Examining Beginner Computer Programmers’ Procedural Logic using the Escape Room Model

Janiya R. Peters

Dr. Frances Trees

Department of Computer Science, Rutgers, The State University of New Jersey

Ronald E. McNair Postbaccalaureate Achievement Program
Table of Contents

I. Abstract ........................................................................................................................................... 3
II. Background ........................................................................................................................................ 4
III. Introduction ..................................................................................................................................... 4
IV. Literature Review
   i. What is an Escape Room? .............................................................................................................. 6
   ii. Why Do Beginner Programmers Struggle? ................................................................................... 9
   iii. Reducing Abstractions Through Microworlds ........................................................................... 12
   iv. Constructing Mental Models through Games ............................................................................ 14
   v. How the Escape Room Functions as an Experiential Learning Model .................................... 16
V. Methodology
   i. Research Questions ..................................................................................................................... 18
   ii. Subjects ....................................................................................................................................... 21
   iii. Setting ......................................................................................................................................... 22
   iv. Time ............................................................................................................................................ 23
   v. Limitations .................................................................................................................................... 23
   vi. Completing the Escape Room and Post-Survey ........................................................................ 24
   vii. Observer’s Role .......................................................................................................................... 24
   viii. Analyzing Survey Results ....................................................................................................... 25
VI. Anticipated Results .......................................................................................................................... 27
VII. Discussion ....................................................................................................................................... 30
VIII. Conclusion ..................................................................................................................................... 32
IX. References ....................................................................................................................................... 34
X. Appendix .......................................................................................................................................... 36
Abstract

This observational study uses the escape room as an experiential learning model to teach beginner computer programmers. The escape room is a timed challenge where participants race to find clues and solve puzzles to break out of a themed setting. This study places each of the six teams of introductory programmers into an experimental escape room. Each team has one hour to solve the room. Afterwards, participants complete surveys in which they numerically rank their team performance, construct flow chart algorithms, and record artifacts’ states and behaviors. Participants and teams receive a procedural score based on the accuracy of their flow chart models in the survey. Researchers will quantitatively determine whether teams who finish the room faster have higher procedural scores. Researchers will then qualitatively analyze whether participants with higher procedural scores better understand artifacts’ states and behaviors. This research can isolate gaps in beginner programmers’ understanding of object-oriented programming and constructing algorithms.

Keywords: canonical procedure, experiential learning, states and behaviors
Background

The researcher arrived at this work after observing a disconnect between peers and their undergraduate computing studies. Some peers expressed their frustration of representing intangible objects in a computer program, specifically storing their states as variables and writing their behaviors as functional methods. The researcher observed a way to bridge this disconnect by relating computing to real life experiences. Experiential learning, specifically the escape room, becomes a compelling tool to increase the accessibility of these topics. By engaging beginner programmers in solving real-life puzzles, they develop problem-solving skills and a strong procedural approach. The researcher seeks to use the escape room to ground students’ understanding of programming in procedural thinking.

Introduction

Beginner programmers struggle to learn object-oriented programming due to unfamiliar syntax and difficulty visualizing abstract concepts. Their biggest impediment, however, is their struggle to execute canonical procedure to support object logic. Canonical procedure is the natural sequence of steps executed to solve a problem. Object logic is the understanding of objects’ properties and functions in code. The escape room is a fast-paced challenge in which participants collaborate to solve puzzles, interpret clues and escape in time. Consequently, it is a unique model for examining students’ canonical procedure because the operations players perform, such as opening a lock with the correct code, can be modeled by flow charts. Also, the artifacts used to solve these puzzles are representative of objects with properties and methods in a computer program. This study offers the escape room as an experiential learning model to examine the effectiveness of students’ canonical procedure as well as how it supports their object logic.
The proposed study employs quantitative and qualitative methods to answer the following: (1) Do teams who solve the room faster demonstrate stronger canonical procedure? And, (2) Does a strong canonical procedure aid participants’ object logic? Each of the six teams of introductory programmers will have one hour to solve the researchers’ designed escape room. An observing researcher will watch the team’s interactions inside the escape room and rank their performance in areas like team cognition, situational awareness, and problem-solving. They will record these rankings in an observer’s template (see Appendix, p. 38). Teams will then receive a post-survey structured into four parts (see Appendix, p. 40). Part One asks individual participants to rank their performance in the same areas as the observer’s template. This section measures if a relationship exists between how the observer and participants’ rank their team’s performance.

Part Two asks individual participants to construct their own flow chart model based on a puzzle they encountered in the escape room. Researchers will count the number of correctly labeled routines and arrows to assign that individual a procedural score. They will average each participants’ procedural score to yield a team procedural score, which they will then rank on a scale from 0-32. The 0-18 range indicates weak procedure; 18-28 indicates adequate procedure; and 28-32 indicates a strong procedure. Researchers will record each team’s procedural scores and completion time in a table. Then researchers will track the two variables on a line graph and calculate $r$, the correlation between them. Tables, graphs, and calculations will be done in Excel.

Part Three asks participants to identify artifacts’ properties and behaviors. Researchers will examine the difference in answers between teams with higher and lower procedural scores. Finally, Part Four asks participants to indicate whether this experiment was their first escape room activity, as well as list major, course enrollment, and prior programming experience. Researchers will record these findings in an Excel bar chart.
The proposal offers the escape room as a potential model that will inform introductory computing curriculum. The escape room allows students to comprehend an algorithm’s routines-including inputs, processes, outputs, and decisions- through the interactive use of puzzles. In the future, students can code their own escape rooms and model artifacts’ states and behavior in object-oriented languages like Java. Finally, computing instructors can isolate weak points for beginner programmers through this experiential learning activity and devote additional class time and assignments to bridging these gaps in knowledge.

Literature Review

What is an Escape Room?

The escape room is a fast-paced challenge where players collaboratively problem-solve to “escape” an artificial setting. In Nicholson’s (2016) comparative study, the author surveyed 175 escape room facilities around the world to examine how different proprietors design their rooms and integrate puzzles into a narrative. The game host initiates and monitors the game. Before entering an escape room, the game host shares the theme and narrative which outlines the key details of the room the players are escaping. For example, if the theme of an escape room is a haunted house then a narrative could be:

Welcome to the House of Horrors, an abandoned mansion in the middle of nowhere. You all are locked inside with no way out. To escape, you must discover the fate of the missing owner and find his only keys to the house. But, beware of ghoulish elements and floating objects.

This narrative clearly articulates a premise and a goal. The premise builds on the “haunted house” theme to detail what players will encounter such as a “missing owner”, “ghoulish elements”, and “floating objects”. The goal is two-fold. In other words, it is essential for the team to discover what happened to the mansion’s owner in order to locate his keys.
After receiving the narrative, players enter the room and have one hour to escape. The

game host can communicate a certain number of hints with players through video from outside the
escape room (Nicholson, 2016, p.14). Inside the room, players encounter artifacts, objects in the
room that correspond with the theme and narrative (see Figure 1, p. 36). Some artifacts contribute
towards solving puzzles. Other artifacts are “red herrings”, or misleading objects that players
believe contribute to a solution but do not. Puzzles are an integral part of the escape room that
progress the narrative: solving one puzzle leads players to the next until they have solved the final
puzzle that leads them to their goal. Whereas, Nicholson (2016) explores the different types of
puzzles proprietors incorporate, this proposal will define how a puzzle is structured and how it
functions inside an escape room.

