Unit 7 Lesson 1

Create PT - Review the Task

Resources

Create PT - 2b

2b. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and / or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development. *(Must not exceed 200 words)*

Scoring Guidelines	
Row and Task	**Decision Rules**
Row 2 - Response 2B Describes or outlines steps used in the incremental and iterative development process to create the entire program.	**Do NOT award a point if any one of the following is true:** ● the response only includes the process for determining the program idea and does not address the development process used to create the entire program; or ● the response does not indicate iterative development; ● refinement and revision are not connected to feedback, testing, or reflection; or ● the response only describes the development at two specific points in time.
Row 3 - Response 2B Specifically identifies at least two program development difficulties or opportunities. **AND** Describes how the two identified difficulties or opportunities are resolved or incorporated.	Response earns the point if it identifies two opportunities, or two difficulties, or one opportunity and one difficulty AND describes how each is resolved or incorporated. **Do NOT award a point if any one of the following is true:** ● only one distinct difficulty or opportunity in the process is identified and described; or ● the response does not describe how the difficulties or opportunities were resolved or incorporated.

Student Response A - [Video] [Written Response]	**Scoring Guidelines**	
I independently developed this program starting with my original method and continuing to build off of that. During the process, as I would think of an element that I needed to add to the game, I would work on the code needed for that certain part of the game. An element that gave me some trouble was the first part my wolfMove method which was meant to have the wolf move to blue chickens and eat them. I had a few problems with this method because at first, my wolf would move, but end up halfway underground. After a lot of trial and error, I was able to make the wolf move up while moving forward and make sure that when it stopped moving it was above ground. I	**Row 2**	**1**
	The response earned a point for this row. The response describes the iterative process in developing the program: "starting with my original method and continuing to build off of that. During the process, as I would think of an element that I needed to add to the game, I would work on the code needed for that certain part of the game."	
	Row 3	**1**
	The response earned a point for this row. The response identifies a **difficulty**: "the first part my wolfMove	

was also presented with opportunities to add aspects to make my game better. One of these moments was when I decided that my game was too easy to play. To make the game more fun, I looked up a tutorial on how to make a countdown timer. Once I added the timer to the game, there was more pressure, which made the game realistic.	method." The response indicates that it is resolved "after a lot of trial and error ... able to make the wolf move up while moving forward and make sure that when it stopped moving it was above ground." The response identifies an **opportunity** "to add aspects to make my game better ... the game was too easy to play." The opportunity is addressed when a timer was added to the game: "there was more pressure, which made the game realistic."
Student Response B - [Video] [Written Response]	**Scoring Guidelines**
An incremental process was when I drew a flowchart detailing the flow of the program, including how the program has three functions collect_dict(), study(), quiz(), that collects term-definition-pairs, helps user study, and quizzes the user, respectively, and a fourth function run_quiz() that calls the three former functions in the correct order. Another incremental process incorporated into the code is a for loop that prints blank lines until the counter reaches 100. An iterative process was when I decided to improve the program by using the lower() method to make it non-case-sensitive for users. Both these processes were done independently. A difficulty encountered was when my tester found that the program always calculates the user's quiz-scores 1-digit higher than it actually is. I resolved this by assigning points_scored to 0. Another difficulty was encountered when my tester noticed that the quiz feature was printed directly below the "study" feature and users taking the quiz can cheat off the answers displayed in the study section. I resolved this by printing 300 blank lines between the "study" and "quiz" portions. These 2 difficulties were resolved by collaborating with a classmate.	**Row 2** — **1** **The response earned a point for this row.** The response describes the incremental process used to create the program. The response lists these steps: "drew a flowchart, incorporated a for loop that prints blank lines, decided to improve the program using the lower() method." The iterative process occurred during the difficulty where testing was used to find a problem where the "program always calculates the user's quiz-score 1-digit higher than it actually is" and the response includes how it was resolved. **Row 3** — **1** **The response earned a point for this row.** The response describes a **difficulty** ("calculates the user's quiz-scores 1-digit higher than it actually is") and its resolution ("assigning points_scored to 0"). The response includes a second **difficulty** ("users taking the quiz can cheat off the answers displayed in the study section") and its resolution ("printing 300 blank lines between the 'study' and 'quiz' portions").
Student Response C - [Video] [Written Response]	**Scoring Guidelines**
I completed the program independently. I started out by setting up the background and arranging the cards before developing the Card class. After looking at the pixel size of the cards, I determined the coordinates I needed to use. I tested the program multiple times to make sure I got the coordinates and placement of the cards right with each version of the positioning algorithm. I was able to visually confirm the accuracy of the program by running it to check if it worked. Then, I worked on getting them to flip. Lastly, I developed the algorithm that tests if the card flipped had an x behind it and made the Gameover class to end the game. At first, I wanted to make the program work so that whenever I clicked the card, the card would flip. However, I had trouble implementing Greenfoot's mouseClicked() method. In the act() method of the Card class, I called the turnCard() method I wrote, which would flip the card. Although I used the mouseClicked() method properly, it did not flip the card; the image of the card did not change as intended. To solve this issue, I replaced the	**Row 2** — **1** **The response earned a point for this row.** The response describes the incremental process to develop the program: "setting up the background and arranging the cards before developing the Card class ... determined the coordinates ... tested the program multiple times ... confirm the accuracy of the program by running it to check if it worked ... getting them [cards] to flip." **Row 3** — **1** **The response earned a point for this row.** The response identifies a **difficulty** ("trouble implementing Greenfoot's mouseClicked() method") with a resolution ("replaced [it] with a call to Greenfoot's isKeyDown() method in the same location"). The response identifies a second **difficulty** ("needed an easier way to restart the game") and its resolution ("developed a way to press r in

mouseClicked() call with a call to Greenfoot's isKeyDown() method in the same location to flip the card based on pressing its order value (1-9). Each card is instantiated with its order value, so after assigning that parameter to a private instance variable, I could use that variable as a parameter in the isKeyDown() call. I also needed an easier way to restart the game, so I developed a way to press r in order to reset the background when the Gameover screen pops up. With an if statement checking if the user presses "r" in the act() method of the Gameover class, pressing "r" would reset the cards.	order to reset the background when the Gameover screen pops up").
Student Response D - [Artifact] [Written Response]	**Scoring Guidelines**
The bulk of my project was created independently, but I had a minimal amount of help from a partner. I started off by planning out how to make the bowling pins get knocked out as the bowling ball went forward to hit them. While doing this, I encountered a huge problem in my code. Every time the bowling ball touched the bowling pins, they didn't disappear although the code specifically stated that they should disappear once they were in contact with one another. My friend helped me with this by suggesting that I should make a code for the bowling pins so that when the bowling pins broadcasted messages and the bowling ball received them, the program would run smoothly and the bowling pins would disappear every time the bowling ball received their individual broadcast message. Another issue I faced was when all the pins were down, the host wouldn't say "Great job!" I resolved this problem by broadcasting a message when the last pin fell down so that when the host received the message, he would say "Great job!"	**Row 2** **0** **The response DID NOT earn a point for this row.** The response does not describe the overall development process of the entire program. The response focuses almost entirely on the two difficulties. **Row 3** **1** **The response earned a point for this row.** The response earned the point for this row. The response describes two difficulties. The first **difficulty** is that "every time the bowling ball touched the bowling pins, they didn't disappear." This is resolved by making the code for the bowling pins broadcast messages for the bowling ball to receive. The second **difficulty** is that "when all the pins were down, the host wouldn't say 'Great job!'" This is resolved by "broadcasting a message when the last pin fell down."
Student Response E - [Video] [Written Response]	**Scoring Guidelines**
There were many problems that arose while coding the program. One of the early problems encountered was deciding how I should set up my study guide. For example, I could have chosen to do flashcards along with doing multiple choice. However, I felt that the flash cards would be more effective and efficient way of creating this app. Furthermore, when making this study guide app, I felt that there needed to be something else that could have made the study guide more useful for the reader. Originally, there were just flashcards, but I felt there was something else that could be done. So I included another button that allowed the user to type in the definition as the word was being given. This was a major development addition as it is more effective for the user to write the information than by just looking at cards. This is also more effective for memorization.	**Row 2** **0** **The response DID NOT earn a point for this row.** The response does not describe the incremental or iterative process used in developing the entire program. The response focuses on two decisions that were made in determining what would be in the program. **Row 3** **0** **The response DID NOT earn a point for this row.** The response identifies an **opportunity** as adding functionality to allow users to enter the "definition as the word was being given." This is resolved by including "another button." The **difficulty** identified is the decision to use flashcards over multiple choice, which is a **design choice, not a program development difficulty**.
Student Response F - [Video] [Written Response]	**Scoring Guidelines**

There were several problems that were presented in the code. One problem was setting the timer to countdown 30 seconds and stopping the time when the pause button was clicked, then resuming the time again. This was an independent development. I had to find out how to resume the game, such that the time that was paused did not restart again after clicking the resume button. To solve this, I made a function called Time and called it in the event handler of the resume button, which in result would change just the time in intervals of 1000 milliseconds, but not move back to 30 seconds. Another problem, also an independent development, was updating the high score the user received every time the score was higher. The code would update the high score, but to the score the user received, regardless if it was higher or not. I decided to add in an if statement, such that if the score the user received was greater than the score from other games, it would update the high score value to that score.	**Row 2**	**0**
	The response DID NOT earn a point for this row. The response does not describe the incremental and iterative design process for the entire program.	
	Row 3	**1**
	The response earned a point for this row. The response identifies two difficulties and how they are resolved. The first **difficulty** is "setting the timer to countdown 30 seconds and stopping the time when the pause button was clicked." Its resolution is "a function called Time and called it in the event handler of the resume button, which in result would change just the time in intervals of 1000 milliseconds, but not move back to 30 seconds." The second **difficulty** is "updating the high score the user received every time the score was higher." Its resolution is "to add in an if statement, such that if the score the user received was greater than the score from other games, it would update the high score value to that score."	
Student Response G - [Video] [Written Response]	**Scoring Guidelines**	
One difficulty I faced was figuiring out how to get the Snake to grow. To solve this problem I made the snake create a clone every time it got an apple. When there are new clones, it waits a while then deletes some of the clones so the snake isn't too long. This was done independently. Another problem I had was channging the speed of the snake. To solve this I had to add a certain amount of steps to the original speed of the snake every time it got an apple. This was done collaboratively.	**Row 2**	**0**
	The response DID NOT earn a point for this row. The response does not describe the overall development of the entire program.	
	Row 3	**1**
	The response earned a point for this row. The response identifies two difficulties during the program development. The first **difficulty** is "figuring out how to get the Snake to grow" and is resolved by making "the snake create a clone every time it got an apple ... it waits a while then deletes some of the clones so the snake isn't too long." The second **difficulty** is "changing the speed of the snake." This is resolved by "add a certain amount of steps to the original speed of the snake every time it got an apple."	
Student Response H - [Video] [Written Response]	**Scoring Guidelines**	
The main goal of the app I created is to have people learn new words in different languages. The quiz works as follows: If the correct answer is clicked, then the next question appears; if the wrong one is clicked, then the quiz restarts. The main structure of the app is a series of onEvent functions to lead you from one question to the next; this series of repeated onEvent functions exemplifies repetition, a key element in learning a new language. To make the app more interactive, I decided to add in user input on three of the twelve questions. I had trouble with	**Row 2**	**0**
	The response DID NOT earn a point for this row. The response does not describe the incremental and iterative development process for the program. Instead, the response describes how the program functions and states the main components of the program code.	
	Row 3	**0**

