account information. Information Security Buzz (2018) has done extensive research noting a history of massive data breaches within these systems affecting hundreds of thousands or even millions of students. As such, students must become well-versed in following good principles and avoiding bad habits, so that their online experience will be as secure as humanly possible. One unique pedagogy is to train students about online safety through the use of gamification. When game mechanics are applied within a non-game context, it offers many benefits as an educational method. This paper goes into the process of applying the four phases of the game development life cycle (GDLC) to develop a web-based trivia gamification to educate college-level students about cybersecurity concepts.

Identifying Commonalities of Cyber Adversaries Attacking the Maritime Transportation System*

Student Paper Abstract

Rebecca J. Rohan
School of Business, Innovation, Leadership, & Design
Marymount University, Arlington, VA 22207
Rebecca_Rohan@marymount.edu

Our global economy is dependent upon the maritime transportation system (MTS) to move cargo and goods around the world. This paper looks to identify commonalities in cyber adversaries attacking the maritime transportation system (MTS). An exploratory document analysis is used to identify cyberattacks impacting the MTS and the adversaries behind the cyberattacks. Aspects of the MTS more often experience cyberattacks impacting availability, involving data exfiltrations or ransomware, or state-sponsored by China or Russia.

Online Video Gaming Fostering Leadership Skills*

Student Paper Abstract

Samantha Rose King
School of Technology and Innovation
Marymount University, Arlington, VA 22207
srk25218@marymount.edu

Online gaming attracts people of all ages and demographics with more than 2.95 billion people reporting that they play online videos games at least once a week. With entertainment as the primary aim, online games are usually not regarded as tools for building social or leadership skills. However, a study conducted of online gamers finds that online gamers attribute their personal development in the areas of collaboration, communication, and leadership skills to playing online games.

Findings from this study show that self-identified gamers can connect online gaming skills with real-world problem solving as they find these skills essential for online game playing. The research also uncovers what skills learned through gaming, prepare individuals for leadership positions in business and corporations. This research has become increasingly important as the world shifts towards work in technological spaces with remote workforces employing new generations of workers who grew up playing video games. Finally, the paper makes suggestions as to how organizations can incorporate video gaming into the corporate culture to connect with and train teams for success.

Demystifying Long-Short Term Memory Model to Predict Stock Prices*

Student Paper Abstract

Li Zhang[1,2], Ruichen Kong[1,2], Hanyu Wang[1,2], Vitaly Ford[1]
[1]Arcadia University, Glenside, PA 19038
fordv@arcadia.edu
[2]Jiangsu University, Jiangsu, Zhenjiang, China

Correctly predicting stock prices is one of the most effective ways to make profits. "Long Short-Term Memory" (LSTM) is a sophisticated machine learning algorithm containing a "memory cell" that can hold the information in memory for a long period of time. Our first research question is whether LSTM is usable in predicting the stock market. Furthermore, if LSTM cannot predict the stock market on its own, can we improve it to facilitate better stock prediction? If it works, can we improve the accuracy of its predictions? Therefore, we posed the second research question: can we improve the LSTM model beyond just changing the LSTM's parameters, such as the number of hidden layers, number of hidden nodes per layer, epochs, and batch size? Can we use other techniques to improve the LSTM model besides changing its parameters?

First, we tested the short-term prediction effect of LSTM with two methods. One is finding how many days the predicted trend matched the real stock market movement. The other one is finding how many days the predicted price is between maximum and minimum prices. We discovered that the results were not satisfactory in both short- and long-term predictions. Hence, we concluded that the traditional LSTM is not suitable for predicting stock price, and it should be improved. To determine the best outcome, we experimented with LSTM parameters to determine the best set of configurations. The essence of this work is the concept of swarm optimization that facilitated slight improvements of the traditional LSTM model as it helped us to identify the best fit.

Our research confirmed that the use of the traditional LSTM model alone is meaningless for stock market forecasting. Even slight improvements to LSTM do not produce satisfactory results.

Portfolio Optimization: Theory and Practice*

Student Paper Abstract

Yujie Fan, Haocheng Fang, Yuping Luo, Weihong Ni
Arcadia University
Glenside, PA 19038
{yfan_02, hfang, yluo_01, niw}@arcadia.edu

Recently, there have been spikes and dips in the stock market overall. In view of this trend, with the main idea of maximizing return while minimizing risks, our project aims to calibrate a model to construct optimal portfolios using historical stock data over a fixed time horizon. Initially, we will briefly introduce the Markowitz modern portfolio theory [1], covering topics such as efficient frontier, capital asset/market lines. As an application, we then implemented the Quadratic Programming Solver in R and obtained optimal investment allocations based on the top 30 most valuable companies in SP 500, using a series of risk aversion coefficients. Furthermore, we evaluated the performance of the proposed portfolios and that of the SP 500. Finally, we conclude by making suggestions for potential improvements of our work.

Reference

Best, Michael J. Portfolio Optimization. CRC Press, 2010.