A puzzle can be conceived as an information-processing model segmented into inputs,
processes, and outputs. Figure 2 depicts a jigsaw puzzle. A jigsaw puzzle is a disassembled picture
composed of oddly shaped pieces that fit together in a specific orientation. The puzzle pieces from
the box represent the inputs, the data needed to run a process. The act of spatially arranging the
puzzle pieces to form a recognizable image represents the processing of data to produce a new output. In this
example, the output is the final assembled picture. The jigsaw puzzle or
joining of a solid item appears in 40% of surveyed escape rooms (Nicholson, 2016). In this
example, the primary processing involves spatial reasoning to determine how the pieces, or inputs,
fit together. The act of transforming data from one form into another is an integral part of puzzles.
Different puzzles that engage different types of processing. 43% of puzzles involve deciphering images within larger pictures, 53% include counting, and 58% of puzzles require communication between players (Nicholson, 2016, p.11). Each of these puzzle types prompt players to execute an operation to solve a puzzle. These operations engage various physical and cognitive skills. In puzzles requiring players to decipher images, players may engage their visual perception skills. In puzzles requiring players to count, players engage their number sense skills. In puzzles requiring players to communicate, players engage their verbalization skills.

Some puzzles require that players verbalize their actions to reach a solution. Verbalization is important in distributed escape rooms. These specially designed rooms require teammates to communicate via visual and audio links to escape from two physically distant rooms (Shakeri, Singhal, Pan, Neustaedter, & Tang, 2017, p.115). The physical separation of the teammates makes communicating a challenge. Some players will excel at certain puzzles more than others. For example, players who possess strong language skills will dominate linguistic puzzles and players with strong mathematical skills will tackle numerical puzzles. The diversity of puzzles engages a wide array of players’ skills.

Solving a puzzle in an escape room is complicated by the experiential environment. First, an escape room may hide a puzzle’s inputs from players. This constraint forces players to consider what inputs they need and how to acquire them. For instance, they may obtain the inputs by physically searching the room for jigsaw pieces or deciphering a clue to learn where they are located. Second, inputs may not be tangible objects. Suppose players in the House of Horrors escape room receive the clue: “The answer stares back at you.” In this example, the input, or data that must be processed to get the desired output, is the information from the clue. The process is the cognitive analysis of this information: considering and searching for all artifacts that “stare back” or reflect an image to a viewer. Players narrow this list to a few artifacts in the room that
match this requirement (i.e. a hand mirror, crystal ball) and stop searching when they locate an item that fits the clue. Finally, players locate a hand mirror with a code inscribed on the glass. The hand mirror is the output of this process. These constraints make the escape room an even more hands-on, experiential learning environment.

This proposal defines team success as the team’s ability to escape their selected room within one hour. Team success may also be referred to as status of completion, wherein completion indicates a team has escaped the room and non-completion indicates a team has not. Nicholson (2016) indicates that teams’ average completion rate at surveyed facilities is 41%, though actual completion rates vary by region (p.15). The primary goal of analyzing team success is to correlate team completion rates with the strength of their procedural logic. This proposal will explore procedural logic in further depth.

**Why do Beginner Programmers Struggle?**

Before examining the escape room as a research model, the proposal will explore how beginner programmers struggle. Eckerdal, Thune, & Berglund (2005) interviewed fourteen first year undergraduate students about their experiences in a beginner object-oriented programming course. She organized their responses into five hierarchal categories of understanding. Categories one, two, three represent an incomplete understanding of programming. Categories four and five represent the most comprehensive understanding of programming. According to Eckerdal et al. (2005), nearly 50% of students interviewed expressed an understanding described in category two:

> As above, and in addition learning to program is experienced as learning a way of thinking, which is experienced to be difficult to capture, and which is understood to be aligned with the programming language. (p.137)

Most students did not articulate an understanding aligned with ‘programming thinking’. ‘Programming thinking’ is defined by category four:
As above, with the difference that learning to program is experienced as learning a way of thinking which enables problem solving, and which is experienced as a “method” of thinking. (p.137)

The differences between these two understandings are subtle but have major implications on how two students view the same coursework.

Category four emphasizes programming as a method of thought that engages individuals’ abilities to problem-solve and logically construct solutions. It implies that ‘programming thinking’ is a “systemic way of thinking” that must be developed over time with continuous practice (Eckerdal et al., 2005, p.140). Category two suggests that students believe ‘programming thinking’ is an inaccessible and specialized method of thinking. In one interview, Student D expressed, “it feels as some people just understand programming…..but I also think that some people who have been programming before have probably learned to think like that” (Eckerdal et al., 2005, p.138). Student D, like many students whose understanding aligns with category two, are confused as to how some people learned to program. These students only know that those individuals have some prior experience. These answers reveal a disconnect between novices and programming, which causes novices to struggle.

Eckerdal et al. (2005) relates the disconnect between novices and programming to Piaget’s ‘process-object duality’ theory. According to the theory, there are two ways to conceive of a mathematical notion: as a process and as an object. As a process, a mathematical notion is conceived “as a potential rather than an actual entity, which comes into existence upon request in a sequence of actions” (Sfard, 1991, p.4). This is called process conception because the learner understands the subject matter through a dynamic, multi-step order of operations. As an object, the learner conceives the mathematical notion “as if it were a real thing- a static structure, existing somewhere in space and time” (Sfard, 1991, p.4). This is called object conception because the
learner understands the subject matter by its features and behaviors, including how it appears and acts. Process conception and object conception are the two components of process-object duality.

Process conception can aid object conception. Dubinsky (1991) attributed the passage from process to object conception to Piaget’s *reflective abstraction* theory. This is when the learner distinguishes between a mathematical notion’s process and content before transferring the learned processes onto the content being learned (Dubinsky, 1991, p. 98). When a learner develops strong process conception, they can reach object conception in which they comfortably recognize an abstract concept or solution by its structural features. Consequently, when students lack adequate process conception, they struggle to achieve a higher, abstract form of learning. This is the source of struggle for beginner programmers that Eckerdal (2005) observes in her study.