having the user input box cleared after the check button is clicked. I kept running the app, but came back to the same problem. It was after a couple times that I realized that I needed to add in a setText function to clear the input box every time the check button is clicked. This app was entirely an independent project, with no outside collaboration.	**The response DID NOT earn a point for this row.** The response does not identify two difficulties and/or opportunities and how they are resolved or handled. The response only identifies one **difficulty**, clearing the user input box.
Student Response I - [Video] [Written Response]	**Scoring Guidelines**
Whilst developing the code for the program, two main problems came about. One main problem was to learn how to get the app to display in the countdown slot in the screen the values chosen by the user preliminary set by me on the dropdown text. This was an independent development. At first, my idea was to implement many numbers from where to choose from, but that would had resulted in an exponentially longer code to map to the screen and only giving the option of one-digit values. Therefore, I decided to implement a dropdown textbox with predetermined values as to now only map the value of it to the screen, resulting in a successful simplification of my code. Second problem was the most essential to fix, how to subtract one progressively. This was also resolved independently. For this I implemented a function that would repeat itself every second, which would take countdown and subtract one from its value until it reached zero. But to keep it from going into negative numbers, I implemented a function which would reset the value of countdown and set it to zero. Which took less functions and is more cleaner than implementing a for-loop.	**Row 2** **0** **The response DID NOT earn a point for this row.** The response does not describe the overall interactive or iterative development of the entire program. **Row 3** **0** **The response DID NOT earn a point for this row.** The response states that "One main problem was to learn how to get the app to display in the countdown slot..." However, this is not a problem or opportunity during the program development process, **it's a design issue.** The second difficulty and its resolution are given.
Student Response J - [Video] [Written Response]	**Scoring Guidelines**
In our app " Sports Trivia", it included player's faces and team logos for the participant to guess from right to wrong. We have a title screen which shows which sport you want to be questioned on. The game consists of 2 questions for each sport that has to be answered. So basically whatever sport you want you can choose it from the welcome screen and it'll bring you to the desired sport you want to be questioned on. It was hard deciding which particular questions my partner and i would come up with because we would always have debates on which question is the hardest or easiest. Also, it was hard figuring what genre our game would be based on, there are plenty of different ideas for a trivia game. Developing the game was a hardship, figuring out our code and transitions for each screen was always a debate, there was a lot of testing and bugs along the way but we came to a final conclusion for all these tough situations.	**Row 2** **0** **The response DID NOT earn a point for this row.** The response does not describe the incremental and iterative development of the entire program. **Row 3** **0** **The response DID NOT earn a point for this row.** The response does not identify two difficulties and/or opportunities related to the program development process and how these programming issues were resolved or handled. The issues identified have to do with "what particular questions my partner and I would come up with" and "what genre our game would be based on."

Create PT - 2c

2c. Capture and paste a program code segment that implements an algorithm (marked with an **oval** in **section 3** below) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. *(Must not exceed 200 word)*

Scoring Guidelines	
Row and Task	**Decision Rules**
Row 4 - Code Segment in Response 2C Selected code segment implements an algorithm.	**Do NOT award a point if any one of the following is true:** ● the algorithm consists of a single instruction; or ● the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or ● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).
Row 5 - Response 2C Selected code segment implements an algorithm that uses mathematical or logical concepts **AND** Explains how the selected algorithm functions **AND** Describes what the selected algorithm does in relation to the overall purpose of the program.	● The algorithm being described can utilize existing language functionality or library calls. ● Response earns the point even if the algorithm was not newly developed. (I.e., a student's reimplementation of the algorithm to find the minimum value.) ● Mathematical and logical concepts can be a part of the selected algorithm or part of either of the included algorithms. **Do NOT award a point if any one of the following is true:** ● the selected algorithm consists of a single instruction; or ● the selected algorithm consists solely of library calls to existing language functionality; or ● the selected algorithm does not include mathematical or logical concepts; ● the response only describes what the selected algorithm does without explaining how it does it; or ● the response does not explicitly address the program's purpose; or ● the code segment consisting of the selected algorithm is not included in the written responses section or is not explicitly identified in the program code section; or ● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).

Row 6 - Response 2C Selected code segment implements an algorithm that includes at least two or more algorithms **AND** At least one of the included algorithms uses mathematical or logical concepts **AND** Explains how one of the included algorithms functions independently.	● Responses are still eligible to earn this row, even if they do not earn row 5. ● The included algorithms can be sub-parts of the algorithm in row 5. **Do NOT award a point if any one of the following is true:** ● the selected algorithm consists of a single instruction; or ● the selected algorithm consists solely of library calls to existing language functionality; or ● neither of the included algorithms nor the selected algorithm that includes two or more algorithms uses mathematical or logical concepts; or ● the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or ● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).

Student Response A - [Video] [Written Response]	**Scoring Guidelines**	

world.wolfMove ([Obj] chicken)
No variables
// moves wolf to blue chickens and "eats" them
If (chicken . color == [redacted])
wolf turn to face chicken
While ((wolf distance to chicken) > 1)
wolf move amount = 1 meter toward target = chicken duration = .08 seconds
wolf move up 0.1 meters duration = 0 seconds
wolf move to chicken duration = 0 seconds
wolf move up (((subject = wolf 's height) / 2)) duration = 0.1 seconds
Do together
chicken set opacity to 0 (0%)
chicken set color to [redacted]
// the counting method is called here
world.scoreCounter
// color reset method below
world.resetColors
Else
Do Nothing
	Row 4	**1**

	The response earned a point for this row. The code that is given represents an algorithm.	
	Row 5	**1**
	The response earned a point for this row. The response includes an algorithm that has math/logic (while loop, height/2). The response explains how the wolfMove algorithm works: "The method moves the wolf to the blue chickens and then the chickens' opacity is set to zero." The response describes what the algorithm does with respect to the entire program. It "contains the code that tells my wolf how to move."	

world.resetColors ( )
No variables
// keeps track of the number of blue chickens and resets the colors if there are none left
world.numBlue set value to 0
For all world.chickens , every item_from_chickens together
If (item_from_chickens . color == [redacted])
world.numBlue set value to ((world.numBlue + 1))
Else
Do Nothing
For all world.chickens , every item_from_chickens together
If (world.numBlue == 0)
world.setrandomcolor chicken = item_from_chickens
Else
Do Nothing
	Row 6	**1**

	The response earned a point for this row. The response gives an algorithm (wolfMove) that uses two algorithms (resetColors and scoreCounter). The response explains how resetColors works: "count the number of blue chickens and proceeds to use logic and call my "setrandomcolors" ... method if there are no blue chickens left." The resetColors function includes logic using if statements.	
world.scoreCounter ( ) *No variables* // counts the number of chickens eaten increment world.counter by 1 chickenCounter set text to (world.counter as a string)		

I chose to use my “wolfMove” method, which I wrote independently, that contains the code that tells my wolf how to move, as well as two other methods “resetColors” and “scoreCounter”. “wolfMove” uses math to tell the wolf how to move towards the blue chickens and sets their opacity to zero. The method moves the wolf to the blue chickens and then the chickens’ opacity is set to zero. The “resetColors” method uses math to count the number of blue chickens and proceeds to use logic and call my “setrandomcolors” (I will talk more about this method in 2D) method if there are no blue chickens left. My “scoreCounter” method uses math to count the number of chickens that have been “eaten” or disappeared.

Student Response B - [Video] [Written Response]

```
for (a, b) in users_dict.items():
    guess = str(input("Input the term that has the definition of " + b + ": "))
    if guess.lower() == a.lower():
        points_scored += 1
        streak += 1
    else:
        words_missed.append(a)
        streak = 0
    if streak >= 2:
        if streak % 2 == 1:
            print("You're on fire!! YAYY...\n")
        else:
            print("You're doing great! Keep it up!\n")
    elif streak == 1:
        print("Keep it up!! You're almost there...\n")
    else:
        print("Oh come on, I know that you can do it...\n")
```

An algorithm in my program is a for-loop that includes 3 sets of if-else statements. The for-loop iterates over each item in the dictionary and asks the user to enter the term which corresponds to a given definition. The 1st set of if-else-statements functions by adding 1 each to user’s score and streak-point if the user’s answer = key of the dictionary-item. Else, it resets streak to 0. This demonstrates use of logical concepts since if-statements use Booleans. The 2nd set of if-else-statements determine which motivational-message to print based on their streak-points. If the streak-point >= 2, it runs the algorithm’s 3rd set of if-else-statements, which uses modulus—a mathematical concept—to determine which motivational message to display by using modulus 2 to determine the parity of the streak-score. Additionally, if streak-point < 2, the algorithm prints out a message that tells the user to try harder. All in all, this entire algorithm helps to achieve intended purpose of the program by quizzing the user, and the 3 smaller algorithms (the if-else-statements) included within the algorithm determines the score and streak-points of the user and uses that information to print adequate motivational messages. This entire algorithm’s developed independently. .

Scoring Guidelines

Row 4	**1**

The response earned a point for this row. The circled code segment provided in the response represents an algorithm.

Row 5	**1**

The response earned a point for this row. The response includes math/logic through the use of if statements. The response explains how it works overall ("The for-loop iterates over each item in the dictionary and asks the user to enter the term which corresponds to a given definition"). The response also describes what this algorithm does in relation to the entire program ("quizzing the user, and the three smaller algorithms (the if-else-statements) included within the algorithm determines the score and streak-points of the user and uses that information to print adequate motivational messages”).

Row 6	**1**

The response earned a point for this row. The response earned the point for this row. The response explains that the three if-else structures are the independent algorithms within the main algorithm. Each of these is explained and each uses logic (an if statement along with math).

Student Response C - [Video] [Written Response]

Scoring Guidelines

Row 4	**1**

```
private void turnCard()
{
    if (pressVal == rVal)
    {
        setX();
    }
    else
    {
        setCheck();
    }
}
```

This algorithm is the central decision making part of my program. Located in the Card class and called through each card's act method, the turnCard method will set the image to either a check or x. Each card is instantiated with their order value (1-9), which is shown on the picture of the card. In the Background class, I set a variable to a random integer value, which every card is instantiated with as a parameter. In the constructor, the order value and the random value are assigned to the class's private instance variables. The order value of the card is then compared with the random value that decides the "x" card as a parameter. This method tests if the card's value is the same as the "x" value, then decides if it should call the setX() method or the Create Sample C 1 of 3 setCheck() method.

```
public void setX()
{
    cardVal = false; //false = x
    setImage(new GreenfootImage("x.png"));
    GameOver g = new GameOver();
    getWorld().addObject(g, 315,345);
}
```

If the card's value is the same as the "x" value, it calls the setX method, which changes the image of the card to an "x" and creates a new "GameOver" object and adds it to the middle of the background.

```
public void setCheck()
{
    cardVal = true; //true = check
    setImage(new GreenfootImage("check.png"));
}
```

If the card's value doesn't match the "x" value, the algorithm calls the setCheck method to just set to the "check" image (the player can still flip more cards.) If wished, these two smaller methods could be called independently of the algorithm by a Card object.

The response earned a point for this row. The code segment provided is an algorithm.

Row 5	**1**

The response earned a point for this row. The algorithm provided uses math/logic (if statement). The response explains how the algorithm works: "I set a variable to a random integer value, which every card is instantiated with as a parameter ... this method tests if the card's value is the same as the "x" value, then decides if it should call the setX() method or the setCheck() method." The response also describes what the purpose is for this algorithm in relation to the entire program: "The turnCard method will set the image to either a check or x."

Row 6	**0**

The response DID NOT earn a point for this row. The response identifies two algorithms used by the turnCard method, but the two methods do not use math/logic. Setting a variable to true or false is not considered use of logic, just as x = 4 is not considered use of math.