Risk Analysis of Legacy Systems on County Government Assets*

Student Paper Abstract

Joseph James Villani. Jr.
Marymount University
Arlington, VA 22207
jjv40342@marymount.edu

County governments in the United States provide standard services and critical infrastructure spanning hospitals, prison systems, airports, energy, water supplies, and emergency services. Counties have provided these services for decades; some of them continue to be delivered today by legacy IT and OT systems. Therefore, local government preparedness to mitigate and respond to cybersecurity incidents is paramount for critical services and infrastructure. Unfortunately, the commonality of services and the presence of dated systems make counties an attractive target for cybersecurity attacks. Hence, legacy systems possess an increased likelihood of cybersecurity incidents with costs beyond disrupted services or budgetary, including operational failures that could trigger cascading events and loss of life. This research includes survey data from a striated sample of Pennsylvania county governments and publicly available data. With more than 70% of the counties relying on legacy systems, risk mitigation strategies warrant a deeper dive. This poster provides a holistic review of legacy system definitions, concerns, and prevalence within Pennsylvania's county government. In addition, it also offers recommendations to mitigate legacy systems exposures and risks through inventories, classifications, and regular roadmap planning. These findings are relevant and applicable across the Counties of the United States.

On Building a Mind: Replicating a Neural Network Model of a Neuron*

Student Paper Abstract

Daniel Matthew Foreacre and Jingnan Xie
Millersville University
Millersville, PA 17551
{dmforeac,jingnan.xie}@millersville.edu

This project is a replication of work published by David Beniaguev, et al. entitled "Single cortical neurons as deep artificial neural networks." Published in the journal Neuron in 2021, the paper describes the authors' creation and training of a temporal convolutional deep neural network (TCN) to accurately model a single L5PC neuron. Our replication began with the code written by the authors posted to public repositories, including the pre-trained Keras models and original training data. One model was selected, and the code was examined to determine the structure of the neural network and the files used to train and test the models. To generate a new data set for testing, we turned to the NEURON program, created to simulate biophysical models of neurons, and used by the authors to generate their data. While the exact model was not included with the published code, the original source was cited, and a model was provided through ModelDB. This model was modified via the HOC language to approximate the one used by Beniaguev and was used to generate a similar data set with inputs and corresponding outputs. After consolidating the data into a form readable by the TCN model, it was tested, and accuracy values were recorded. Given the difficulty in recreating the NEURON model faithfully, the results show that the TCN model can accurately replicate the results of a complex neuron simulation given the same inputs. Upon further research into the phenomenon of long-term potentiation, we realized that as a static classifier a TCN would not be able to simulate the same behavior. This, however, leaves an exciting avenue for future research.

Fusion-Plasma Response Modeling Through Neural Network Machine Learning*

Student Paper Abstract

Sanika Kothari[1,2], Shira Morosohk[2], Eugenio Schuster[2]
[1]Parkland High School
Allentown, PA 18104
sanikakothari05@gmail.com
[2]Lehigh University
Bethlehem, PA 18015
{smm418,schuster}@lehigh.edu

A neural network is a system with interconnected nodes called 'neurons,' which communicate an output by processing it through an activation function. It is based on the human brain, which means it trains itself to reduce errors. The objective of this project was to model efficient temperatures for nuclear fusion reactions to occur for maximum energy output. Nuclear fusion is a theoretical concept today but with the research done, a path to making it experimental can be paved. The neural network was developed through Anaconda Distribution and uses the software Spyder to code in Python. The source of the output is an activation function, which in this case was Te $=(10^5)(\mathrm{Ip}^{0.85})(\mathrm{P}^{0.3})(\mathrm{ne}^{0.61})$, where Te represents the temperature of the electrons. The actual output of the function is 4.19x10^{-5} eV to 4.19x10^{-4} eV and represents the value the neural network tried to predict. After being run for 1000 epochs, the loss function of the neural network was measured. The loss function measured how far the predicted values were off regarding the actual output. The decrease in loss function signified a negative trend meaning the accuracy of the outputs increased. This research has a great societal impact because it will allow the world to develop clean energy. Other impacts include the development of artificial intelligence and an efficient way to find a way to clean and sustainable energy.

Research Analysis & Topological Structures: Behavioral Analysis Using Machine Learning Techniques Data Analysis*

Student Paper Abstract

Luke Cossman, Lucas Acosta-Morales, Caitlin McNamara
Harkiran Bhullar, Carl Harrison Tyce
DeSales University, Center Valley, PA, 18034
{lc2027,la8080,cm1515,hb9633,ct8275}@desales.edu

Based solely on human experience, it is difficult to attribute behaviors such as pain and anxiety to mice. This project addresses this difficulty through behavioral analysis of mice using machine learning and data analysis. This involves two main parts. The first takes data in the form of videos of mice, extracts positional data for multiple parts of a mouse (ex: head, tail base, body center) using the machine learning program DeepLabCut by the Mathis Lab. We augment the dataset by including directional information, determining total distance traveled, etc. Afterwards, we identify prominent features via persistent homology. These features are not physical positions or characteristics like a location in a cage or a foot, but rather abstract mathematical representations of key behaviors. We reduce this dataset to the 12 most notable features (about 30% of the original data) through Kernel Density Estimation. In the next part of the project, we create and train a machine learning algorithm using the reduced dataset. This produces a computational classifier capable of detecting the behavioral state (ie pain or no pain) in novel test subjects, thus opening up a new level of objectivity in, and potentially improving the quality of, rodent research.