To achieve object conception, a learner must achieve process conception. Orit Hazzan (2003) explored how learners in mathematics and computing invoke canonical procedure to tackle problems and reduce abstractions. Canonical procedure is “a procedure that is more or less automatically triggered by a given problem….because prior training has firmly linked this kind of problem with this procedure” (Hazzan, 2003, p.108). When a learner absorbs new material, the desired solution is often too abstract for them to comprehend. Canonical procedure emphasizes a standardized set of steps to reach the solution rather than the solution itself. Canonical procedure is a natural approach to accomplishing tasks. Suppose a student must bake a cake. The student knows vaguely how the final product should taste- creamy, soft, and light. But the student needs a recipe. By following the recipe in a sequential order- cracking the egg, stirring the milk, preheat the oven, baking the cake for an allotted time- the cake starts to form. The solution or ‘object of content’, as Dubinsky calls it, starts to take shape.
Beginner programmers often struggle due to their approach to programming. Eckerdal et al. (2005) is concerned that students who do not reach an understanding of ‘programming thinking’, represented by category four, do so because they lack a strong canonical procedure. Therefore, programming feels too abstract and inaccessible. Escape rooms offer a solution to improving canonical procedure. The timed element of an escape room compels students to reduce large into manageable routines. It is more natural to follow procedures implied by a puzzle and realize the features of the objects (artifacts, clues, solutions, etc.) as they are called upon during the procedure. In the methodology section, the proposal will further define how to measure the strength of canonical procedure.

**Reducing Abstractions through Microworlds**

Microworlds are a way to address beginner programmers’ learning needs. Pears et al. (2007) indicate that students struggle to visualize programming abstractions, particularly algorithms and data structures because they have “no obvious graphical form” (p. 209). Microworlds, virtual interactive programming environments based on real-world or fictional narratives to teach students program behavior, can simplify these abstractions. The physical metaphor underlying microworlds reduces abstractions into concrete problems and bridges students’ understanding of how an algorithm executes in real life with how it executes in code (Xinogalos, Satratzemi, & Dagdilelis, 2006, p.150). The introduction of “Alice”, a programming microworld based on physical metaphors, further helps students.

“Alice” is a microworld that teaches students how to program real-world objects in a virtual simulation through “scripts” or human-readable code. Scripts allow students to command objects through simple methods, or a block of procedures the computer program calls and performs on an object, without agonizing over the syntax of a specific programming language. Advantageously,
Alice allows users to command objects and watch their behaviors execute on screen *without* using variables (Cooper, Dann, & Pausch, 2000, p. 116). Programming variables represent changeable values stored as particular data types in computer memory. The idea of intangible variables occupying “space” in computer memory can confuse beginners. Microworlds like Alice uniquely address beginners’ needs by simplifying abstract concepts.

Cooper et al.’s (2000) study conducted on new programmers’ interactions with Alice has both its limitations and insights. The study’s narrow sample size raises questions as to whether Alice would have been as successful with students of different backgrounds. While Pears et al.’s (2007) study doesn’t address the correlation between class and education either, it does pull insights from a large pool of researchers from the United States, Sweden, Finland, Denmark, and the United Kingdom. Nonetheless, the implementation of Alice in Cooper et al.’s (2000) study supports Pears et al.’s (2007) approach towards visually stimulating, hands-on learning.

Alice’s minimalist, human-friendly environment aids students’ cognitive representation of abstract concepts, an area Pears et al. (2007) observed to be a significant struggle for new programmers. In addition, Alice as a programming microworld is similar to escape rooms because both rely on physical metaphors to actualize their environments. Just as Alice features a range of animated environments based on real life, an escape room can be constructed around multiple themes, like a haunted house or museum. The physical metaphor is a way to ground students’ and players understanding of object properties and methods. With Alice, students can see an object’s methods execute onscreen. For example, they can command a helicopter in Alice to “Move”, “Turn”, and “Roll” and the helicopter will visibly perform these actions (Cooper et al., 2000, p.111). In a similar manner, the escape room allows players to interact with artifacts and develop knowledge about their methods through visible interactions.
Constructing Mental Models through Games

Players who develop a common model of their escape room tend to collaborate more effectively. Shakeri et al. (2017) defines this phenomenon as having a shared mental model or a “shared mental representation of their situation, which includes knowledge of the tools available to them, the goal of their task, and what other team members know” (p. 117). Shared mental models integrate what individual players know about the room’s artifacts with other players’ knowledge of these items. This integrated knowledge facilitates how players interact with the room. Pan, Lo, and Neustaedter (2017) adds that players with a shared mental model are more efficient and synchronized (p. 1355-6). Having a shared mental model does not guarantee that players think alike. Rather, players have uniform understanding of how the room works based on similar experiences.

Shared mental models aid team cognition. Fiore and Salas (2004) distinguishes team cognition as “the seamless execution of coordinated behaviors” such that individual members’ skills and contributions are combined into a “functional entity” (p. 236-7). Players with a shared mental model will develop team cognition more easily. If players have a mutual understanding of a problem or how an object functions from similar past experiences, then they can more easily work together. Thus, players better collaborate and communicate when they enter the escape room.

Students may develop shared mental models from real life experiences to reduce abstractions. Hazzan (2003) discusses how beginner programmers are already familiar with some data structures, like queues and stacks, before taking a formal class. Data structures are a system of organizing data in a computer. In real life, a queue can be represented by a customer check-out line in a store, where the first customer to enter the line is the first customer to leave the line when a new register opens. Similarly, a stack can be exemplified by a tower of lunch trays in a cafeteria:
the last tray added to the tower is also the first tray removed. Students develop a strong mental model of data structures and how they work based on these real-life examples.

While mental models are helpful, they can oversimplify the nature of these data structures. Peled (2001) discovered some students in a Visual Basic course believed that an array with fewer than two elements is not an array (as cited in Hazzan, 2003, p. 105-6). After exposing one student to a one-element array, the student expressed to Peled (2001), “So, there is an array with only one element. But why?...What purpose does it serve?” (p. 106). First, the student confuses the structure of an array with its implementation. An array is structured to hold multiple items of the same data type, but it is possible for it to hold only one item at a time. Second, the student is relying on experience; intuitively, individuals use lists to store, organize, and recall more than one item. The student does not understand why an array, or list with one item, is necessary. Mental models are effective in reducing abstract concepts to relatable ideas like the queue and stack. But they can also constrain students’ understanding of how these concepts are structured and implemented, as Peled (2001) demonstrated.

Incorporating games into a programming curriculum is another way to reduce abstraction. Games are an accessible teaching mechanism because students are familiar with the rules and structure of popular games. For example, Hangman requires players to guess letters to complete the blank word or phrase. It is great for modeling one-dimensional arrays and is naturally algorithmic because it distributes turns to players based on a sequence of player inputs, conditionals, and procedures (Drake & Sung, 2011, p. 619-620). The blank word or phrase is represented as an array of characters. Each blank space in the word represents an index in the array. Each correctly guessed letter is assigned to the appropriate index. This familiarity of Hangman, including its rules and structures, allow students to create a shared mental model of how an array
is structured and functions. Because escape rooms are also a type of game, they have potential to use and create these shared mental models, too.

Games are a useful tool for beginner computer programmers because they engage them in mutual experiences. Familiar rules and features facilitate effective shared mental models. In a team setting, these models allow for stronger team cognition. In programming, these models allow beginners to engage with objects on both an abstract and concrete level such that they can evaluate their algorithms based on how they have played the game in real life. These shared mental models become increasingly important as the study begins to examine students’ procedural logic.