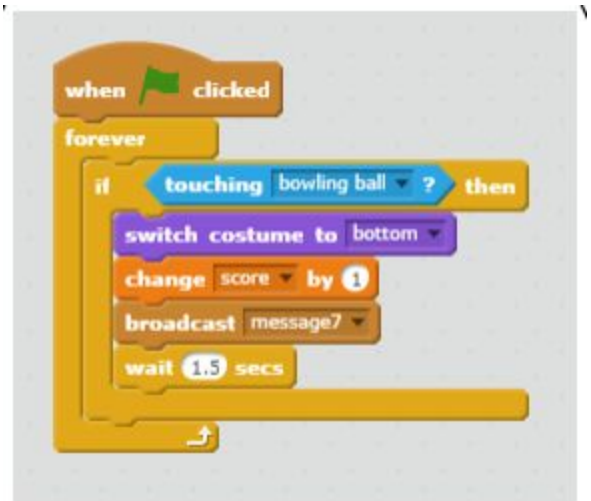
Code.org Commentary: According to the Scoring Notes, this row could have received the point because the selected algorithm contained an if-statement. However, it's best to use math or logic in at least one of the included algorithms to ensure the point.

The point should NOT be receive if:

- neither of the included algorithms nor the selected algorithm that includes two or more algorithms uses mathematical or logical concepts; or

Student Response D - [Artifact] [Written Response]

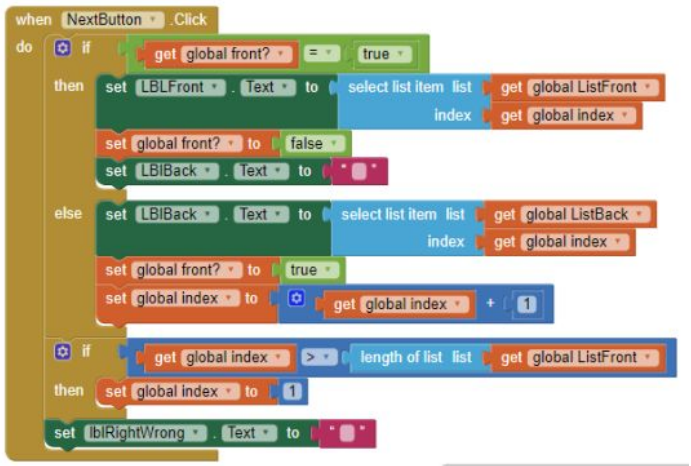
Scoring Guidelines



This algorithm was one that was fundamentally important in aiding the program to successfully run. In this algorithm, the bowling pin is sensing whether the bowling ball is touching it or not and if it is, the costume of the bowling ball switches from a regular bowling pin to the bottom of a bowling pin to make it look as if the bowling pin has fallen. Next, the score is changed by 1, meaning that every time the bowling ball touches a bowling pin, the score is increased by one point. After that, “message7” is broadcasted so that when the bowling ball receives “message7”, it will immediately disappear, wait 1.5 seconds, and then return to its original position so that the user can continue playing the game. This algorithm functions independently because it is solely for the purpose of making sure that the bowling pin is knocked down. Also, in combination with others, this algorithm helps determine the course of the game because if the bowling pin isn’t being touched, the algorithm won’t run. This algorithm allows other algorithms to be notified that if the bowling pin is being touched, the rest of the program should continue moving forward.

Row 4	1
The response earned a point for this row. The block selected represents an algorithm.	
Row 5	**1**
The response earned a point for this row. The algorithm uses mathematical and logical concept via the score change by adding 1 in the if statement. The response explains how this function works in detail. The response also describes that this algorithm's purpose is, "sensing whether the bowling ball is touching it or not."	
Row 6	**0**
The response DID NOT earn a point for this row. The algorithm given does not include two or more additional algorithms that function independently.	

Student Response E - [Video] [Written Response]



Scoring Guidelines

Row 4	1
The response earned a point for this row. The code segment provided is an algorithm.	
Row 5	**1**
The response earned a point for this row. The algorithm provided uses math/logic via if statements. The response explains how the algorithm works: "When the next button is clicked, it displays the label font text and then doesn't show the back text. The index also determined which flashcard that you are on, in which you keep going to the next term as the next button is clicked. Furthermore, if the index is bigger than the number of items in the list then it restarts back to 1, or the first item in	

This particular algorithm is essential to the program because it allows the user to go to the next flashcard. When the next button is clicked, it displays the label font text and then doesn't show the back text. The index also determined which flashcard that you are on, in which you keep going to the next term as the next button is clicked. Furthermore, if the index is bigger than the number of items in the list then it restarts back to 1, or the first item in the list. This is the same for the other algorithm as they both use an index. One of the independent algorithms makes so that user can type in the term as the other algorithm is displaying the definition as a flashcard. Together as a combination, this makes it so that the user has a study guide environment in which they can type in the necessary term to the definition and then be able to go to the next set of terms.

the list. This is the same for the other algorithm as they both use an index." The response also describes what the purpose is for this algorithm in relation to the entire program: "allows the user to go to the next flashcard" and "the user has a study guide environment in which they can type in the necessary term to the definition and then be able to go to the next set of terms."

Row 6	0

The response DID not earn a point for this row. The response briefly states that one algorithm "makes [it] so that user can type in the term as the other algorithm is displaying the definition as a flashcard." However, it is not clear where each of these algorithms is in the supplied code segment, so it is not clear if these algorithms are included in the identified algorithm.

Student Response F - [Video] [Written Response]

```
onEvent("btnPlay", "click", function(event) {
   setScreen("playScreen");
   Time();
});
onEvent("Catimage", "click", function(event) {
   setPosition("Catimage", randomNumber(0,200),
      randomNumber(0,395));
   getText("Score", Score);
   Score = Score + 1;
   setText("Score", Score);
});
onEvent("btnRestart", "click", function(event) {
   setScreen("playScreen");
   seconds = 30; Score = 0;
   setText("Time", seconds);
   setText("Score", Score);
   Time();
});
onEvent("btnPause", "click", function(event) {
   setScreen("PauseScreen");
   clearInterval(myInterval);
});
onEvent("btnResume", "click", function(event) {
   setScreen("playScreen");
   Time();
});
onEvent("btnRestart1", "click", function(event) {
   setScreen("playScreen");
   seconds = 30;
   Score = 0;
   setText("Time", seconds);
   setText("Score", Score);
   Time();
});
```

I used this algorithm in order to set the code in an organized way, such that it would work efficiently and when the a certain button is pressed, it would execute the code written for the button in the event handler. For example when the event handler “Catimage” was called, it would execute the code in that event handler when the Nyan Cat image was clicked, then adding adding 1 point to the score and setting the text of the score to update

Scoring Guidelines

Row 4	1

The response earned a point for this row. The code given in the response is an algorithm because it involves sequencing, selection and/or iteration.

Row 5	0

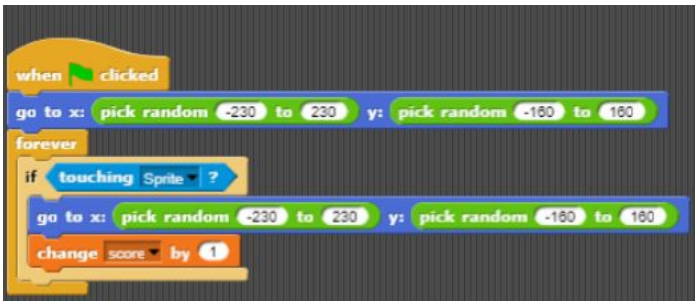
The response DID NOT earn a point for this row. Although the algorithm given includes math minimally (incrementing the score) and it explains how the functions work, it does not describe what this algorithm does in relation to the overall program.

Row 6	0

The response DID NOT earn a point for this row. The response does not clearly identify two algorithms used by the selected algorithm.

everytime the Nyan Cat was clicked such that the score would keep increasing. Something else to notice is the Time function being called in several event handlers. This was done in order to set the timer to restart or to start again depending on the event handler that is executed.

Student Response G - [Video] [Written Response]

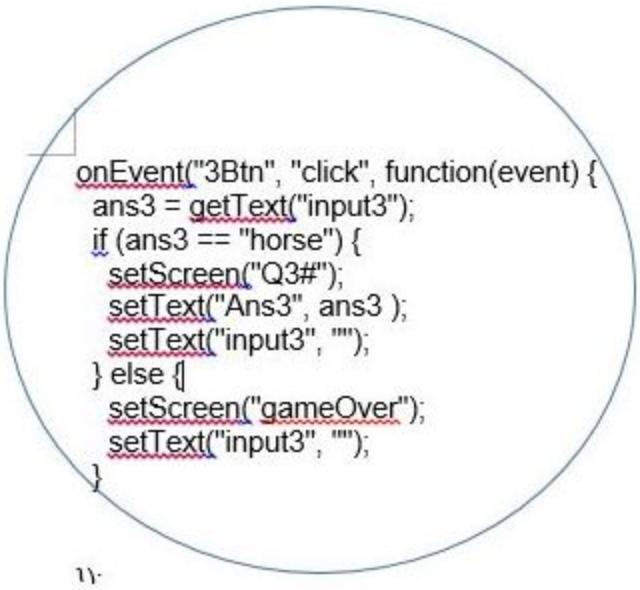


The purpose of this algorithm is to make the apple appear in a new place. Each time the snake gets an apple, it reappears somewhere else. When the snake touches the the sprite that is the apple, the random number function makes it reapear any where between x: -230 to230 and y: -160 to 160. This also adds to the difficulty of the game the closer the apple goes to the edges, because the snake has to avoid the edges or the game will end.

Scoring Guidelines

Row 4	**1**
The response earned a point for this row. The selected block of code represents an algorithm.	
Row 5	**1**
The response earned a point for this row. The included code segment contains math/logic by adding the score to 1. The response explains how the algorithm works: "When the snake touches the the [sic] sprite that is the apple, the random number function makes it reapear any where [sic] between x: -230 to 230 and y: -160 to 160." The response describes what this algorithm does in relation to the program: "to make the apple appear in a new place."	
Row 6	**0**
The response DID NOT earn a point for this row. The response does not identify two algorithms that are used within the selected algorithm nor how one of them works.	

Student Response H - [Video] [Written Response]



```
onEvent("3Btn", "click", function(event) {
  ans3 = getText("input3");
  if (ans3 == "horse") {
    setScreen("Q3#");
    setText("Ans3", ans3 );
    setText("input3", "");
  } else {
    setScreen("gameOver");
    setText("input3", "");
  }
```

The two algorithms within the selected algorithm are the onEvent function (to move on to the next screen) and the if… else…. statement. This algorithm is used when the user must type in a response. The first algorithm allows for the check button to be clicked and either move on to the next screen which shows the correct answer and allows

Scoring Guidelines

Row 4	**1**
The response earned a point for this row. The response identifies a segment of code that is an algorithm.	
Row 5	**1**
The response earned a point for this row. The response uses math and/or logic (an if statement). The response states what the algorithm does in relation to the program ("used when the user must type in a response"). The response explains how the algorithm works ("There is only one correct answer, in this case it is horse. If the user types in horse exactly then they got it right and they move on; anything other than horse will be counted as wrong and the user will have to restart.").	
Row 6	**0**
The response DID NOT earn a point for this row. The response identifies the event handler as the first algorithm, and the if/else statement as a second algorithm. This does not meet the requirements for this row. The response needs to identify an algorithm that uses two additional algorithms that function independently.	

the user to continue the quiz, or go to the game over screen when the user has to restart. The second algorithm uses logical concepts and allows for the user's input to be checked and verified by using an if/else statement. There is only one correct answer, in this case it is horse. If the user types in horse exactly then they got it right and they move on; anything other than horse will be counted as wrong and the user will have to restart. Another small part of the algorithm is that everytime the check button is clicked, the input text box is reset to be blank for the next time around. The two algorithms together make the program run smoothly as it is supposed to.