A Pedagogical Approach to Leveraging Gamification for Cybersecurity Awareness in Higher Education*

Student Paper Abstract

Jeremy Alexander Gilmore
Kutztown University of Pennsylvania
Kutztown, PA 19530
jgilm180@live.kutztown.edu

In a time when the Internet is a dominant tool used by billions of people, cybersecurity becomes a necessary topic that needs to be treated with the utmost importance. "Bad actors" have been known to come up with crafty and sophisticated attack vectors or approaches to catch unsuspecting victims and hijack their hardware, compromise their system, or steal personally identifiable information (PII), among other things. One such group of victims is college students. Between time spent on homework, research, extracurriculars and personal activities, students are among those who spend the largest amount of time on the Internet. Not only that, but information systems used by universities contain troves of sensitive data, such as social security numbers or bank account information. Information Security Buzz (2018) has done extensive research noting a history of massive data breaches within these systems affecting hundreds of thousands or even millions of students. Because of this, students must become well-versed in following good principles and avoiding bad habits, so that their online experience will be as secure as humanly possible. One unique pedagogy is to train students about online safety through the use of gamification. When game mechanics are applied within a non-game context, it offers many benefits as an educational method. This poster goes into the process of applying the four phases of the game development life cycle (GDLC) to develop a web-based trivia gamification to educate college-level students about cybersecurity concepts.

How Can accessiBe's ADHD-friendly Profile, Content And Color Adjustments Improve Reading and Comprehension for Philadelphia College Students With ADHD?*

Student Paper Abstract

Hafsah Abdul-Malik, Andrea Wentzell, Susan Ceklosky
Chestnut Hill College
Philadelphia, PA 19118
{abdul-malikh,wentzella,cekloskys}@chc.edu

Prioritizing accessibility in websites for users with adult attention-deficit hyperactivity disorder (ADHD) is crucial. Without accessible design, adults with ADHD are restricted from engaging with websites. Fortunately, enhancements such as bold color contrast between text and backgrounds, distinctive headings and subheadings, and consistent site navigation when integrated into web design can boost productivity and improve the user experience for adults with ADHD. Accessibility not only expands consumer reach by accommodating disabled users, but it also aligns with the Americans with Disabilities Act (ADA) and Web Content Accessibility Guidelines (WCAG). The ADA establishes guidelines that prohibit discrimination and the isolation of online services based on an individual's disability, while the WCAG enforces a list of standards that web content accessibility must fulfill to accommodate disabled users around the world. Although many websites follow the ADA and WCAG, more than half of the internet fails to meet minimum accessibility standards that assist users with disabilities in navigating the web. It is important to understand the content and experience that users need to use the internet effectively, as obstacles may impede and limit the accessibility of the websites they visit.

The focus of this study is to assess how ADHD accessibility features improve the reading and comprehension of Philadelphia college students with ADHD. Two websites will be created using the web content management system, WordPress. One site will deploy accessiBe, an artificial intelligence plugin that seeks to create a more ADA and WCAG-compliant web experience. This site will implement accessiBe's ADHD-friendly profile for more focus by displaying a spotlight that follows the user's cursor, adjustable font sizing to increase or decrease text size, and color adjustments to change the font color. The second website will provide a baseline ADHD accessibility experience in WordPress, such as minimalistic design to reduce distraction. Each site will supply different ADA-compliant and WCAG support, however, the content on both sites will contain the same ASL lesson plan. A tutorial will be provided where the viewer can follow along and learn short signs such as hello and thank you.

In total, participants will be asked to use the two websites to learn eight ASL signs. They will then be asked to take a quiz to reflect on the material learned. Lastly, they will complete a questionnaire to compare their experiences using each site. Content retention and memory recall of American Sign Language (ASL) will be assessed in this study as will the effectiveness of accessiBe's ADHD-friendly profile, content, and color adjustments in improving reading and comprehension.

Data Security and Privacy Issues in the Internet of Things Ecosystem*

Student Paper Abstract

Syed Rizvi, Dylan Bender, Devon Claudino
Joshua Krish, Benjamin Weiland
Department of Information Sciences and Technology
The Pennsylvania State University
Altoona, PA 16601
{srizvi,djb6761,dcc5494,jsk5590,bmw6117}@psu.edu

The Internet of Things (IoT) is a large network of devices that are connected and able to communicate together. Everything from smartphones and computers to home devices and cars uses this system of communication. Over the last few years, the need for the interconnection of computer systems has evolved to the point where almost every new device now uses the IoT to communicate. However, as beneficial as this might be, it has created new avenues for cybercriminals to attack networks and online infrastructure. Cyber attackers are now able to disrupt the use of certain devices like smart cars and garage door openers, therefore, potentially causing serious harm or death for the victim. In this paper, we will be discussing the issues with cross-platform device security and the certain vulnerabilities that many devices on the IoT are now being exploited by cybercriminals. Many devices that enter the IoT are not secure and are easily manipulated by an offender remotely. Cyber security must evolve alongside these new devices to ensure that users are not at risk of becoming a victim of attacks.