**How the Escape Room Functions as an Experiential Learning Model**

In examining past alternatives, this study examines the puzzle structure of the escape room model to reduce abstractions. Nicholson (2016) conducted a comparative study examining how various escape room facilities around the world construct their room and integrate puzzles into a themed narrative. He outlines the three primary puzzle types found in escape room facilities: open, sequential, and the path-based type (see Figure 3, p. 36). These puzzle types parallel concepts a new programmer would encounter in an introductory class.

The open puzzle, for example, allows players to work simultaneously on multiple puzzles which all construct the solution to a larger puzzle. In object-oriented programming, this design would be like the creation of a class where a suite of tools called methods help construct the behaviors of that class. The sequential puzzle, on the other hand, forces players to solve puzzles in a linear order. This design closely parallels procedural programming where a series of statements and functions are processed in order to execute a goal. The path-based puzzle is a hybrid that integrates the sequential and open puzzle structure. Players can solve separate puzzles at once, but each puzzle requires a series of steps be linearly executed. Because the escape room’s puzzles are
constructed like multi-dynamic programs, beginner programmers can reduce abstractions by experiencing how a program would execute.

The escape room as a pedagogical tool is not completely novel. Guo & Goh (2014) conducted an observational study in which graduate students proposed a design for a digital escape room platform that would teach students how to navigate the library’s informational process. Similarly, Borrego, Fernandez, & Blanes (2007) built their own real-life escape room to engage second-year Computer Engineering students, who applied the knowledge acquired in class to escape the simulation. Their riddle flow- a dynamic sequence of riddles that contained clues to other riddles- drew similarities to the path-based puzzle structure outlined in Nicholson’s (2016) study. In Borrego’s et al.’s (2007) riddle flow model, each riddle includes three fields outlining the requirements, goal, and data received from the riddle. This organization is clear to players and scalable to any number of riddles or puzzles in an escape room model.

The location of these studies presents its advantages and disadvantages. Guo & Goh (2014) pursued their research on a virtual escape room at the Wee Kim Wee School of Communication and Information at Singapore’s Nanyang Technological University. Borrego et al. (2007) executed their study at the Department of Information and Communications Engineering (CE) at the Universitat Autònoma de Barcelona in Spain. The diversity of location suggests the escape room model can be applied to curriculum around the world, but that cultural distinctions may make the model easier to execute in some places than in others. Borrego et al. (2007) may argue with Guo that the escape room model is more effective as a physical simulation because, as a virtual simulation, students may mistake the model for a simple computer game and lose sight of the problem-solving aspect. Because Guo & Goh (2014) only propose a design for a virtual model, it cannot verify or deny this claim.
This study aims to explore how players’ problem-solving logic and object interactions in an escape room translate into introductory programming. In exploring the background literature in this field, some limits are apparent. Specifically, Cooper’s research about escape room facilities lack empirical data. Moreover, while Borrego’s experimental study was profound in integrating the escape room model with computer concepts, the study’s sample of second-year Computer Engineering students is significantly more advanced than this study’s target population. Nonetheless, these sources indicate some evidence that the escape room model could bridge abstract and concrete cognitive representations of programming topics. The study will further bridge this gap by observing beginner programmers execute procedure in the escape room model.

**Methodology**

**Research Questions**

The study seeks to answer two research questions.

1. Do teams who solve the room faster demonstrate stronger canonical procedure?
2. Does a strong canonical procedure aid participants’ object logic?

Regarding question (1), the study hypothesizes that teams who solve the escape room faster, or who progress further in the escape room, will demonstrate stronger canonical procedures.

Figure 4 models a natural sequence of steps needed to solve a common problem in an escape room: cracking a lock. Locks in escape rooms prevent participants from accessing a clue or artifact needed to solve the next puzzle. Figure 4 supposes a lock is secured around such item. This lock requires a 5-digit input. Participants must cycle through a sequence of operations and decisions called routines to open the lock. The routines in Figure 4 can be deconstructed by their functions in the algorithm.
**Terminator:** Modeled by (1), (12). Terminators signify the start and end of an algorithm. At (1), the problem is presented: the participants must open a 5-digit lock in order to unlock the contents of a secured box/drawer/door/etc. At (12), the problem is solved: the lock is opened, and the participants gain access to the secured content.

**Input:** Modeled by (3). Input signifies the data needed for an operation. At (3), participants require a 5-digit code to open the lock. Participants may have to search the room and/or decipher a clue to locate this 5-digit code.

**Output:** Modeled by (11). A participant enthusiastically announces the result of the algorithm to their team. The team’s efforts have successfully opened the lock.

**Operation:** Modeled by (2), (4), (5), (8), (10). Operations modify and/or use inputs to accomplish a task. For example, (5) takes the input from (3) and loads it into the lock. Operations also modify and/or use variables introduced in earlier operations. For instance, the variable $n$ is a counter to track the
number of participants who have attempted to open the lock. At (2), the variable $n$ is initialized to 0, signifying that no participant has attempted to open the lock. (8) increments $n$ by 1 each time a new participant tries to open the lock.

**Decision:** Modeled by (6), (7), (9). Decisions signify conditional statements. Depending on whether a condition is true or false, the algorithm proceeds in a particular direction. If the code opens the lock in (6) then participants proceed to the (11) after which they proceed to the next puzzle. If the code does not open the lock by (6) then participants continue to modify their approach.

The study equates having a strong canonical procedural approach with thinking in terms of a flow-chart model because both are naturally algorithmic. Participants who are comfortable executing algorithms know how to proceed from (1) to (11) in a logical manner. On the other hand, participants not comfortable executing algorithms may ignore decision block (7), skip the routines after (7), and resort back to (3). This can happen if participants do not factor their method of input into why the lock will not open. If the participant opening the lock does not reset it or put the digits in the correct order, then participants may incorrectly assume the code is wrong and consider looking for another. For this reason, the flow chart model suggests at least three participants should try the same code to rule out individual error as the reason why the lock does not open. This is represented by the condition in (9). However, teams can change the maximum $n$ to any number they perceive is appropriate; if they believe 5 people must try the lock before searching for a new code, they can adjust their condition to “Is $n=5$?” and proceed according to diagram.

Regarding question 2, the study hypothesizes that participants with a strong canonical procedural approach will have better object logic. Within the experimental setting, object logic extends the Sfard’s (1991) definition of “object conception” to include participants’ understanding of the artifacts in the room, specifically their properties and behaviors. Researchers are particularly interested if participants can identify the unique behaviors of an artifact that function towards a solution in an escape room.
The study outlines specific criteria for how to identify states and behaviors of artifacts in an escape room. Pan et al. (2017) provides a unique example:

Players in Team 3 were standing around a dimly lit lamp wondering what it could be used for. P9 noticed that there was a power cable running from the lamp to the other side of the wall in a separate room that they previously unlocked. P10 went into the other room to look around to see where the power cable went. After some searching, he found that it went to a hand crank. He started cranking it not knowing what effect it might have. This started to generate enough power to increase the brightness of the lamp in the first room but P10 couldn’t see it happening. P9 starting yelling to P10 in the other room to keep cranking. (p.1358)

In this example, the artifact is the lamp (see Figure 5, p. 37). The lamp has distinctive states, or features, that characterize its existence. These states include the lamp’s degree of brightness, whether the lamp is on or off, and its dimensions. The lamp also has behaviors that can modify these states. For example, the lamp can increase or decrease its brightness by turning its dial. However, artifacts may have different states and behaviors in an escape room than in real life.