Student Response I - [Video] [Written Response]

```
1  //line 2 of code makes the timer run when start button is pressed
2  onEvent(▼"start", ▼"click", function(){
3    //Line 4 of code sets the countdown slot to the value selceted by user from the dropdown section
4    var countdown = getText(▼"timer");
5    //The code below displays the algorithm where the imput value will be substracted one
6    //and be displayed, repeating this until the value equals 0
7    var i = setInterval( function() {
8      //Line 9 to 10 substracts one from countdown and displays the value on the countdown slot
9      countdown = countdown - 1;
10     setText(▼"countdown", countdown);
11     if( countdown === 0 ) {
12       //Line 13 of code renews the value displayed in countdown slot in correlation to setInterval
13       clearInterval(i);
14       //Line 15 of code implements the function to make the countdown slot blink
15       blinkCountdown();
16     }
17     //Algorithm segment setInterval loops through the function every 1000 millisecond
18     }, 1000);
19   //Line 20 of code displays the end result in the countdown slot
20   console.log("Interval timer ID: " + i);
21 });
```

The red oval in the image captures the algorithm which is essential for the function of the program. Within it there is an algorithm in lines 9-10, which takes the value given to countdown by the user from the dropdown textbox and subtracts one from it and then displays Create Sample I 1 of 2 the result on the countdown slot. This algorithm was developed independently. The second algorithm begins in code line 7 as setInterval, and its value is displayed in line 18 of the code as 1000 milliseconds. The value of the setInterval was also developed independently. Both algorithms are essential because in unison they allow the program to work, as without the first algorithm the code would not work as nothing would be displayed nor nothing would be subtracted so the function would eventually reach zero. While without the second algorithm result would always be one less then the value set for countdown and there be no command telling it to repeat the process in algorithm number one.

Scoring Guidelines

Row 4	**1**
The response earned a point for this row. The selected block of code represents an algorithm.	
Row 5	**1**
The response earned a point for this row. The circled code contains math/logic. The response explains how the algorithm works in detail. The response describes what this algorithm does in relation to the program: "Both algorithms are essential because in unison they allow the program to work" AND then **states what would happen to the program if either part is missing**. By explaining what happens if each part is missing, the response is explaining what these algorithms do for the whole program.	
Row 6	**0**
The response DID NOT earn a point for this row. The response identifies two algorithms, one of which is inside the other. There is no additional algorithm identified.	

Student Response J - [Video] [Written Response]

```
onEvent("WelcomeScreen", "click", function(event) {
   console.log("WelcomeScreen clicked!");
});
onEvent("football", "click", function(event) {
   console.log("football clicked!");
 });
onEvent("football", "click", function(event) {
   setScreen("Footballq1");
```

Scoring Guidelines

Row 4	**0**
The response DID NOT earn a point for this row. The code given in the response is not considered an algorithm because each event handler consists of a single	

```
});
onEvent("Footballq1", "click", function(event) {
    console.log("Footballq1 clicked!");
});
onEvent("Footballq2", "click", function(event) {
    console.log("Footballq2 clicked!");
});
onEvent("abbutton", "click", function(event) {
    setScreen("Footballq2");
});
onEvent("Gobackbutton", "click", function(event) {
    console.log("Gobackbutton clicked!");
});
onEvent("Gobackbutton", "click", function(event) {
    setScreen("WelcomeScreen");
});
onEvent("Baseballq1", "click", function(event) {
    console.log("Baseballq1 clicked!");
});
onEvent("baseball", "click", function(event) {
    setScreen("Baseballq1");
});
onEvent("gobackbutton2", "click", function(event) {
    console.log("gobackbutton2 clicked!");
});
onEvent("gobackbutton2", "click", function(event) {
    setScreen("WelcomeScreen");
});
onEvent("nybutton", "click", function(event) {
    setScreen("Baseballq2");
});
onEvent("basketball", "click", function(event) {
    setScreen("Basketballq1");
});
onEvent("button11", "click", function(event) {
    setScreen("Basketballq2");
});
onEvent("button17", "click", function(event) {
    setScreen("WelcomeScreen");
});
onEvent("Soccer", "click", function(event) {
    setScreen("Soccerq1");
});
onEvent("button5", "click", function(event) {
    setScreen("Soccerq2");
});
onEvent("button20", "click", function(event) {
    setScreen("WelcomeScreen");
});
```

Within the code, the algorithm is essential to the program by each sport having it's own command or action. it controls which sport you want to go to. The code represents the amount of time it takes you to answer a question, taking too long might make the game go back to the welcome screen so there's limited time to play. Picking the right answer will get you to the next question automatically. From there it's just repetitive for the rest of game, for all sports. All buttons are on click actions, as well, just know whatever sport you pick you must have a little background knowledge on it.

instruction that does not involve sequencing, selection, or iteration.

Row 5	**0**

The response DID NOT earn a point for this row. The provided code does not use math or logic, and the code segment is not considered an algorithm.

Row 6	**0**

The response DID NOT earn a point for this row. The response did not earn the point for this row. The response does not clearly identify two additional algorithms that are used by the selected algorithm. This cannot be done because the selected code is not considered an algorithm.

Create PT - 2d

2d. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a **rectangle** in **section 3** below). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. (*Must not exceed 200 words)*

Scoring Guidelines	
Row and Task	**Decision Rules**
Row 7 - Code Segment in Response 2D Selected code segment is a student-developed abstraction.	● Responses that use existing abstractions to create a new abstraction, such as creating a list to represent a collection (e.g., a classroom, an inventory), would earn this point. **Do NOT award a point if any one of the following is true:** ● the response is an existing abstraction such as variables, existing control structures, event handlers, APIs; or ● the code segment consisting of the abstraction is not included in the written responses section or is not explicitly identified in the program code section; or ● the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction).
Row 8 - Response 2D Explains how the selected abstraction manages the complexity of the program..	● Responses should not be penalized for explanations of abstractions that are not developed by the student. **Do NOT award a point if any one of the following is true:** ● the explanation does not apply to the selected abstraction; or ● the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction).

Student Response A - [Video] [Written Response]	Scoring Guidelines

```
world.setrandomcolor ( [Obj] chicken)
 randomNum = 1
  // Gives all the chickens different colors
  randomNum set value to ( random number minimum = 1 maximum = 9 integerOnly =
 true )
  If ( randomNum == 1 )
     chicken set color to [red] duration = 0.5 seconds
  Else
     If ( randomNum == 2 )
        chicken set color to [blue] duration = 0.5 seconds
     Else
        If ( randomNum == 3 )
           chicken set color to [magenta] duration = 0.5 seconds
        Else
           If ( randomNum == 4 )
              chicken set color to [blue] duration = 0.5 seconds
           Else
              If ( randomNum == 5 )
                 chicken set color to [red] duration = 0.5 seconds
              Else
                 If ( randomNum == 6 )
                    chicken set color to [blue] duration = 0.5 seconds
                 Else
                    If ( randomNum == 7 )
                       chicken set color to [magenta] duration = 0.5 seconds
                    Else
                       If ( randomNum == 8 )
                          chicken set color to [orange] duration = 0.5 seconds
                       Else
                          If ( randomNum == 9 )
                             chicken set color to [blue] duration = 0.5 seconds
                          Else
                             Do Nothing
```

The abstraction I chose is my method “setrandomcolor.” This method uses math to set a random number to each of the chickens that are used in my game. The method then uses logical if/else statements to assign different colors to the range of numbers that are given to the chickens. This abstraction helped manage the complexity of my program because it ensured that I would not have to put together the large amount of code every time that I called the method. I call this method once in the beginning of my program and then later if there are not any blue chickens left. Because of this abstraction I did not have to re-write the same ten if/else statements more than once.

Row 7	1

The response earned a point for this row. The boxed code segment in the response represents a valid abstraction (a procedure or method).

Row 8	1

The response earned a point for this row. The response explains how the abstraction manages complexity: "I did not have to re-write the same ten if/else statements more than once" and "it ensured that I would not have to put together the large amount of code every time that I called the method."

Student Response B - [Video] [Written Response]	Scoring Guidelines

```python
def quiz():
    for y in range(300):
        print()
    points_scored = 0
    words_missed = []
    print("QUIZ TIME!!!")
    streak = 0
    for (a, b) in users_dict.items():
        guess = str(input("Input the term that has the definition of " + b + ": "))
        if guess.lower() == a.lower():
            points_scored += 1
            streak += 1
        else:
            words_missed.append(a)
            streak = 0
        if streak >= 2:
            if streak % 2 == 1:
                print("You're on fire!! YAYY...\n")
            else:
                print("You're doing great! Keep it up!\n")
        elif streak == 1:
            print("Keep it up!! You're almost there...\n")
        else:
            print("Oh come on, I know that you can do it...\n")
    print("\nGame over. You got", points_scored, "outta", num_terms, "!!!")
    return words_missed
```

Row 7	1

The response earned a point for this row. The code given in the response represents an abstraction (a procedure).

Row 8	1

The response earned a point for this row. The response explains how this abstraction manages complexity by "improving readability ... and making debugging easier."

An abstraction developed individually is quiz(), which is used in the function run_quiz(). quiz() quizzes the user , records terms the user messes up on, calculates total points_scored and streak-points, and gives the user motivational messages based on their streak-points. Example of using a mathematical concept is using modulus-2 to determine the parity of the user's streak-points. Example of a logical concept is an if-else statement that determines if the user answered correctly. My function does this by returning a Boolean value (True or False) which describes whether the user's answer equals correct answer, and using that value to determine whether or not to run add 1 each to points_scored and streak-points. Having this abstraction helped manage the complexity of this program by one, improving readability, and two, making debugging easier. Firstly, since all the code regarding the quiz is in a function, running the quiz itself only requires quiz(). Second, this abstraction made debugging significantly easier. An example of this is when I found out that the quiz was not functioning the way I had expected it to, so I immediately started looking for errors in the quiz() function rather than having to look through my entire program for the error.

Student Response C - [Video] [Written Response]

```
int numCard = 1;
for(Card c: cardList)
 {
     addObject(c,x,y);
     numCard++;
     if(numCard ==4 || numCard ==7)
     {
         x=105;
         y+= 230;
     }
     else
     {
         x+=210;
     }
 }

}

}
```

I created a list consisting of 9 cards to apply abstraction. The iterative process of going through each item in the list increases code efficiency because I no longer had to add every card to the background separately (I had around 9 separate addObject statements and 9 separate calls to the card's act() method.) I made a counter variable which I use to decide what coordinates to update. For example, if the program has already added 3 cards, it must know that it needs to start the second row. I also made variables for the x and y coordinate. If the counter variable equals 4 or 7, I would need to add 230 to the y coordinate variable

Scoring Guidelines

Row 7	**1**

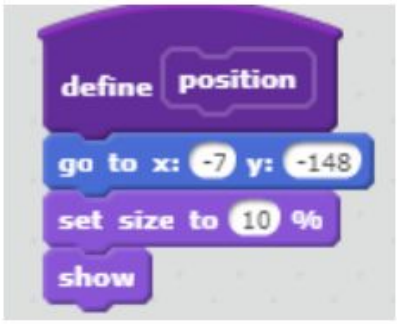
The response earned a point for this row. The selected code segment shows the use of a list (cardList) to create a new abstraction to represent a collection of cards.

Row 8	**1**

The response earned a point for this row. The response explains how the abstraction manages program complexity: "the iterative process of going through each item in the list increases code efficiency because I no longer had to add every card to the background separately ..." Use of the list helps manage complexity by allowing for the use of a loop to process the list, reducing the amount of coding as explained: "I had around 9 separate addObject statements and 9 separate calls to the card's act() method."

and reset the x coordinate variable to 105, Create Sample C 2 of 3 which sets the image to the far left. Using those updated variables, the program then adds the card object.

Student Response D - [Artifact] [Written Response]



While I was writing the program for my bowling game, I realized something. Every time the bowling ball touched a bowling pin, I programmed it to hide, go back to its original position, set the size to the original 10%, and then be visible again, I noticed that this code could be made shorter and simpler. I figured that if I made a custom block called “position”, I could fit all of those blocks of code into “position” and make it an abstraction block. This helped me significantly because it ensured that every time I would look at the code, I wouldn’t get confused to see the same things repeating over and over again and it would make the code seem simpler and more human friendly. It also helped manage the complexity of my program because every time I looked at the block of code “position”, it made me understand that the block was simply asking for the bowling ball to be returned back to its original position so that the user could roll the ball again.