Data Analysis for Video Game Reviews*

Student Paper Abstract

Shuyao Cai, Shunyi Zhang, Lin Zhu
Arcadia University
Glenside, PA 19038

The Video game industry is currently one of the largest in the global market, and the time people spend playing video games has steadily increased. Given this trend, we are curious about players' viewpoints on popular video games. For example, what are the sentiments and main topics of their game reviews? Moreover, we are interested in the analysis of the people who played video games for unusually long hours. More importantly, we want to investigate whether there are some significant differences between different game categories.

We collected video game review data of eleven popular video games on the Steam platform and classified them into five categories based on Steam's classification methods: shooter/First Person Shooter (FPS), strategy, Role Playing Games (RPG), sandbox/open world, and sports. We conducted exploratory analysis, sentiment analysis, and topic modeling of the game review data at the individual-game level as well as game category level.

Through sentimental analysis, we found that the strategy and shooter/FPS categories show slightly more positive sentiments and the sports category shows slightly more negative sentiments. Our exploratory analysis shows that players tend to spend significantly more hours in shooter/FPS and strategy games, and these two categories of games contain a significantly higher percentage of outliers, who spend extraordinarily long hours gaming. We also found that the outliers had fewer friends despite their extremely long playing hours.

Finally, we created bigram word clouds and conducted topic modeling for each category. We found that shooter/FPS and strategy games are the two most popular game categories. Topic modeling for these two categories shows some interesting discoveries, such as players' enthusiasm about the games, the game community, time players spent on the games, and the impact of the games on their lives.

Facilitating Internet of Things (IoT) Experience in Computer and Information Systems Education*

Poster Abstract

Abdur R. Shahid
Robert Morris University, Moon Township, PA 15108
shahid@rmu.edu

Over the past few years, with the advancement of sensing technologies, wireless communication, cloud computing, data analytics, machine learning, conversational artificial intelligence, and embedded systems, we have observed a dramatic increase in the research effort to turn physical objects into smart devices and enable them to talk to each other through the internet. The network formed by these different smart devices is the Internet of Things (IoT). We are starting to see its applications in various industries, including healthcare, home automation, automobile, manufacturing, retail, and education. It is a multi-billion-dollar industry projected to grow into the trillion-dollar range. However, there is a significant gap between industry demand and the supply of professionals from the universities. Therefore, it is highly expected that graduates from different Computing-related fields will fill the void. Motivated by the problem of introducing IoT education in an undergraduate Computer and Information Systems (CIS) program, we developed a proof-of-concept IoT curriculum. We incorporated it into an existing Advance Java programming language course. This approach has several benefits. First, it avoids creating an entirely new course on IoT. Second, it allows the students to be engaged in developing a solution for a real world-centric problem under physical constraints. Third, it helps us to attract a wider audience. Three groups of students created java programs to solve problems for raspberry pi-based intelligent cars. All projects followed the Bridge21 Pedagogy Model, emphasizing teamwork, learning by doing, and technology-mediated project work. In addition, the projects introduced several critical concepts to the students, including Linux systems, system setup, scripting, and working with multiple programming languages (Java and Python).

Wireframe to Launch Day: Experiential Learning Development Through Community Outreach*

Poster Abstract

Andrea Wentzell
Chestnut Hill College, Philadelphia, PA, 19118
wentzella@chc.edu

In the competitive job market of Computer Science, having the right experience can be crucial. Therefore, experiential learning, such as internships, is essential for students to gain real-world skills before graduation and is a requirement for all Computer Science majors at Chestnut Hill College (CHC). However, with the move to remote learning and work during the pandemic, these valuable internship opportunities became increasingly difficult for students to locate.

Creating on-site internships with local nonprofit and community organizations was necessary to ensure students graduated on time. As a result, a pilot program was launched in Spring 2021, with several small nonprofits needing website design and development assistance. Students worked one-on-one with clients but also with professors.

The pilot resulted in successful experiences, teaching students the skills to develop a website from wireframe to launch day while providing portfolio collateral to share with potential employers. At the same time, students developed further skills in problem-solving issues, communicating with clients, and working independently.

The pilot has since been expanded and encompasses additional opportunities beyond website development, resulting in multiple iOS and Android applications for CHC students, various websites, and overall community engagement. Clients have included education nonprofits, churches, community outreach organizations, etc.

This poster will discuss the overall pilot and development of the program, along with the student outcomes.

Project-Based Learning: Teaching Mobile App Development in Capstone Courses*

Poster Abstract

Weidong Liao
Shepherd University
Shepherdstown, WV 25443
wliao@shepherd.edu

Mobile apps have become an integrated part of our daily life, especially for today's college students. For students who are in the computing majors, it would be exciting and motivating if they could put their knowledge and programming skills together and develop mobile apps. This poster presents and discusses our experience in teaching modern mobile app development in our undergraduate capstone class in the past few semesters. We first categorize and describe the mobile apps that our students have designed and implemented. Choices of programming languages and development platforms that our students have adopted to design and develop their mobile apps are presented and compared. Model apps in each category are then demonstrated with sufficient detail. We will also outline the business aspects of mobile app development at the end of this presentation.