Participants should be able to distinguish between these different states and behaviors. For example, the hand crank in the escape room specifically behaves to increase and decrease the brightness of the lamp. This behavior is unique to the hand crank in this instance and progresses the game. This behavior is more specific than how the hand crank behaves in real life, which is simply to turn and generate power for a device. This unique behavior becomes apparent when reviewing the procedure players in Team 3 took to reach this breakthrough: discovering a dimly lit lamp, following the power cable from the lamp to the hand crank, turning the hand crank, discovering it generates light, and communicating this discovery to players in the other room. Thus, executing procedure becomes essential in realizing artifacts’ states and behaviors.

Subjects

The study will observe six teams with five participants on each team. Participants will be gathered from three introductory programming courses at Rutgers University: Computing for Math
and the Sciences (01:198:107), Computers and Their Applications (01:198:110), Introduction to Computer Science or CSI (01:198:111). Participation is voluntary. Researchers will visit these courses at the end of instructional time to gather participants’ name and contact. Participants will select a primary and secondary date and time, then receive an email confirming their primary availability slot. If a participant cannot attend their primary availability slot, researchers will contact participants who listed that slot as their secondary availability slot. Each participant on each team will receive a $10 gift card at the completion of the experiment. The team who solves the room the fastest will receive an additional $5 gift card for each participant. Researchers will refer to participants in the study by pseudonyms- such as P1, P2, and P3- to ensure confidentiality.

**Setting**

Researchers will partner with the Rutgers Leadership and Experiential Learning, a Rutgers Student Affairs division who annually hosts escape room activities for students, to construct the experimental setting. Figure 6 provides a prototype model of the experimental setting (see p. 37). The experimental setting would be themed as a museum. The narrative is as followed:

*A valuable diamond has been stolen and hidden inside the Museum of Natural Art. Your team has been chosen to search the museum for the stolen diamond and return it to its rightful owner. But be careful: you have one hour to do so before the museum is alerted of your presence and the authorities arrive. Use the clues and hints in the room to find the diamond and escape in time.*

The narrative is essential to both the puzzles and the construction of the room. First, because the premise is centered on obtaining an object, it does not require locking the two doors to enter and exit the room. Participants are not physically secured inside the room and can leave at any time, if they choose. This reduces safety concerns: in an emergency evacuation, participants and researchers can easily exit the room and facility. If the team leaves the room without finding the diamond, given it is not an emergency, they are “caught by authorities” and lose the game. The
narrative only gives the illusion that participants must remain in the room until they locate the diamond, which signals the game has ended. The room will feature a path-based puzzle sequence.

The research team and Rutgers Leadership and Experiential Learning will borrow or make any artifacts needed to construct the experimental setting as a museum. The Rutgers Makerspace, located on Livingston Campus, is a great resource to construct realistic artifacts. The facility is free for Rutgers students to use but materials are chargeable. Artifacts created in the Makerspace would be budgeted and include paintings on walls and sculpted figures.

**Time**

Researchers will conduct this experiment on six different days between September and October 2020. Each team is scheduled for one of the six days. Researchers should prep the escape room and materials an hour before a team arrives. Participants should arrive at the site fifteen minutes prior to their designated time to receive instructions and sign waivers. Each team has one hour to complete the escape room. After they have completed the escape room, participants will have thirty minutes to complete a survey about their experience.

**Limitation**

Researchers foresee that teams who select an earlier date may communicate important elements of the experimental setting to teams who select a later date. This is possible because all teams are selected from the same three classes. Researchers will combat the sharing of answers from one team to another by offering an additional $5 gift card to each participant on the team that solves the room in the fastest time. This will incentivize teams to not share details about the experimental setting. It also ensures that later teams do not have an advantage on earlier teams and that the collected data is not skewed.
Completing the Escape Room and Post-Survey

Participants will enter the experimental setting on their selected day and time. The game host will debrief participants on the rules and emergency procedures of the escape room. The game host will collect all phones for security concerns, announce the escape room’s theme, read the narrative, and state the goal: locate the missing diamond. The game host will then answer questions. The observing researcher will enter the escape room with the participants and record their rankings in the observer’s template. The observer cannot touch artifacts, provide hints or clues, or propose solutions.

Each team has one hour to complete the escape room. If a team does not solve it in one hour, the game host will end the game, enter the room, and walk participants through the rest of the puzzles. The game host will answer any questions about puzzles or artifacts participants wish. The observing researcher will then record how long it took the team to solve the room. After teams finish the room, they will leave the experimental setting and receive a post-survey with four parts. Participants complete the post-survey independently, but can ask the observing researcher to clarify any questions. Finally, the game host and observing researcher will collect and file all surveys and reset the experimental setting for the next team.

Observer’s Role

The observing researcher (observer) has two objectives:

1. Watch the team’s interactions and performance inside the escape room and score the team in ten proficiencies. They will also record notes next to each proficiency.

2. Record the team’s completion status and time.

The observer will record scores, notes, and record the team’s completion status in the observer’s template (see Appendix, p. 38). Proficiencies are components necessary to solve an
escape room, including team cognition, situational awareness, problem-solving. These proficiencies are presented as questions such as “On a scale of 1-4, how well did this team work together?” 1 signifies low proficiency in this area and 4 signifies high proficiency in this area. Scoring helps observers quantify the team’s performance in each proficiency.

The observer will enter the escape room with the participants and complete the observer’s template as the game unfolds. The observer cannot touch artifacts, provide hints or clues, or propose solutions. At the end of the game, the observer will record the team’s completion status and time. The “Completion Status?” row indicates if the team completed the room. The “Time of Completion” row indicates how long it took for participants to solve the room. Researchers will compare the observer’s notes and scores to participants’ scores in the survey.

Analyzing Survey Results

All participants receive a post-survey with four parts (see Appendix, p. 40).

Part One: This section asks the participants to score their team’s performance in the same eleven proficiencies as the observer’s template. The questions are modified, however, to fit the participants’ perspective. For example, they would receive a question like, “On a scale of 1-4, how well did your team work together?”. Ultimately, this section will gather how participants on a single team perceive their performance in each proficiency versus how the observer perceived their team performance. Researchers will record the most frequent participant ranking and observer ranking in each proficiency in an Excel table. If two rankings tie for most frequent participant ranking, researchers will record both and the observer will offer why this difference may exist.

Part Two: This section will quantify the strength of individual participants’ canonical procedure by having participants construct a flow chart procedure. The survey describes the puzzle in Figure 4 (see p.19) and provides the routines participants should use to construct their flow chart
model. First, researchers will consider the following: Is the procedure legible? Is the procedure logical? The first question ensures the procedure is syntactically correct: terminators start and end procedures, every routine has the appropriate number of arrows, etc. The second question ensures that the procedure is semantically correct, and that its routines are presented in a logical order.