Scoring Guidelines

Row 7	**1**
The response earned a point for this row. The block given is an abstraction (a procedure).	
Row 8	**1**
The response earned a point for this row. The response explains how complexity is managed: "I noticed this code could be made shorter and simpler ... it ensured that every time I would look at the code, I wouldn't get confused to see the same things repeated over and over."	

Student Response E - [Video] [Written Response]

This particular abstraction is used to determine if the word you type in, is correct. This abstraction uses mathematical concepts by determining if the word you type in and the actual term, are equal. If they do happen to be equal, then this will be shown in the “lblWriteWrong.” Furthermore, this abstraction uses logical concepts by determining if the word the user types in is true, then it will be displayed as correct through the Write Wrong label. However if the word the user types in is a false word, then it will show that

Scoring Guidelines

Row 7	**1**
The response earned a point for this row. The supplied code block represents a student-developed abstraction, which is a procedure.	
Row 8	**1**
The response earned a point for this row. The response gives a reason why the abstraction manages complexity: "By creating this abstraction it makes the general coding clearer and easier to read as it is already being used once."	

it is incorrect through the Write Wrong label. By creating this abstraction it makes the general coding clearer and easier to read as it is already being used once.

Student Response F - [Video] [Written Response]

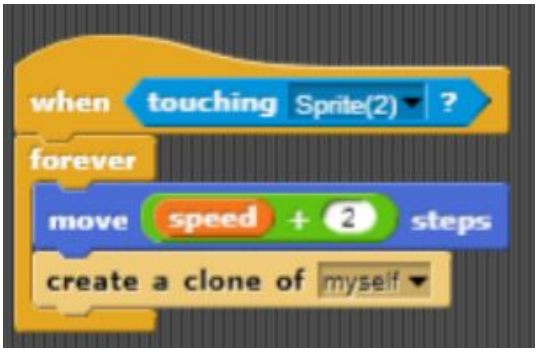
```
function Time() {
   myInterval = setInterval(function() {
      seconds = seconds - 1;
      setText("Time",seconds);
      if (seconds === 0){
         if (Score > HighScore) {
            HighScore = Score;
            setKeyValue("HighScore", Score, function () {
               setText("HighScore", HighScore);
            });
         }
         setText("txtScore","You have received a score of " + Score + "
Nyan Cats that you clicked!!");
         setScreen("GameOverScreen");
         clearInterval(myInterval);
      }
   }, 1000);
}
```

In order to use the , such as certain lines of code are not repeated, I made my code use abstraction is by using this Time() function, and implementing it into the event handlers so the code is reduced and not being repeated on certain event handlers. My first thought for my Time() function was using the setInterval() function and the if statement and applying it to certain event handlers over again, such that the 30 second interval would run, and the screen would change when the 30 seconds would end. Using abstraction helped me implement the setInterval() function and the if statement such that it would be unnecessary to repeat code and have extra lines of code, adding less complexity to the algorithms that I used into making this code.

Scoring Guidelines

Row 7	**1**
The response earned a point for this row. The code segment given represents an abstraction (a procedure or function).	
Row 8	**1**
The response earned a point for this row. The response explains how the abstraction manages complexity for the program by stating: "the code is reduced and not being repeated on certain event handlers."	

Student Response G - [Video] [Written Response]



The purpose of this abstraction is to cause the snake to grow every time it eats an apple. This is done by ceating a new clone that will follow the snake every time it eats an apple. This abstraction also increases the speed of the snake each time it collects an apple. This causes it to be more chanllenging to get the apple and also avoid the edges.

Scoring Guidelines

Row 7	**1**
The response earned a point for this row. The boxed code fragment does not represent an abstraction.	
Row 8	**0**
The response DID NOT earn a point for this row. The response does not explain how the boxed code manages complexity in the program. Making the program "more challenging" is not the purpose of an abstraction.	

Student Response H - [Video] [Written Response]	Scoring Guidelines

//
//
//if wrong answer is clicked -- leads back to the beginning
onEvent("oops", "click", function(event) {
setScreen("Q1");
});
//
//
 	
The purpose of my program is to educate and encourage people to learn foreign languages. To do that, my game/quiz asks questions, and the user must select one of four answers, one is in Spanish, one is in French, one is in Japanese, and one is in German. Because repetition is key for learning a new language, I decided to incorporate it into my program. Except for three questions where the user types in a response, the rest are multiple choice questions. For each question, there is only one correct answer, so when any one of the incorrect answers is clicked, the game over screen comes up. On that screen the user then clicks on a button to try again. The trick of the game is that every time an incorrect answer is clicked, it brings you back to the first question. This program code segment allows the game to work as a constant loop, always waiting for the game over screen to appear.
	Row 7 — **0**
	The response DID NOT earn a point for this row. The code fragment given is not a student-developed abstraction. The abstraction given, an event handler, is built in to this language.
	Row 8 — **0**
	The response DID NOT earn a point for this row. The response does not explain how the supplied abstraction manages complexity for the program. The response only explains how the code fragment works in relation to the entire program.

Student Response I - [Video] [Written Response]	Scoring Guidelines


1 //line 2 of code makes the timer run when start button is pressed
2 onEvent(▼"start", ▼"click", function(){
3 //Line 4 of code sets the countdown slot to the value selceted by user from the dropdown section
4 var countdown = getText(▼"timer");
5 //The code below displays the algorithm where the imput value will be substracted one
6 //and be displayed, repeating this until the value equals 0
7 var i = setInterval(function() {
8 //Line 9 to 10 substracts one from countdown and displays the value on the countdown slot
9 countdown = countdown - 1;
10 setText(▼"countdown", countdown);
11 if(countdown === 0) {
12 //Line 13 of code renews the value displayed in countdown slot in correlation to setInterval
13 clearInterval(i);
14 //Line 15 of code implements the function to make the countdown slot blink
15 blinkCountdown();
16 }
17 //Algorithm segment setInterval loops through the function every 1000 millisecond
18 }, 1000);
19 //Line 20 of code displays the end result in the countdown slot
20 console.log("Interval timer ID: " + i);
21 });
	Row 7 — **0**
	The response DID NOT earn a point for this row. The boxed code fragment does not represent an abstraction.
	Row 8 — **0**
	The response earned a point for this row. The response does not explain how the boxed code manages complexity in the program. The response merely states how the code fragment works.

The blue rectangle in the image showcases a code segment which implements a mathematical function to the program which was developed independently. The code takes the value given to countdown by the user in the dropdown textbox or code getText from line 4 and subtracts one from it. After so, the code segment displays the result in the countdown slot on the screen. This code segment is part of a function which serves as an algorithm to repeat the code segment. Without the code segment the hole program would not run as there would be no segment giving it the means to subtract and reach zero, therefore the displayed value would always be the one input by the user.

Student Response J - [Video] [Written Response]	Scoring Guidelines
 This manages the app so you get where you want to be questioned on. Allowing you to go to the home screen, after you finish your question. The game has a go back button which allows you to go home after you finish. But it was complex making the game move automatically to the next question after you answer it correctly. This was great experience making this computational artifact.	**Row 7** **0** **The response DID NOT earn a point for this row.** The code segment given does not represent a student-defined abstraction. An event handler is built-in to this language. **Row 8** **0** **The response DID NOT earn a point for this row.** The response does not explain how this abstraction chosen manages complexity of the program. The response states "It was complex making the game move automatically to the next question after you answer it correctly." This is not the correct meaning of complexity for this row.

```
31 onEvent("gobackbutton2", "click", function(event) {
32   console.log("gobackbutton2 clicked!");
33 });
34 onEvent("gobackbutton2", "click", function(event) {
35   setScreen("WelcomeScreen");
36 });
37 onEvent("nybutton", "click", function(event) {
38   setScreen("Baseballq2");
39 });
40 onEvent("basketball", "click", function(event) {
41   setScreen("Basketballq1");
42 });
43 onEvent("button11", "click", function(event) {
44   setScreen("Basketballq2");
45 });
46 onEvent("button17", "click", function(event) {
47   setScreen("WelcomeScreen");
48 });
49 onEvent("Soccer", "click", function(event) {
50   setScreen("Soccerq1");
51 });
52 onEvent("button5", "click", function(event) {
53   setScreen("Soccerq2");
54 });
55 onEvent("button20", "click", function(event) {
56   setScreen("WelcomeScreen");
57 });
```

Create PT - 2017 Sample C - Score: 7/8

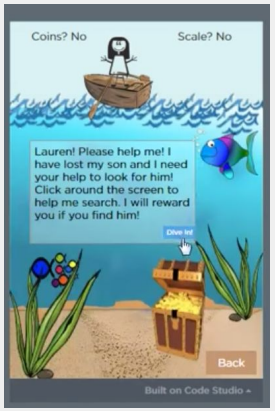
Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6	Row 7	Row 8	*This document combines student sample, scoring guidelines and scoring commentary from: Create PT 2017 Sample C*
Sample: C	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**1**	

Video
Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. **Your video must not exceed 1 minute in length and must not exceed 30MB in size.**

Program Purpose and Development
2a. Provide a written response or audio narration in your video that:

- identifies the programming language
- identifies the purpose of your program; and
- Explains what the video illustrates.

(Must not exceed 150 words)

Student Response	Scoring Guidelines	
	Row and Task	**Decision Rules**
 *My program is an interactive game created with Javascript. The purpose of this game is to create something fun that interacts with the user by incorporating an adventure that the player completes by performing certain tasks or winning*	**Row 1** **Response 2A** The video demonstrates the running of at least one feature of the program submitted. **AND** The response (audio narration or written response) identifies the purpose of the program (what the program is attempting to do).	Response earns the point if it explains the function of the program instead of identifying the purpose. Response earns the point if the illustrated feature runs, even if it does not function as intended. Response earns the point if the response is included in the video via narration or some form of closed captioning and addresses the purpose or function of the program. **Do NOT award a point if any one of the following is true:** ● a video is not submitted; ● the video does not illustrate the feature mentioned in the response; or ● the video does not illustrate the running of the feature (screen shots or storyboards are not acceptable and would not be credited).

certain games within each adventure. Once all of the subgames on each level of the app are won, the user completes the adventure. The video shows one of two adventures that are incorporated in my game to display how the program tracks score within each mini game and on each overall level/adventure to determine when the overall game/adventure is won. The video also shows the functionality of the games and how they are played, in addition to the storyline format that the adventures follow.	**The response earned the point for this row.** The video is continuously running and demonstrates two adventures in the adventure game. The response indicates that the purpose of this program is to create a fun and interactive adventure game.

2b. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and/ or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development. *(Must not exceed 200 words)*

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**
I wrote this program, first by creating an identity for the user by incorporating user choice and a user input box to allow for personalization, and then by creating worlds with mini games within them for the user's chosen character to complete. My first problem arose when I tried to carry the username input from the character choice screen throughout the rest of the game. Because I had initially used a local variable, the user input was only recorded to the screen on which it was provided, instead of to the entire program. To resolve this issue, I changed the username input to a global variable that was able to be	**Row 2 - Response 2B** Describes or outlines steps used in the incremental and iterative development process to create the entire program.	**Do NOT award a point if any one of the following is true:** ● the response only includes the process for determining the program idea and does not address the development process used to create the entire program; or ● the response does not indicate iterative development; ● refinement and revision are not connected to feedback, testing, or reflection; or ● the response only describes the development at two specific points in time.
	The response earned the point for this row. The response describes the development process to create the entire program. The response states, "I wrote this program, first by creating an identity for the user...and then by creating worlds with mini games within them for the user's chosen character to complete." The response describes how they used an iterative development process by testing the use of the username throughout the game. "To resolve this issue, I changed the	

called on multiple screens in the program. I encountered a similar issue when I attempted to move the user's chosen character between screens. To make it appear as though the image traveled to the new screen as the user switched screens, I created a variable that stored the value of whichever character the user clicked on the first screen (i.e. boy= 1 when the boy image was clicked) and then used an algorithm that incorporated an if statement to set other images throughout the program to the user's character of choice. Both of these difficulties were handled independently.	username input to a global variable that was able to be called on multiple screens in the program." **Code.org Commentary:** While this response received the point in 2017, we are not fully convinced it would receive the point in 2019. More explanation of the incremental development (step by step) would ensure the point.	
	Row 3 - Response 2B Specifically identifies at least two program development difficulties or opportunities. **AND** Describes how the two identified difficulties or opportunities are resolved or incorporated.	Response earns the point if it identifies two opportunities, or two difficulties, or one opportunity and one difficulty AND describes how each is resolved or incorporated. **Do NOT award a point if any one of the following is true:** ● only one distinct difficulty or opportunity in the process is identified and described; or ● the response does not describe how the difficulties or opportunities were resolved or incorporated.
	The response earned the point for this row. The response describes two difficulties that were encountered and how these were resolved. The first difficulty is with the carrying of inputs to a different screen. This is resolved by using a global variable instead of a local variable. The second difficulty is the moving of the user's character between screens. This is resolved by using a variable and an if statement to set the image of the user's character throughout the program.	