This project is sponsored by the course development grant from NASA-WV Space Consortium.

Enhancing the Security of Real-Time Gang Tasks on a Multicore*

Poster Abstract

MD Amiruzzaman and Ashik Ahmed Bhuiyan
West Chester University
West Chester, PA 19383
{mamiruzzaman, abhuiyan}@wcupa.edu

As the world is becoming data-dependent, so does data saving into a backup server for future use. Saving data in real-time is a challenging task since the data needs to be protected and processed simultaneously. The TCP protocol provides the reliability of the data communication; however, three-way handshaking adds some overhead to it. Very few works focused on transferring data using the RTP protocol, especially considering resource constraint device-to-server and client- to-server models. We aim to develop an RTP-based real-time data backup system in this work. This protocol will compensate for the packet loss by introducing log files for both the client and server sides. Furthermore, contrary to the traditional protocols, we will improve the throughput by introducing a two-way handshake. Finally, we will integrate a real-time security enhancement scheme (during idle periods, mainly during client-server communication) without compromising the overall system performance.

The Need for In-House Development Never Goes Away*

Poster Abstract

Pipop Nuangpookka[1], Donna Schaeffer[1], Patrick Olson[2]
[1]Marymount University
Arlington, VA 22207
[2]National University
San Diego, CA 92123
{pipop.nuangpookka, donna.schaeffer}@marymount.edu, polson@nu.edu

Many educational institutions rely on web portals, with services provided by various vendors are available inside the portal. Authentication systems are integrated with directory tools and Single Sign-On mechanisms. This poster explains how one university upon discovering a breach, saw the need to develop a Java-based tool to prevent future occurrences of similar computer attacks.

Preventing breaches in important because various services (e.g., Learning Management Systems, Video Conferencing Systems, Human Resource Management Systems, etc.) available in the portal associate with different sets of users' data, some of which may be sensitive and personally identifiable. This, in addition to using secured source code and commercial security tools to keep the organizational servers and applications as highly secure as possible, a more internally specific tool that monitors specifically abnormal behaviors should be fundamental.

This case study describes the infrastructure, the discovery of a breach, the algorithms that were developed in-house, and a summary of benefits accrued.

A Weighted TFIDF Technique for Inclusion of Reposting Engagement on Social Media*

Poster Abstract

Maha Allehaibi and Samah Senbel
Sacred Heart University, Fairfield, CT 06825
{allehaibim@mail,senbels}@sacredheart.edu

Text analysis is a field of study in data science that analyses human-generated textual data generated through books, articles, reviews, social media posts and discussions among other sources. Text is initially converted to numerical data using the well-known "term frequency-inverse document frequency" (TFIDF) technique. TFIDF measures the importance of each word in a document relative to other words in the document and the whole data corpus. Data generated from social media has a unique nature: In addition to the text itself, a post on social media can have different types of engagement such as likes, replies, and reposts.

We modified the TFIDF technique to be better suited to social media posts, by considering the number of reposts. For example, on Twitter, a tweet that has zero retweets is not equal in value to a tweet that has thousands of retweets. We modified the TFIDF technique by considering the reposts by using a weighted factor that can be controlled. The weight of the original post was fixed to 1. A repost is given a weight that can vary between 0 (ignore reposts) and 1 (consider reposts as equal to original posts). The new technique was coded and tested on a dataset of 383,288 tweets.

We measure the effect of the weights on the TFIDF values using different weights ranging from 0 to 1 with a step of 0.1. We observed that the different weights strongly affect the TFIDF values, mainly be increasing the numerical range of values thus emphasizing the more common used (and retweeted) words with smaller TFIDF values and the rarely used words with much higher values. In further studies, other factors, such as likes, replies, and the number of followers will be considered into the TFIDF calculation to reflect the importance of a word through audience reaction.

3D Printing, Service-Learning, and Interdisciplinary Peer-Assisted Learning*

Poster Abstract

Jeffrey A. Bush and Sara Benham
Moravian University, Bethlehem, PA 18018
{bushj,benhams}@moravian.edu

Service-learning and peer-assisted learning (PAL) are two pedagogies that are used to help students engage with real-world problems, move beyond independent learning, and develop "soft" skills such as communication and teamwork. Service-learning is a form of experiential education in which students have a real-world client to which they provide a needed service but in doing so they achieve learning outcomes. Through PAL, peer modeling and peer mentoring are utilized for students to assist others to learn. Since 2020, graduate occupational therapy (OT) students and undergraduate 3D printing students at Moravian University have worked in teams to create custom 3D printed assistive devices to address real-life patient functional problems. During the multi-week project, the 3D printing students teach the OT students about the benefits, design process, and limitations of 3D printing while the OT students teach the 3D printing students about disabilities and physical conditions, and the associated functional limitations and problems. Together they create a custom device that is delivered and fitted to an actual patient. After the major teaching sessions, the students evaluate each other using a survey derived from standard teaching evaluations. At the end of the project, the students were given the Peer Teaching Experience Questionnaire (McKenna and French, 2011) and a modified Technology Acceptance Model (Davis, 1989) questionnaire to evaluate their beliefs on the use of the PAL model and the use of service-learning to create usable devices and encourage acceptance of 3D printing by OTs in the future. While there are only partial results from the 2020 and 2021 classes due to the pandemic, the 2022 results indicate that the interdisciplinary PAL model reduces anxiety in performance and increases communication although it may reduce the apparent learning value. The TAM results indicate increased willingness to use 3D printing in the real world.