Next, researchers will quantitatively evaluate each participants’ answers by the following criteria: How many routines are correctly labeled? How many arrows correctly link one routine to another? How many arrows are correctly labeled? These three criteria will be measured based on Figure 4. Each correctly labeled and positioned routine and arrow is 1 point. There are 12 routines and 14 arrows in the flow chart model; 6 of these arrows are labeled “yes” or “no”. In total, participants can earn a score from 0-32.

The team’s procedural score is an average of each participant’s procedural score on Part Two. Each team’s procedural score is plotted on a scale from 0-32. A team score between 0-18 signifies a weak procedure that would fail inside the escape room. A team score between 18-28 represents an adequate, but an underdeveloped procedure. If participants executed this procedure inside the escape room, it is possible, but not guaranteed, they would arrive at the desired solution. Finally, a team score between 28-32 signifies a well-formed, executable procedure. If participants were to execute this procedure in real life, they would most likely arrive at the desired solution.

**Part Three:** Researchers will compare answers to the object chart in Part Three between teams with higher and lower procedural scores. First, researchers will parse responses for key phrases that match the answers they are looking for. For example, a participant who describes a behavior of the hand crank in the escape room as “control lamp’s brightness” nears the researchers’ desired answer, which is “increases brightness of lamp”. A participant whose answer to this entry does not reference changing the degree of lamp brightness has not given a sufficient answer. The
participants’ individual procedural score will then be considered to see if a high score correlates with providing a sufficient response.

**Part Four:** Researchers will account for participants’ major, current course enrollment, prior programming experience, and whether this is their first escape room experience. They will consider: Do participants in computer-science related majors have an advantage over non-computer science related majors? Do participants who have played an escape room before have an advantage over those who have not? Researchers will record these results in an Excel bar chart.

**Anticipated Results**

*Part One*

**Figure 7. Team One’s Performance Ranking**

<table>
<thead>
<tr>
<th>Questions</th>
<th>P1 Ranking</th>
<th>P2 Ranking</th>
<th>P3 Ranking</th>
<th>P4 Ranking</th>
<th>P5 Ranking</th>
<th>Participants’ Most Frequent Ranking</th>
<th>Observer Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>On a scale of 1-4, how difficult was the escape room?</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>On a scale of 1-4, how well did your team work together?</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>On a scale of 1-4, how well did your team communicate?</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
Figure 8. Team Six’s Performance Ranking

<table>
<thead>
<tr>
<th>Questions</th>
<th>P25 Ranking</th>
<th>P26 Ranking</th>
<th>P27 Ranking</th>
<th>P28 Ranking</th>
<th>P29 Ranking</th>
<th>Participants’ Most Frequent Ranking</th>
<th>Observer Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>On a scale of 1-4, how difficult was the escape room?</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>On a scale of 1-4, how well did your team work together?</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>On a scale of 1-4, how well did your team communicate?</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Part Two

Figure 9. Teams’ Procedural Scores on a Scale

Figure 10. Teams’ Time of Completion and Procedural Scores

<table>
<thead>
<tr>
<th>TEAMS</th>
<th>COMPLETION STATUS?</th>
<th>TIME OF COMPLETION</th>
<th>TEAM PROCEDURAL SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
<td>Yes</td>
<td>0:10:00</td>
<td>31</td>
</tr>
<tr>
<td>Team 2</td>
<td>Yes</td>
<td>0:20:00</td>
<td>29</td>
</tr>
<tr>
<td>Team 3</td>
<td>Yes</td>
<td>0:30:00</td>
<td>24</td>
</tr>
<tr>
<td>Team 4</td>
<td>Yes</td>
<td>0:40:00</td>
<td>22</td>
</tr>
<tr>
<td>Team 5</td>
<td>Yes</td>
<td>0:50:00</td>
<td>11</td>
</tr>
<tr>
<td>Team 6</td>
<td>No</td>
<td>1:00:00</td>
<td>5</td>
</tr>
</tbody>
</table>
Part Three

Figure 11. Measuring Teams’ Time of Completion versus Team Procedural Score

![Graph showing the relationship between time and score.]

Figure 12. Object Chart

<table>
<thead>
<tr>
<th>Artifact</th>
<th>What properties does this artifact have in real life?</th>
<th>What properties does this artifact have specifically in the escape room?</th>
<th>What behaviors does this artifact have in real life?</th>
<th>What behaviors does this artifact have specifically in the escape room?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lamp</td>
<td>• brightness</td>
<td>• dimmer, lower range of brightness</td>
<td>• increase brightness</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>• dimensions</td>
<td></td>
<td>• decrease brightness</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• weight</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• color</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• on/off</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• weight</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hand crank</td>
<td>• dimensions</td>
<td>N/A</td>
<td>• generate power</td>
<td>• increase brightness of lamp</td>
</tr>
<tr>
<td></td>
<td>• weight</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• gears</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• handle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power cable</td>
<td>• length</td>
<td>N/A</td>
<td>• connect two devices</td>
<td>• connects hand crank and lamp</td>
</tr>
<tr>
<td></td>
<td>• weight</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• plug</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• socket</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Part Four

Figure 13. Measuring the Number of Respondents who have Played an Escape Room

Figure 14. Measuring the Number of Respondents in Computer Science-Related Majors

Discussion

Part One: The study predicts that teams who solve the room faster will rank the difficulty of the room lower than the teams who take longer to solve the room or who do not solve it at all. It is also likely that teams who solve the room faster will rank areas such as team cognition and communication higher than the teams who take longer to solve the room or who do not solve it at all. This is demonstrated by hypothetical rankings in Figure 7 and Figure 8.
In Figure 7, Team One, the team to solve the room the fastest, ranked the difficulty of the room as low (1), but their team cognition and team communication as high (4). Team Six, however, ranked the difficulty of the room as high (4), and their team cognition and team communication as low (1). By comparing the observer’s ranking to the participants’ most frequent ranking, researchers can discuss how the participants’ perception of their team’s performance differs from the observer’s perception of their team’s performance. If the observer and participants’ most frequent ranking do not match, the observer may discuss why they ranked the team as they did and what factors may account for the difference. The study predicts that Team One will rank the difficulty of the room lower than the observer and rank their team cognition higher than the observer. Team One may have an inflated view of their performance because they finished quickly. Team Six may under-rank their performance because they did not solve the room. This is evident in row three of Figure 8 where the observer ranks Team Six’s team cognition higher than the participants themselves.

**Part Two:** The study predicts that as a team’s time of completion increases, the team procedural score decreases. Figure 10 assigns hypothetical values to six teams where the time of completion increases with each listed team. The study predicts that Team One, which finishes the room in ten minutes will have a higher team procedural score (after individual participant scores are averaged from Part One). This contrasts with Team Six who does not solve the room, remains in the room for the full hour, and has a lower team procedural score. Figure 11 depicts the negative correlation between the two variables on a line graph.

Given the values in Figure 10, researchers would calculate that the correlation between the time of completion and team procedural score would be -0.9681024339. This value demonstrates that there is a strong relationship between time of completion and team procedural score which is
negative nearly 97% of the time. The study does not predict this exact correlation will appear in the data but reasons participants who demonstrate better procedural logic on the post-survey are more likely to have solved the escape room in a faster time.