2c. Capture and paste a program code segment that implements an algorithm (marked with an **oval** in **section 3** below) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. *(Must not exceed 200 words)*

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**

1. A.
onEvent("coin", "click", function(event) {
treasureScore = treasureScore+1;
setPosition("coin", randomNumber(105, 220), randomNumber(190, 250));
setText("coinCollected", "Number of coins collected: "+treasureScore);
if (treasureScore==20) {
setScreen("treasureWin");
gotCoins = 1;
}
});
onEvent("treasureContinue", "click", function(event) {
treasureScore = 0;
setScreen("underwaterAdventures");
didWinUnderwater();
if (gotCoins==1) {
setText("coins", "Coins? Yes");
}
});
 	
B.
onEvent("shell", "click", function(event) {
setScreen("fishWin");
scale = 1;
});
onEvent("fishContinue", "click", function(event) {
setScreen("underwaterAdventures");
didWinUnderwater();
if (scale==1) {
setText("scales", "Scale? Yes");
}
});
 	
2. function didWinUnderwater() {
if (gotCoins==1&&scale==1) {
setPosition("fishTalk", 25, 150, 210, 130);
showElement("fishTalk");
setText("fishTalk", "You have collected all of the elements you need to deliver to the shark. Would you like to proceed?");
showElement("yesUA");
showElement("noUA");
setPosition("underwaterCharacter", 250, 205);
reset(gotCoins, scale);
}
}
	Row 4
	Response 2C
	Selected code segment implements an algorithm.
		Do NOT award a point if any one of the following is true:
		● the algorithm consists of a single instruction;
		● the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or
		● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).

<table>
<tr><td rowspan="4">In the adventure “Underwater Adventures” there is a mini game in which the user must collect 20 coins to win. Once this task is complete, the user must play a second mini game in which he/she collects a scale. When both mini games are won, the Underwater Adventures level is complete. The code to execute this concept incorporates a variety of variables and functions to create an algorithm that calculates when the treasure game and scale games have been won, and when the overall level has been completed. The first set of code shown above (1.A.) is the function that runs the treasure game. Each time the coin is clicked, the variable treasureScore increases by 1. Once the score reaches 20, another variable, gotCoins, increases by 1 and the screen switches to a win screen with a continue button. The scale game works similarly, with a variable scale increasing by one if the game is won (1.B.). If the user clicks the continue button after winning either game, the second piece of code shown (2.) uses an algorithm to check if both games have been won by calling the function didWinUnderwater, which uses an if statement to determine if both gotCoins and scale have the value of 1.</td><td colspan="2">The response earned the point for this row.
The selected code segment implements an algorithm.

Code.org Commentary: The code segment is an algorithm because it includes sequencing (more than one step) and selection (if-statements).</td></tr>
<tr><td>Row 5
Response 2C

Selected code segment implements an algorithm that uses mathematical or logical concepts.
AND
Explains how the selected algorithm functions.
AND
Describes what the selected algorithm does in relation to the overall purpose of the program.</td><td>The algorithm being described can utilize existing language functionality, or library calls. Response earns the point even if the algorithm was not newly developed. (i.e., a student’s reimplementation of the algorithm to find the minimum value). Mathematical and logical concepts can be a part of the selected algorithm or part of either of the included algorithms.

Do NOT award a point if any one of the following is true:
● the selected algorithm consists of a single instruction;
● the selected algorithm consists solely of library calls to existing language functionality;
● the selected algorithm does not include mathematical or logical concepts;
● the response only describes what the selected algorithm does without explaining how it does it;
● the response does not explicitly address the program’s purpose;
● the code segment consisting of the selected algorithm is not included in the written responses section or is not explicitly identified in the program code section; or
● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).</td></tr>
<tr><td colspan="2">The response earned the point for this row.
The selected algorithms use logical concepts by including if statements. The response explains how the coin event algorithms functions. The response states, “Each time the coin is clicked, the variable treasureScore increase by 1. Once the score reaches 20, another variable gotCoins, increase by 1 and the screen switches to a win screen with a continue button.” The response describes how this algorithm does in relation to the overall purpose. It states, “Once this task (the coin event game) is complete, the user must play a second mini game in which he/she collects a scale. When both mini games are won, the Underwater Adventure level is complete.”</td></tr>
<tr><td>Row 6
Response 2C

Selected code segment implements an algorithm that</td><td>Responses are still eligible to earn this row, even if they do not earn row 5. The included algorithms can be sub-parts of the algorithm in row 5.

Do NOT award a point if any one of the following is true:
● the selected algorithm consists of a single instruction;</td></tr>
</table>

	includes at least two or more algorithms. **AND** At least one of the included algorithms uses mathematical or logical concepts. **AND** Explains how one of the included algorithms functions independently..	● the selected algorithm consists solely of library calls to existing language functionality; ● neither of the included algorithms nor the selected algorithm that includes two or more algorithms uses mathematical or logical concepts; ● the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or ● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).
	The response DID NOT earn the point for this row. While three algorithms are shown, there is not an algorithm that is using at least two or more algorithms. The two algorithms independently call the third. **Code.org Commentary:** To receive the point, at least three algorithms need to be defined and explained. The selected algorithm (main algorithm) needs to include or reference two other algorithms (defined as "included algorithms" in the Scoring Notes).	

2d. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a **rectangle** in **section 3** below). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. *(Must not exceed 200 words)*

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**

1.
``
function setUpBlastOff() {
mars = 0;
jupiter = 0;
hideText("alienTalk","okMars");
hideText("alienTalk", "okJupiter");
setText("saveMars", "Saved Mars? No");
setText("saveJupiter", "Saved Jupiter? No");
setPosition("spaceCharacter", 90, 285);
setCharacter("spaceCharacter");
showText("alienTalk", "playBlastOff");
showElement("blastOffInstructions");
setText("alienTalk", ("Hello "+username)+"! All of the planets in space are becoming uninhabitable! I guess I'll just have to invade Earth.");
setText("blastOffInstructions", "Prevent the alien from invading Earth by making the other planets in the galaxy safe for life. Click on the planets to explore them.");
}
``
2.
``
function hideText(textArea, button) {
hideElement(textArea);
hideElement(button);
}
``
3.
``
function showText(textArea, button) {
showElement(textArea);
showElement(button);
}
``
4.
``
function setCharacter(image) {
if (boyClick>=1) {
setImageURL(image, "https://www.carstickers.com/prodimages/1033_blue_sky_family_boy_stick_figure_sticker_decal.gif");
} else {
setImageURL(image, "https://www.carstickers.com/prodimages/1032_blue_sky_family_girl_stick_figure_sticker_decal.gif");
}
}
``
1. function setUpBlastOff() { mars = 0; jupiter = 0; hideText("alienTalk","okMars"); hideText("alienTalk", "okJupiter"); setText("saveMars", "Saved Mars? No"); setText("saveJupiter", "Saved Jupiter? No"); setPosition("spaceCharacter", 90, 285); setCharacter("spaceCharacter"); showText("alienTalk", "playBlastOff"); showElement("blastOffInstructions"); setText("alienTalk", ("Hello "+username)+"! All of the planets in space are becoming uninhabitable! I guess I'll just have to invade Earth."); setText("blastOffInstructions", "Prevent the alien from invading Earth by making the other planets in the galaxy safe for life. Click on the planets to explore them."); }
	Row 7
	Response 2D
	
	Selected code segment is a student-developed abstraction.
		Responses that use existing abstractions to create a new abstraction, such as creating a list to represent a collection (e.g., a classroom, an inventory), would earn this point.
		
		Do NOT award a point if any one of the following is true:
		● the response is an existing abstraction such as variables, existing control structures, event handlers, APIs;
		● the code segment consisting of the abstraction is not included in the written responses section or is not explicitly identified in the program code section; or
		● the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code

<table>
<tr><td rowspan="4">My code uses abstraction by incorporating a variety of functions to condense repeated code into smaller pieces. These functions can then be called within other functions or onEvents to make the code easier to read and understand. The function setUpBlastOff (1.) calls the functions hideText (2.), showText (3.), and setCharacter (4.), which are used to reduce the number of lines of code contained within the function. The setCharacter function (4.) is also an abstraction because it uses an if statement to check the value of the two character variables (boy and girl) and uses this logic to set the on-screen character to the one that the user chose. Overall, the function setUpBlastOff is an abstraction because it is called multiple times in the program to reset the 321BlastOff screen when the level is either won or quit. This prevents the programmer from having to retype the same several lines of code each time they want to reset the screen. These abstractions make my program more manageable because they take repeated sections of code that would add significant complexity to my algorithms and reduce them into single functions. For example, the setUpBlastOff function would require at least 12 lines of code each time it was called if it were not condensed into a function.</td><td></td><td>segment containing the abstraction).</td></tr>
<tr><td colspan="2">The response earned the point for this row.
The selected code segment includes a student-developed procedure setUpBlastOff. The additional procedures that are shown are used in this procedure.</td></tr>
<tr><td>Row 8
Response 2D

Explains how the selected abstraction manages the complexity of the program.</td><td>Responses should not be penalized for explanations of abstractions that are not developed by the student.

Do NOT award a point if any one of the following is true:
● the explanation does not apply to the selected abstraction; or
● the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction).</td></tr>
<tr><td colspan="2">The response earned the point for this row.
The abstraction being described is a procedure setUpBlastOff that calls three additional procedures. The response explains that the abstraction makes the program more manageable “because they take repeated sections of code that would add significant complexity to my algorithms and reduce them into single functions. For example, the setUpBlastOff function would require at least 12 lines of code each time it was called if it were not condensed into a function.”</td></tr>
</table>

3. Program Code

Capture and paste your entire program code in this section.

- ➢ Mark with an oval the segment of program code that implements the algorithm you created for your program that integrates other algorithms and integrates mathematical and/or logical concepts.
- ➢ Mark with a rectangle the segment of program code that represents an abstraction you developed.
- ➢ Include comments or acknowledgments for program code that has been written by someone else.

Create PT - Sample F 2018 - Score: 5/8

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6	Row 7	Row 8	*This document combines student sample, scoring guidelines and scoring commentary from: Create PT Sample F*
Sample: F	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	

Video
Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. **Your video must not exceed 1 minute in length and must not exceed 30MB in size.**

Program Purpose and Development
2a. Provide a written response or audio narration in your video that:

- identifies the programming language
- identifies the purpose of your program; and
- Explains what the video illustrates.