The Visibility Project: Virtual Reality Experience for Racial Bias Training in Psychiatry*

Poster Abstract

Teanna Barett, Kennedy Thompson, Chidindu Alim
Howard University, Washington, DC 20059
{teanna.barrett,kennedy.thompson,chidindu.alim}@bison.howard.edu

Consistent research findings have shown a high rate of psychotic disorder diagnoses among patients of color, especially African Americans.The COVID-19 pandemic has highlighted this. Clinicians who focus on treating solely the presenting symptoms without giving consideration to the social, political, and historical context are destined to be inadequate. Mental health professionals are beginning to utilize cultural competence training but without an understanding of how biases negatively impact patient care, this training is inadequate. The Visibility Project is a virtual immersion experience with the goal of reducing bias and fostering empathy within the field of psychiatry.

Virtual Reality (VR) technology is traditionally used to illustrate various experiences for the user to witness first-hand. Currently, VR is most common in the gaming and entertainment industry. However, previous experiences demonstrating virtual reality can additionally be used as a tool for education and research in understanding health disparities; particularly as it relates to psychiatry. VR can help people of color by unveiling biases that others may hold against them in workplace or patient-doctor settings. The Visibility Project uses VR immersion as a medium to display social and racial bias commonly exhibited by older White male psychiatrists. The immersive experience is created in Unity to produce realistic environments and characters with assets for customization and avatar lip syncing. The differential diagnosis plan will be used to understand if participants exhibit changed behavior and empathy after interacting with the immersive experience. The project is the first of its kind for specifically combating racial bias in psychiatry. This paper describes the goals of this project and how VR is used to communicate and influence emotions of medical professionals that step into the shoes of their patients.

A TDA Approach of Analyzing Election Speeches with NLP Techniques*

Poster Abstract

Jici Huang
Department of Computer Science and Information Technology
Kutztown University of Pennsylvania
Kutztown, PA 19530
huang@kutztown.edu

In this poster we present new results on applying topological data analysis (TDA) to reveal the features of election speeches using natural language processing (NLP) techniques. We use latent dirichlet allocation (LDA) in NLP to preprocess the speeches into four main topics. For each topic, we apply persistent homology in TDA. We thus generate persistence diagrams for each speech in each categorization of the four topics. Persistent homology was first used to analyze nursery rhymes text datasets and extract topological features from the data by Xiaojin Zhu in 2013. Along the line of analyzing nursery rhymes text datasets, we advanced the algorithms for nursery rhymes and apply the new approach on the election speech datasets, which was comprehensively collected by UC Santa Barbara. The goal of our poster is two-fold. We show how to use persistent homology to analyze text datasets. We demonstrate out results by applying LDA and TDA techniques implemented in R to capture significant features, including patterns and themes in presidential election speeches during 2000-2020. We will also discuss potential future directions and implications for our work.

Enhancing the Security of Real-Time Gang Tasks on a Multicore Platform*

Poster Abstract

Ashik Ahmed Bhuiyan, MD Amiruzzaman
West Chester University
West Chester, Pennsylvania 19383
{abhuiyan, mamiruzzaman@wcupa.edu}

A real-time system is a computing system that must respond to the event generated by the surrounding environment within precise time constraints. In modern society, real-time system applications are widely deployed, such as chemical and nuclear plant control, automotive applications, flight control systems, smart grid, telecommunication systems, robotics, and space missions. However, these broad ranges of applications come with a trade-off, i.e., an exposure to various security threats. Moreover, shifting the real-time applications towards the multicore domain makes security more vulnerable. In this study, we propose a scheduling framework to enhance system security without hampering the time precision of real-time tasks. The security tasks will share the multicore platform with the real-time gang task, such that these security tasks (i) do not jeopardize standard execution patterns of the real-time gang tasks, and (ii) meet the desired monitoring frequency for threat detection.

Developing Basic Artificial intelligence (AI) and Machine Learning (ML) Solutions Using BeagleBone AI: An AI Hardware Learning Tool for Undergraduate Students*

Poster Abstract

Natalia Bell, Osei Hyiamang
College of BILT
Marymount University
Arlington, Virginia 22207
{nbell, o0h41091}@marymount.edu

BeagleBone® AI-64 is a credit card sized hardware system that can help students explore how Artificial Intelligence (AI) and Machine Learning (ML) can be used in everyday life by developing basic automation solutions in industrial, commercial and home applications. BeagleBone® AI-64 system is created by The BeagleBoard.org Foundation, a USA-based 501(c)(3) non-profit organization and uses the open source Linux approach. BeagleBone AI is a low-cost, fan-less single-board miniature computer (SBC), but considered to have the power of a full-on industrial computer workhorse. The users of BeagleBone AI and the Beagle™ community share supporting documentation and resources on the GitHub platform and are fully accessible online. The author will share device specifications, setup instructions, and the necessary details to create and develop in-class projects using BeagleBone AI. Additionally, the author will discuss open-source resources available to guide users on how to provide hands-on experience for undergraduate students. BeagleBone AI is an affordable solution for undergraduate introductory courses in AI, ML, hardware and automation.