**Part Three:** The study anticipates that teams who solve the room faster will more likely provide the sufficient responses, shown in Figure 12, than teams who take longer to solve the room. For example, participants from Team One are more likely to identify the behavior of the hand crank in the escape room (column 5, row 3) as being able to “increase the brightness of the lamp” than participants from Team Six. Team Six’s answers for this entry may be “turn” or “generate power.” Team Six’s answers are insufficient for this entry. While Team Six’s answers are true of the hand crank in real life, they are not particular to its behavior in the escape room.

**Part Four:** The study anticipates that participants who have completed an escape room prior will have an advantage over participants who don’t. In Figure 13, every Team Six participant indicates this experiment as their first escape room. The study also anticipates that faster teams have more participants majoring in a computer science-related field. This is demonstrated by Figure 14 in which Team One has all computer science-related major participants and Team Six has none. Computer science-related majors are more likely to have developed some problem-solving skills in their classwork prior to entering the experimental setting. This relationship would reaffirm that ‘programming thinking’ is not a magical skill, but is rather reinforced and developed over time. As participants continually think about the best procedure to solve a problem, they will be more able to apply algorithms to solve programming problems.

**Conclusion**

The escape room is an unconventional, yet applicable model to teach beginner programmers procedural skills. First, the puzzles allow participants to understand programming as
a series of routines including inputs, process, outputs, and decisions. Second, by solving puzzles through these routines, participants learn more about the states and behaviors of artifacts in the escape room, which are crucial to understanding object-oriented programming. Finally, teachers understand which components students struggle with as puzzles isolate these gaps in knowledge.

The escape room as an experiential learning model has wide implications. Researchers can extend this study to have participants code artifacts’ states and behaviors in Java or another object-oriented language. This would further bridge their escape room experience with actual coding. The properties of the hand crank and lamp can be stored as data types. For example, the state of the lamp being on or off is representable by a Boolean variable, where the value of the variable is either true or false. Additionally, the interactions between the hand crank and lamp are a prime example of how objects interact in code through method calls.

The escape room can help instructors focus curriculum on fixing common procedural errors. For example, in the lock problem, if instructors know students struggle with tracking counter variables that help loop them through routines, they can address this through additional class time devoted to implementing counter variables. Additionally, students can code their own escape rooms with puzzles and a narrative. This extends to Guo and Goh’s (2014) study because it moves students from blueprinting to implementing the design. Ultimately, the escape room model can help instructors move beginner programmers closer to ‘programming thinking’, as Eckerdal et al. (2005) studied, by reinforcing a procedural framework.
References


Appendix

Figure 1. Escape Room Prototype

Exit

Entrance

Artifacts

P1

P2

P3

P4

Computer monitor (hints)

Game host

Figure 5. Objects’ States and Behaviors

Figure 6. Escape Room-Experimental Setting
## Observer’s Template

<table>
<thead>
<tr>
<th>QUESTIONS</th>
<th>RANKING</th>
<th>Completion Status?</th>
</tr>
</thead>
<tbody>
<tr>
<td>On a scale of 1-4, how difficult did this team perceive the escape room to be?</td>
<td>1 - The room, including its puzzles and narrative, was extremely easy for the team to solve.</td>
<td>2 - The room, including its puzzles and narrative, was straight-forward.</td>
</tr>
<tr>
<td></td>
<td>3 - The room, including its puzzles and narrative, was somewhat difficult to solve.</td>
<td>4 - The room, including its puzzles and narrative, was very difficult for the team to solve.</td>
</tr>
<tr>
<td>On a scale of 1-4, how well did this team work together?</td>
<td>1 - The team failed to work as a cohesive unit.</td>
<td>2 - The team struggled to work as a cohesive unit throughout the game.</td>
</tr>
<tr>
<td></td>
<td>3 - The team managed to work as a cohesive unit though it took time to accomplish.</td>
<td>4 - The team excelled at working as a cohesive unit from start to finish.</td>
</tr>
<tr>
<td>On a scale of 1-4, how well did this team communicate?</td>
<td>1 - The team failed to communicate solutions, share information, or listen to others.</td>
<td>2 - The team struggled to communicate solutions, share information, or listen to others.</td>
</tr>
<tr>
<td></td>
<td>3 - The team managed to communicate solutions, share information, and listen to others, though it took time to accomplish.</td>
<td>4 - The team effectively communicated solutions, shared information, and listened to others’ despite challenges throughout the game.</td>
</tr>
<tr>
<td>On a scale of 1-4, how well did this team make use of the artifacts in the room as it related to the purpose of escape?</td>
<td>1 - The team failed to use the artifacts in a way that aided their escape. Their interactions with artifacts did not go beyond conception of their states and behaviors in real life.</td>
<td>2 - The team struggled to use artifacts in a way that aided their escape. Their interactions with artifacts were limited to their conception of their states and behaviors in real life.</td>
</tr>
<tr>
<td></td>
<td>3 - The team managed to use artifacts in a way that aided their escape. Their interactions with artifacts were slightly limited to their conception of their states and behaviors in real life.</td>
<td>4 - The team effectively used artifacts in a way that aided their escape. Their interactions with artifacts went beyond their conception of their states and behaviors in real life.</td>
</tr>
<tr>
<td>On a scale of 1-4, how effectively did this team identify inputs needed to solve puzzles?</td>
<td>1 - The team consistently struggled to identify the correct inputs required to solve puzzles based on information from clues, hints, and what is naturally suggested by the given puzzle.</td>
<td>2 - The team struggled on multiple occasions to identify the correct inputs required to solve puzzles based on information from clues, hints, and what is naturally suggested by the given puzzle.</td>
</tr>
<tr>
<td></td>
<td>3 - The team managed to identify the correct inputs required to solve puzzles based on information from clues, hints, and what is naturally suggested by the given puzzle.</td>
<td>4 - The team consistently succeeded in identifying the correct inputs required to solve puzzles based on information from clues, hints, and what is naturally suggested by the given puzzle.</td>
</tr>
<tr>
<td>On a scale of 1-4, how effectively did this team execute procedures to solve a puzzle?</td>
<td>1 - The team neglected to execute procedures together and individuals followed their own plan of action.</td>
<td>2 - The team struggled to execute procedures together. Individuals often could not decide to follow a singular plan of action.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>On a scale of 1-4, how often did this team modify their solutions before considering other solutions?</td>
<td>1 - The team rarely modified their solutions, or considered why their solutions failed, before moving onto other solutions.</td>
<td>2 - The team sometimes modified their solutions, or considered why their solutions failed, before moving onto other solutions.</td>
</tr>
<tr>
<td>On a scale of 1-4, how effectively did this team identify outputs produced from the puzzles?</td>
<td>1 - The team consistently struggled to identify the correct outputs to puzzles. In other words, the goal of each puzzle was very unclear or unrecognizable to the team when obtained.</td>
<td>2 - The team struggled on multiple occasions to identify the correct outputs to puzzles. In other words, the goal of each puzzle was somewhat unclear or unrecognizable to the team when obtained.</td>
</tr>
<tr>
<td>On a scale of 1-4, how dynamic was this team’s mental model of the room?</td>
<td>1 - The team did not have a working knowledge of the room and its puzzles based on prior experience.</td>
<td>2 - The team barely had a working knowledge of the room and its puzzles based on prior experience.</td>
</tr>
<tr>
<td>On a scale of 1-4, how effectively did this team problem-solve inside the escape room?</td>
<td>1 - The team failed to think creatively and logically through puzzles.</td>
<td>2 - The team struggled to think creatively and logically through puzzles.</td>
</tr>
</tbody>
</table>
**Escape Room Post-Survey**

Congratulations, you have completed the escape room challenge! We’d like to you to reflect honestly about your experience below.