(Must not exceed 150 words)

Student Response	Scoring Guidelines	
	Row and Task	**Decision Rules**
 *My program is called "Nyan Cat Clicker". The programming language that was used was JavaScript in a program called AppLab. This program is a clicker game,*	**Row 1** **Response 2A** The video demonstrates the running of at least one feature of the program submitted. **AND** The response (audio narration or written response) identifies the purpose of the program (what the program is attempting to do).	Response earns the point if it explains the function of the program instead of identifying the purpose. Response earns the point if the illustrated feature runs, even if it does not function as intended. Response earns the point if the response is included in the video via narration or some form of closed captioning and addresses the purpose or function of the program. **Do NOT award a point if any one of the following is true:** ● a video is not submitted; ● the video does not illustrate the feature mentioned in the response; or ● the video does not illustrate the running of the feature (screen shots or storyboards are not acceptable and would not be credited).

in which the user clicks "Start" and it changes screen to the "PlayScreen". This is where a timer will start, the user has 30 seconds to click on the Nyan Cat multiple times, in which every click counts as a score of 1. The goal of this program is to click the Nyan Cat as many times as possible to get a higher score everytime in 30 seconds. Once 30 seconds are over, it will switch the screen to the "GameOverScreen". This screen shows that the game is over and how many times you clicked the Nyan Cat, along with the highest score the user received from all the games played. The app also has a restart button and pause button.	**The response earned the point for this row.** The video illustrates features of the program, and the response states the purpose of the program.

2b. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and/ or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development. *(Must not exceed 200 words)*

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**
There were several problems that were presented in the code. One problem was setting the timer to countdown 30 seconds and stopping the time when the pause button was clicked, then resuming the time again. This was an independent development. I had to find out how to resume the game, such that the time that was paused did not restart again after clicking the resume button. To solve this, I made a function called Time and called it in the event handler of the resume button, which in result would change just the time in intervals	**Row 2 - Response 2B** Describes or outlines steps used in the incremental and iterative development process to create the entire program.	**Do NOT award a point if any one of the following is true:** ● the response only includes the process for determining the program idea and does not address the development process used to create the entire program; or ● the response does not indicate iterative development; ● refinement and revision are not connected to feedback, testing, or reflection; or ● the response only describes the development at two specific points in time.
	The response DID NOT earn the point for this row. The response does not describe the incremental and iterative design process for the entire program.	

of 1000 milliseconds, but not move back to 30 seconds. Another problem, also an independent development, was updating the high score the user received every time the score was higher. The code would update the high score, but to the score the user received, regardless if it was higher or not. I decided to add in an if statement, such that if the score the user received was greater than the score from other games, it would update the high score value to that score.	**Code.org Commentary:** The response only focuses on two problems and their solutions. This does not meet the incremental (step by step) development process requirement. The response also does not explain if the problems were solved using an iterative development process related to feedback, testing, or reflection.	
	Row 3 - Response 2B Specifically identifies at least two program development difficulties or opportunities. **AND** Describes how the two identified difficulties or opportunities are resolved or incorporated.	Response earns the point if it identifies two opportunities, or two difficulties, or one opportunity and one difficulty AND describes how each is resolved or incorporated. **Do NOT award a point if any one of the following is true:** ● only one distinct difficulty or opportunity in the process is identified and described; or ● the response does not describe how the difficulties or opportunities were resolved or incorporated.
	The response earned the point for this row. The response identifies two difficulties and how they are resolved. The first difficulty is "setting the timer to countdown 30 seconds and stopping the time when the pause button was clicked." Its resolution is "a function called Time and called it in the event handler of the resume button, which in result would change just the time in intervals of 1000 milliseconds, but not move back to 30 seconds." The second difficulty is "updating the high score the user received every time the score was higher." Its resolution is "to add in an if statement, such that if the score the user received was greater than the score from other games, it would update the high score value to that score."	

2c. Capture and paste a program code segment that implements an algorithm (marked with an **oval** in **section 3** below) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. *(Must not exceed 200 words)*

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**

onEvent("btnPlay", "click", function(event) {
  setScreen("playScreen");
  Time();
});
onEvent("Catimage", "click", function(event) {
  setPosition("Catimage", randomNumber(0,200),
  randomNumber(0,395));
  getText("Score", Score);
  Score = Score + 1;
  setText("Score", Score);
});
onEvent("btnRestart", "click", function(event) {
	Row 4
	Response 2C
	Selected code segment implements an algorithm.
		Do NOT award a point if any one of the following is true:
		● the algorithm consists of a single instruction;
		● the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or
		● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code

```
    setScreen("playScreen");
    seconds = 30; Score = 0;
    setText("Time", seconds);
    setText("Score", Score);
    Time();
});
onEvent("btnPause", "click", function(event) {
    setScreen("PauseScreen");
    clearInterval(myInterval);
});
onEvent("btnResume", "click", function(event) {
    setScreen("playScreen");
    Time();
});
onEvent("btnRestart1", "click", function(event) {
    setScreen("playScreen");
    seconds = 30;
    Score = 0;
    setText("Time", seconds);
    setText("Score", Score);
    Time();
});
```

I used this algorithm in order to set the code in an organized way, such that it would work efficiently and when the a certain button is pressed, it would execute the code written for the button in the event handler. For example when the event handler "Catimage" was called, it would execute the code in that event handler when the Nyan Cat image was clicked, then adding adding 1 point to the score and setting the text of the score to update everytime the Nyan Cat was clicked such that the score would keep increasing. Something else to notice is the Time function being called in several event handlers. This was done in order to set the timer to restart or to start again depending on the event handler that is executed.

	segment containing the algorithm).
The response earned the point for this row. The code given in the response is an algorithm because it involves sequencing, selection and/or iteration. **Code.org Commentary:** This algorithm contains sequencing. We recommend also including selection (if statements) and/or iteration (for loops).	
Row 5 **Response 2C** Selected code segment implements an algorithm that uses mathematical or logical concepts. **AND** Explains how the selected algorithm functions. **AND** Describes what the selected algorithm does in relation to the overall purpose of the program.	The algorithm being described can utilize existing language functionality, or library calls. Response earns the point even if the algorithm was not newly developed. (i.e., a student's reimplementation of the algorithm to find the minimum value). Mathematical and logical concepts can be a part of the selected algorithm or part of either of the included algorithms. **Do NOT award a point if any one of the following is true:** ● the selected algorithm consists of a single instruction; ● the selected algorithm consists solely of library calls to existing language functionality; ● the selected algorithm does not include mathematical or logical concepts; ● the response only describes what the selected algorithm does without explaining how it does it; ● the response does not explicitly address the program's purpose; ● the code segment consisting of the selected algorithm is not included in the written responses section or is not explicitly identified in the program code section; or ● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).
The response DID NOT earn the point for this row. Although the algorithm given includes math minimally (incrementing the score) and it explains how the functions work, it does not describe what this algorithm does in relation to the overall program.	
Row 6 **Response 2C** Selected code segment implements an algorithm that includes at least two or more algorithms.	Responses are still eligible to earn this row, even if they do not earn row 5. The included algorithms can be sub-parts of the algorithm in row 5. **Do NOT award a point if any one of the following is true:** ● the selected algorithm consists of a single instruction; ● the selected algorithm consists solely of library calls to existing language functionality;

	AND At least one of the included algorithms uses mathematical or logical concepts. **AND** Explains how one of the included algorithms functions independently..	● neither of the included algorithms nor the selected algorithm that includes two or more algorithms uses mathematical or logical concepts; ● the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or ● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).
	The response DID NOT earn the point for this row. The response does not clearly identify two algorithms used by the selected algorithm. **Code.org Commentary:** To receive this point, the two included algorithms must be defined and explained in addition to the selected algorithm	

2d. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a **rectangle** in **section 3** below). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. *(Must not exceed 200 words)*

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**

function Time() {
myInterval = setInterval(function() {
seconds = seconds - 1;
setText("Time",seconds);
if (seconds === 0){
if (Score > HighScore) {
HighScore = Score;
setKeyValue("HighScore", Score, function () {
setText("HighScore", HighScore);
});
}
setText("txtScore","You have received a score of " + Score + " Nyan Cats that you clicked!!");
setScreen("GameOverScreen");
clearInterval(myInterval);
}
}, 1000);
}
 	
In order to use the , such as certain lines of code are not repeated, I made my
	Row 7
	Response 2D
	
	Selected code segment is a student-developed abstraction.
		Responses that use existing abstractions to create a new abstraction, such as creating a list to represent a collection (e.g., a classroom, an inventory), would earn this point.
		
		Do NOT award a point if any one of the following is true:
		● the response is an existing abstraction such as variables, existing control structures, event handlers, APIs;
		● the code segment consisting of the abstraction is not included in the written responses section or is not explicitly identified in the program code section; or
		● the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction).

	The response earned the point for this row. The code segment given represents an abstraction (a procedure or function).	

<table>
<tr><td rowspan="2">code use abstraction is by using this Time() function, and implementing it into the event handlers so the code is reduced and not being repeated on certain event handlers. My first thought for my Time() function was using the setInterval() function and the if statement and applying it to certain event handlers over again, such that the 30 second interval would run, and the screen would change when the 30 seconds would end. Using abstraction helped me implement the setInterval() function and the if statement such that it would be unnecessary to repeat code and have extra lines of code, adding less complexity to the algorithms that I used into making this code.</td><td>Row 8
Response 2D

Explains how the selected abstraction manages the complexity of the program.</td><td>Responses should not be penalized for explanations of abstractions that are not developed by the student.

Do NOT award a point if any one of the following is true:
● the explanation does not apply to the selected abstraction; or
● the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction).</td></tr>
<tr><td colspan="2">The response earned the point for this row.
The response explains how the abstraction manages complexity for the program by stating: "the code is reduced and not being repeated on certain event handlers."</td></tr>
</table>

3. Program Code

Capture and paste your entire program code in this section.

- ➢ Mark with an oval the segment of program code that implements the algorithm you created for your program that integrates other algorithms and integrates mathematical and/or logical concepts.
- ➢ Mark with a rectangle the segment of program code that represents an abstraction you developed.
- ➢ Include comments or acknowledgments for program code that has been written by someone else.

Create PT - Sample I 2018 - Score: 3/8

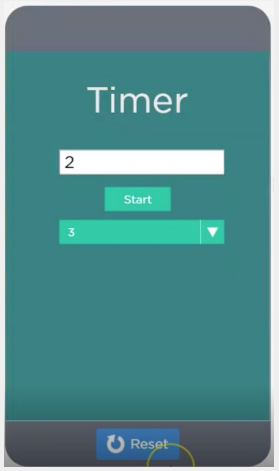
Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6	Row 7	Row 8	*This document combines student sample, scoring guidelines and scoring commentary from: Create PT Sample I*
Sample: F	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	

Video
Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. **Your video must not exceed 1 minute in length and must not exceed 30MB in size.**

Program Purpose and Development
2a. Provide a written response or audio narration in your video that:

- identifies the programming language
- identifies the purpose of your program; and
- Explains what the video illustrates.

(Must not exceed 150 words)

Student Response	Scoring Guidelines	
	Row and Task	**Decision Rules**
 *Hello, today I will be showcasing to you my app and it's purpose is to assimilate a timer. I used App Lab section by Code.org to build my app. The timer was*	**Row 1** **Response 2A** The video demonstrates the running of at least one feature of the program submitted. **AND** The response (audio narration or written response) identifies the purpose of the program (what the program is attempting to do).	Response earns the point if it explains the function of the program instead of identifying the purpose. Response earns the point if the illustrated feature runs, even if it does not function as intended. Response earns the point if the response is included in the video via narration or some form of closed captioning and addresses the purpose or function of the program. **Do NOT award a point if any one of the following is true:** ● a video is not submitted; ● the video does not illustrate the feature mentioned in the response; or ● the video does not illustrate the running of the feature (screen shots or storyboards are not acceptable and would not be credited).