Federated Learning-based Contraband Detection within Airport Baggage X-Rays*

Poster Abstract

Sajedul Talukder, Sai Puppala, Ismail Hossain
School of Computing
Southern Illinois University, Carbondale, IL 62901
{sajedul.talukder,saimaniteja.puppala,ismail.hossain}@siu.edu

The Transportation Security Administration (TSA) runs security checkpoints at airports in the United States to keep hazardous objects off. The TSA uses an array of X-ray and CT scanners at these checkpoints so that Transportation Security Officers (TSOs) can examine the contents of carry-on luggage. However, due to an increasing shortage of airport staff, it can be difficult to locate and identify all potential dangers. As a result, the TSA has recently become interested in automated detection methods based on deep learning that can help TSOs. Machine learning models, which are now used to check luggage for dangerous items, can lower the cost of a human operator, extract qualitative data from the images, or complement a human operator's decisions. However, the majority of these ML models train the data in a centralized learning way, which could lead to security concerns when transmitting data into the central server and privacy concerns when gathering data.

To address this, we suggest a Federated learning (FL)-based framework to find contraband in x-ray baggage security images, keeping the data set for each local airport private while aggregating only the updates from distributed trained models thus protecting user privacy. A federated learning system that can collectively train a decision model by pulling knowledge from many airports can be used to accomplish contraband detection. Since FL only gathers local gradients from airport users without access to their data, it can reduce the risk of collaborative training compromising their data privacy. We investigate a number of design choices, illustrate the effectiveness of our models on held-out evaluation sets, and explain how these systems could be used to automatically detect contraband in airports.

Prediction of Childhood and Pregnancy Lead Poisoning Using Deep Learning*

Poster Abstract

Sajedul Talukder, Sai Puppala, Ismail Hossain
School of Computing
Southern Illinois University, Carbondale, IL 62901
{sajedul.talukder,saimaniteja.puppala,ismail.hossain}@siu.edu

One of the most common and avoidable environmental health risks is lead poisoning. Through ingestion, inhalation, or contact with lead-containing materials, lead can enter the body. Lead exposure can impair muscle coordination, stunt bone and muscle growth, harm the nervous system, impair speech and language, and even induce seizures. Since there is no safe level of lead in the body, prolonged lead exposure can have negative effects on anyone's health, but children and unborn children are particularly at risk. Learning difficulties, developmental delays, behavioral issues, as well as a number of other unfavorable health outcomes, are all known to be exacerbated by childhood lead exposure.

There are several different sources and risk factors for childhood lead exposure in the United States. Studies have shown a correlation between lead exposure and factors like household income, minority or refugee status, Medicaid reliance, older homes constructed before 1978 with lead paint in poor condition, proximity to industry, and people working in lead-exposed environments like manufacturing, repairing, welding, or renovation jobs. The chance of lead exposure can therefore be estimated by looking at census data, housing data, household income data, data from the Adult Blood Lead Registry (ABLR), and data from blood lead testing during pregnancy. We propose a methodology built on deep learning to automatically identify children and expectant mothers likely to be exposed to lead, thus reducing childhood and prenatal lead exposure, delivering timely interventions to lower blood lead levels (BLLs), and improving health and developmental outcomes.

Algorithmically Generated Artwork*

Poster Abstract

John Mancini[2], Maya Newman-Toker[1],
Michael Wehar1[1], Alyssa Zhang[1]
[1] Swarthmore College
Swarthmore, PA 19081
{mnewman2,mwehar1,azhang4}@swarthmore.edu
[2] Independent Researcher
johnmancini30@gmail.com

We are the Algorithmic Art Research Group (Michael Wehar, Alyssa Zhang, Maya Newman-Toker, and John Mancini). For our summer research project, we investigated how computer algorithms can be used to generate artworks, as well as the potential value and use cases for such generated artworks. During this investigation, we designed and implemented nine algorithms that generate visual designs. Our algorithms, coded in JavaScript, are built from procedural processes that draw lines and basic shapes onto a digital canvas. These procedural processes were inspired by mathematical concepts, natural phenomena, and artistic trends, but contain many parameters to allow for image variation. We used our algorithms to build a dataset of over 1,000 algorithmically generated artworks, which we recruited reviewers to compare and rate. We plan to analyze the reviewers' ratings to find associations between different parameterizations of our algorithms and the response of reviewers to the resulting images.