**Part One**

Below are a series of questions asking you to rank your team’s performance in various aspects. Circle the answer that best describes this performance.

1. On a scale of 1-4, how difficult was the escape room?
   1-The room, including its puzzles and narrative, was extremely easy to solve.
   2-The room, including its puzzles and narrative, was straight-forward.
   3-The room, including its puzzles and narrative, was somewhat difficult to solve.
   4-The room, including its puzzles and narrative, was very difficult to solve.

2. On a scale of 1-4, how well did your team work together?
   1-My team failed to work as a cohesive unit throughout the game.
   2-My team struggled to work as a cohesive unit throughout the game.
   3-My team managed to work as a cohesive unit, though it took time.
   4-My team excelled at working as a cohesive unit despite challenges in the game.

3. On a scale of 1-4, how well did your team communicate?
   1-My team failed to communicate solutions, share information, or listen to others.
   2-My team struggled to communicate solutions, share information, or listen to others.
   3-My team managed to communicate solutions, share information, and listen to others, though it took time to accomplish.
   4-My team effectively communicated solutions, shared information, and listened to others from start to finish.

4. On a scale of 1-4, how well did your team make use of the artifacts in the room as it related to the purpose of escape?
   1-My team failed to use the artifacts in a way that aided our escape.
   2-My team struggled to use artifacts in a way that aided our escape.
   3-My team managed to use artifacts in a way that aided our escape.
   4-My team effectively used artifacts in a way that aided our escape.

5. On a scale of 1-4, how effectively did your team identify inputs needed to solve puzzles?
   1-My team consistently struggled to identify the correct inputs required to solve puzzles based on information from clues, hints, and what is naturally suggested by the given puzzle.
   2-My team struggled on multiple occasions to identify the correct inputs required to
solve puzzles based on information from clues, hints, and what is naturally suggested by the given puzzle.

3-My team managed to identify the correct inputs required to solve puzzles based on information from clues, hints, and what is naturally suggested by the given puzzle.

4-My team consistently succeeded in identifying the correct inputs required to solve puzzles based on information from clues, hints, and what is naturally suggested by the given puzzle.

6. On a scale of 1-4, how well did your team execute procedures to solve a puzzle?

1-My team neglected to execute procedures together and individuals followed their own plan of action.
2-My team struggled to execute procedures together. Individuals often could not decide to follow a singular plan of action.
3-My team executed procedures together and individuals followed a singular plan of action often.
4-My team always executed procedures together and followed a joint plan of action.

7. On a scale of 1-4, how often did your team modify solutions before considering other solutions?

1-My team rarely modified its solutions, or considered why solutions failed, before moving onto other solutions.
2-My team sometimes modified its solutions, or considered why solutions failed, before moving onto other solutions.
3-My team often modified its solutions, or considered why solutions failed, before moving onto other solutions.
4-My team frequently modified its solutions, or considered why solutions failed, before moving onto other solutions.

8. On a scale of 1-4, how effectively did your team identify outputs produced from the puzzles?

1-My team consistently struggled to identify the correct outputs to puzzles.
2-My team struggled on multiple occasions to identify the correct outputs to puzzles.
3-My team managed to identify the correct outputs to puzzles.
4-My team consistently succeeded in identifying the correct outputs to puzzles.

9. On a scale of 1-4, how dynamic was your team's mental model of the room?

1-My team did not have a working knowledge of the room and its puzzles based on prior experience.
2-My team barely had a working knowledge of the room and its puzzles based on prior experience.
3-My team had some working knowledge of the room and its puzzles based on prior experience.
4-My team had a strong working knowledge of the room and its puzzles based on prior experience.

10. On a scale of 1-4, how effectively did your team problem-solve inside the escape room?
   1- My team failed to think creatively and logically through puzzles.
   2- My team struggled to think creatively and logically through puzzles.
   3- My team managed to think creatively and logically through puzzles.
   4- My team effectively thought creatively and logically through puzzles.
Part Two

Your team has located a mysterious box. Unbeknown to you it contains the missing diamond! Your team open the lock around the box to gain access to the locked artifact. The lock requires a five-digit input to open. We suggest at least three people try the lock with the same input before trying another five-digit input. Make sure the lock is reset before inserting inputs and that inputs are correctly inserted.

This exercise will require you to construct your own flow chart algorithm based on the description, flow chart items, and routines below. Refer to the researcher for questions about symbols.

A. Label the flow-chart shapes with the appropriate routines listed in the right column. B. Construct a flow chart algorithm based on the routines labeled in Part A. Include arrows to show directionality. Label arrows leaving conditionals “yes” and “no”.

Part A

**SHAPES**

**ROUTINES**

- Reset lock
- Give lock to another participant to trying another 5-digit code
- n++
- End
- Does code open lock?
- Insert code correctly into lock
- Say “Lock opened”

**NOTE:**

n=number of people who have tried to open the lock
Part B
Use this sheet to construct your flow chart model. Use the only shapes with their labeled routines above.
Part Three

7. Complete the chart below.

<table>
<thead>
<tr>
<th>Artifact</th>
<th>What properties does this artifact have in real life?</th>
<th>What properties does this artifact have in the escape room?</th>
<th>What behaviors does this artifact have in real life?</th>
<th>What behaviors does this artifact have in the escape room?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artifact #1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Artifact #2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Artifact #3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Artifact #4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Part Four

1. Was this your first escape room?
   - Yes
   - No

2. If you checked “No” for the previous answer, how many escape rooms have you done prior to this escape room, regardless if you solved the room?

3. Please list your major(s) or intended major(s).

4. Please check the box that corresponds to class year.
   - 1st year
   - 2nd year
   - 3rd year
   - 4th year
   - 5th year and above

5. Please check all boxes that apply to you. Are you currently enrolled in any of the following courses?
   - Computing for Math and the Sciences (01:198:107)
   - Computers and Their Applications (01:198:110)
   - Introduction to Computer Science (01:198:111)

6. Please check all boxes that apply to you. Were you ever enrolled AND passed any of the following courses?
   - Computing for Math and the Sciences (01:198:107)
   - Computers and Their Applications (01:198:110)
   - Introduction to Computer Science (01:198:111)

7. Did you take AP CS A in high school?
   - Yes
   - No

8. Please list any other programming experience, if applicable.