African American History and Applied Data Science*

Poster Abstract

Gloria Washington
Howard University
Washington, DC 20059
gloria.washington@howard.edu

This initiative focuses on redesigning a pre-existing "Applied Data Science" course to teach data analytics from an African American History perspective. Both CS and non-CS majors took course to receive a Math/Statistics credit. Approval to redesign Applied Data Science had been obtained at Howard for teaching the Fall 2018 course and in the coming Fall semesters. The course was positioned around datasets that show the progression, and sometimes digression, of African Americans in the United States throughout history. Topics taught relating to the datasets included census data of Blacks in the United States before Emancipation Proclamation, United States public elections centered around Reconstruction and African-Americans in public office, African American migration, and politics and African Americans in the United States. This class allowed the student to use probability and statistics to tell a story about features they seek to highlight in African American History and present through a final project. An ethics component was also addressed as the the class progressed to showcase the negative and positive side effects of data analytics research. Students learned R and Python to complete their assignments in the class. Data Analytics is a highly-sought after skill that is required across a diverse range of industries. It is the hope that the course broadened students' computing knowledge base, and in turn enhanced their career prospects in the future. Additionally, we believe the data science skills taught in the course are almost necessary for underrepresented CS students to begin to address inequities in their own communities (Kokka, 2018).

A Novel Hierarchical Federated Learning with Self-Regulated Decentralized Clustering*

Poster Abstract

Sajedul Talukder, Sai Puppala, Ismail Hossain
School of Computing
Southern Illinois University, Carbondale, IL 62901
{sajedul.talukder,saimaniteja.puppala,ismail}.hossain@siu.edu

For collaboratively training machine learning models toward a world consensus model while observing privacy restrictions, Federated Learning (FL) is currently the most extensively used system. Without any participants disclosing their own data to the central server, federated learning enables widely dispersed IoT devices and sensors to develop a global consensus model. It is difficult to use them in a FL setting, though, because of the volume of data, communication overhead, latency, data redundancy, and non-independent and identical distribution (Non-IID) data. The FL server has access to more data, but the global model convergence is slowed down by high communication costs and prolonged latency. The cost of communications in FL is a major problem due to the large IoT or user-intensive scenes in 5G or future networks. Clustering local clients based on data similarity estimation among IoT nodes could be useful for reducing the communication overhead required to train the global model. During the clustering formulation, estimating data similarity is one of the main research challenges.

In this paper, we propose a novel hierarchical Federated Learning framework with self-regulated decentralized clustering. In particular, we construct data-similar application-aware clusters and hierarchies in order to minimize the communication overhead. To formulate the data similarity metric in a distributed fashion, we consider a combination of Euclidean distance and temporal and magnitude data correlation. By measuring the magnitude similarity and trend similarity among a long time series's elements, the distributed similarity can result in significant data reduction, perform faster model convergence and achieve high energy efficiency. In particular, it is demonstrated that compared to cloud-based Federated Learning, the model training time and the energy consumption of the end devices may both be decreased by introducing self-regulated hierarchical clustering with communication-computation trade-offs.

Reviewers — 2022 CCSC Eastern Conference

Chris Alvin .... Furman University, Greenville, SC
Ian Barland .... Radford University, Radford, VA
Karla Carter .... Bellevue University, Bellevue, NE
Dawn Childs .... Marymount University, Arlington, VA
Susan Conrad .... Marymount University, Arlington, VA
Mónica Costa .... Instituto Politécnico de Castelo Branco, Castelo Branco, Portugal
Lawrence D'Antonio .... Ramapo College of New Jersey, Mahwah, NJ
George Dimitoglou .... Hood College, Frederick, MD
Dan DiTursi .... Siena College, Loudonville, NY
Sheryl Drake .... Marymount University, Arlington, VA
Ian Finlayson .... University of Mary Washington, Fredericksburg, VA
Mike Flinn .... Frostburg University, Frostburg, MD
Reva Freedman .... Northern Illinois University, DeKalb, IL
Nathan Green .... Marymount University, Arlington, VA
Grigoriy Grinberg .... Montgomery College, Rockville, MD
Jenny Hu .... Valencia Community College, Orlando, FL
Ingyu Lee .... Troy University, Troy, AL
Jiang Li .... Hood College, Frederick, MD
Christian Lopez Bencosme .... Lafayette College, Easton, PA
Bob McCloskey .... University of Scranton, Scranton, PA
John McManus .... Randolph-Macon College, Ashland, VA
Linh B. Ngo .... West Chester University of Pennsylvania, West Chester, PA
Hoang Nguyen .... University of Illinois at Chicago, Chicago, IL
Emmanuel Nuakoh .... North Carolina Agricultural and Technical State University, Greensboro, NC
Sofya Poger .... Felician University, Rutherford, NJ
Veena Ravishankar . The University of Mary Washington, Fredericksburg, VA
Syed Rizvi .... The Pennsylvania State University, Altoona, PA
Rebecca Rohan .... Marymount University, Arlington, VA
Carolyn Pe Rosiene .... University of Hartford, West Hartford, CT
Amirreza Saba .... Marymount University, Arlington, VA
George (Larry) Sanders .. University at Buffalo, The State University of New York, Buffalo, NY
Samah Senbel .... Sacred Heart University, Fairfield, CT
Jim Teresco .... Siena College, Loudonville, NY
Alla Webb .... Montgomery College, Rockville, MD
Xu Wenjuan .... Frostburg University, Frostburg, MD