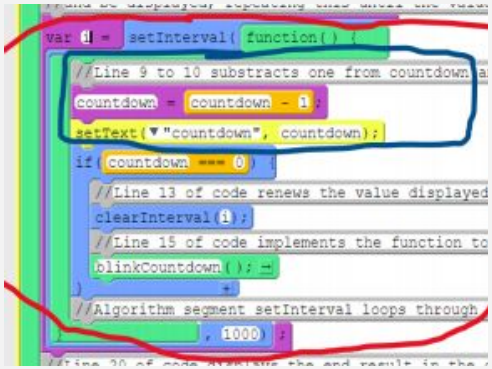
built in a pseudo language that only operates in the App Lab environment. It starts by displaying a blank countdown slot in which an algorithm will take the input value by the user and subtract one from it and display the results. It will repeat such proses until the value displayed is zero. After the fact, a different algorithm will make the timer count internally three seconds and the countdown slot will begin to blink. It will disappear one second and come back for one second, it will repeat this process until the app is restarted completely.	**The response earned the point for this row.** The video illustrates the program running, and the response states the purpose of the program as shown in the video.

2b. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and/ or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development. *(Must not exceed 200 words)*

Student Response	Scoring Guidelines	
	Row and Task	**Decision Rules**
Whilst developing the code for the program, two main problems came about. One main problem was to learn how to get the app to display in the countdown slot in the screen the values chosen by the user preliminary set by me on the dropdown text. This was an independent development. At first, my idea was to implement many numbers from where to choose from, but that would had resulted in an exponentially longer code to map to the screen and only giving the option of one-digit values. Therefore, I decided to implement a dropdown textbox with predetermined values as to now only map the value of it to the screen, resulting in a successful	**Row 2 - Response 2B** Describes or outlines steps used in the incremental and iterative development process to create the entire program.	**Do NOT award a point if any one of the following is true:** ● the response only includes the process for determining the program idea and does not address the development process used to create the entire program; or ● the response does not indicate iterative development; ● refinement and revision are not connected to feedback, testing, or reflection; or ● the response only describes the development at two specific points in time.
	The response DID NOT earn the point for this row. The response does not describe the overall interactive or iterative development of the entire program. **Code.org Commentary:** The response only focuses on two problems and how they were solved instead of explaining the process of creating the program step by step (incremental development) and how parts were improved through testing, reflection, or feedback (iterative development).	

simplification of my code. Second problem was the most essential to fix, how to subtract one progressively. This was also resolved independently. For this I implemented a function that would repeat itself every second, which would take countdown and subtract one from its value until it reached zero. But to keep it from going into negative numbers, I implemented a function which would reset the value of countdown and set it to zero. Which took less functions and is more cleaner than implementing a for-loop.	**Row 3 - Response 2B** Specifically identifies at least two program development difficulties or opportunities. **AND** Describes how the two identified difficulties or opportunities are resolved or incorporated.	Response earns the point if it identifies two opportunities, or two difficulties, or one opportunity and one difficulty AND describes how each is resolved or incorporated. **Do NOT award a point if any one of the following is true:** ● only one distinct difficulty or opportunity in the process is identified and described; or ● the response does not describe how the difficulties or opportunities were resolved or incorporated.
	The response DID NOT earn the point for this row. The response states that "One main problem was to learn how to get the app to display in the countdown slot..." However, this is not a problem or opportunity during the program development process, it's a design issue. The second difficulty and its resolution are given.	

2c. Capture and paste a program code segment that implements an algorithm (marked with an **oval** in **section 3** below) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. *(Must not exceed 200 words)*

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**
 The red oval in the image captures the algorithm which is essential for the function of the program. Within it there is an algorithm in lines 9-10, which takes the value given to countdown by the	**Row 4** **Response 2C** Selected code segment implements an algorithm.	**Do NOT award a point if any one of the following is true:** ● the algorithm consists of a single instruction; ● the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or ● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).
	The response earned the point for this row. The selected block of code represents an algorithm. **Code.org Commentary:** The block of code in this response includes both sequencing and selection thus qualifying as an algorithm.	

<table>
<tr>
<td rowspan="4">user from the dropdown textbox and subtracts one from it and then displays the result on the countdown slot. This algorithm was developed independently. The second algorithm begins in code line 7 as setInterval, and its value is displayed in line 18 of the code as 1000 milliseconds. The value of the setInterval was also developed independently. Both algorithms are essential because in unison they allow the program to work, as without the first algorithm the code would not work as nothing would be displayed nor nothing would be subtracted so the function would eventually reach zero. While without the second algorithm result would always be one less then the value set for countdown and there be no command telling it to repeat the process in algorithm number one.</td>
<td colspan="2">

```
var i = setInterval(function() {
   countdown = countdown - 1;
   setText("countdown", countdown);
   if(countdown === 0){
      clearInterval(i);
      blinkCountdown();
   }
}
```

</td>
</tr>
<tr>
<td>

Row 5
Response 2C

Selected code segment implements an algorithm that uses mathematical or logical concepts.

AND

Explains how the selected algorithm functions.

AND

Describes what the selected algorithm does in relation to the overall purpose of the program.

</td>
<td>

The algorithm being described can utilize existing language functionality, or library calls. Response earns the point even if the algorithm was not newly developed. (i.e., a student's reimplementation of the algorithm to find the minimum value). Mathematical and logical concepts can be a part of the selected algorithm or part of either of the included algorithms.

Do NOT award a point if any one of the following is true:

- the selected algorithm consists of a single instruction;
- the selected algorithm consists solely of library calls to existing language functionality;
- the selected algorithm does not include mathematical or logical concepts;
- the response only describes what the selected algorithm does without explaining how it does it;
- the response does not explicitly address the program's purpose;
- the code segment consisting of the selected algorithm is not included in the written responses section or is not explicitly identified in the program code section; or
- the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).

</td>
</tr>
<tr>
<td colspan="2">

The response earned the point for this row.
The circled code contains math/logic. The response explains how the algorithm works in detail. The response describes what this algorithm does in relation to the program: "Both algorithms are essential because in unison they allow the program to work" AND then states what would happen to the program if either part is missing. By explaining what happens if each part is missing, the response is explaining what these algorithms do for the whole program.

Code.org Commentary: This code segment has an if-statement, which is a logical concept.

</td>
</tr>
<tr>
<td>

Row 6
Response 2C

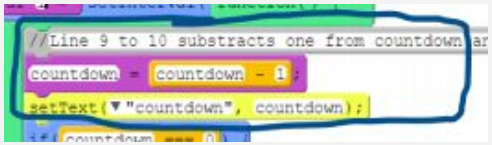
</td>
<td>

Responses are still eligible to earn this row, even if they do not earn row 5. The included algorithms can be sub-parts of the algorithm in row 5.

</td>
</tr>
</table>

	Selected code segment implements an algorithm that includes at least two or more algorithms. **AND** At least one of the included algorithms uses mathematical or logical concepts. **AND** Explains how one of the included algorithms functions independently..	**Do NOT award a point if any one of the following is true:** ● the selected algorithm consists of a single instruction; ● the selected algorithm consists solely of library calls to existing language functionality; ● neither of the included algorithms nor the selected algorithm that includes two or more algorithms uses mathematical or logical concepts; ● the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or ● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).
	The response DID NOT earn the point for this row. The response identifies two algorithms, one of which is inside the other. There is no additional algorithm identified. **Code.org Commentary:** To receive this point, the **selected** algorithm must be clearly defined along with two **included** algorithms that can function independently. There must be at least three distinct algorithms defined and explained within the response.	

2d. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a **rectangle** in **section 3** below). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. *(Must not exceed 200 words)*

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**
 *The blue rectangle in the image showcases a code segment which implements a mathematical function to the program which was developed independently. The code takes the value given to countdown by the user in the dropdown textbox or code getText from*	**Row 7** **Response 2D** Selected code segment is a student-developed abstraction.	Responses that use existing abstractions to create a new abstraction, such as creating a list to represent a collection (e.g., a classroom, an inventory), would earn this point. **Do NOT award a point if any one of the following is true:** ● the response is an existing abstraction such as variables, existing control structures, event handlers, APIs; ● the code segment consisting of the abstraction is not included in the written responses section or is not explicitly identified in the program code section; or

line 4 and subtracts one from it. After so, the code segment displays the result in the countdown slot on the screen. This code segment is part of a function which serves as an algorithm to repeat the code segment. Without the code segment the hole program would not run as there would be no segment giving it the means to subtract and reach zero, therefore the displayed value would always be the one input by the user.		● the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction).
	The response DID NOT earn the point for this row. The boxed code fragment does not represent an abstraction. **Code.org Commentary:** To receive this point, students need to select an abstraction within their program, such as a function or a list. This abstraction is the code segment that is copy/pasted into Response 2d.	
	Row 8 **Response 2D** Explains how the selected abstraction manages the complexity of the program.	Responses should not be penalized for explanations of abstractions that are not developed by the student. **Do NOT award a point if any one of the following is true:** ● the explanation does not apply to the selected abstraction; or ● the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction).
	The response DID NOT earn the point for this row. The response does not explain how the boxed code manages complexity in the program. The response merely states how the code fragment works.	

3. Program Code

Capture and paste your entire program code in this section.

- ➢ Mark with an oval the segment of program code that implements the algorithm you created for your program that integrates other algorithms and integrates mathematical and/or logical concepts.
- ➢ Mark with a rectangle the segment of program code that represents an abstraction you developed.
- ➢ Include comments or acknowledgments for program code that has been written by someone else.

Create PT - Sample J 2018 - Score: 1/8

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6	Row 7	Row 8	*This document combines student sample, scoring guidelines and scoring commentary from: Create PT Sample J*
Sample: F	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	

Video
Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. **Your video must not exceed 1 minute in length and must not exceed 30MB in size.**

Program Purpose and Development
2a. Provide a written response or audio narration in your video that:

- identifies the programming language
- identifies the purpose of your program; and
- Explains what the video illustrates.

(Must not exceed 150 words)

Student Response	Scoring Guidelines	
	Row and Task	**Decision Rules**
 *The computing innovation that is represented by the computational artifact is an app. The app was based on sports trivia.The purpose of this computational artifact, was to make the player get*	**Row 1** **Response 2A** The video demonstrates the running of at least one feature of the program submitted. **AND** The response (audio narration or written response) identifies the purpose of the program (what the program is attempting to do).	Response earns the point if it explains the function of the program instead of identifying the purpose. Response earns the point if the illustrated feature runs, even if it does not function as intended. Response earns the point if the response is included in the video via narration or some form of closed captioning and addresses the purpose or function of the program. **Do NOT award a point if any one of the following is true:** ● a video is not submitted; ● the video does not illustrate the feature mentioned in the response; or ● the video does not illustrate the running of the feature (screen shots or storyboards are not acceptable and would not be credited).

educated off of sports and the varieties of different teams and players.	**The response earned the point for this row.** The video illustrates features of the program and the response states the purpose of the program.

2b. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and/ or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development. *(Must not exceed 200 words)*

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**
In our app " Sports Trivia", it included player's faces and team logos for the participant to guess from right to wrong. We have a title screen which shows which sport you want to be questioned on. The game consists of 2 questions for each sport that has to be answered. So basically whatever sport you want you can choose it from the welcome screen and it'll bring you to the desired sport you want to be questioned on. It was hard deciding which particular questions my partner and i would come up with because we would always have debates on which question is the hardest or easiest. Also, it was hard figuring what genre our game would be based on, there are plenty of different ideas for a trivia game. Developing the game was a hardship, figuring out our code and transitions for each screen was always a debate, there was a lot of testing and bugs along the way but we came to a final conclusion for all these tough situations.	**Row 2 - Response 2B** Describes or outlines steps used in the incremental and iterative development process to create the entire program.	**Do NOT award a point if any one of the following is true:** ● the response only includes the process for determining the program idea and does not address the development process used to create the entire program; or ● the response does not indicate iterative development; ● refinement and revision are not connected to feedback, testing, or reflection; or ● the response only describes the development at two specific points in time.
	The response DID NOT earn the point for this row. The response does not describe the incremental and iterative development of the entire program. **Code.org Commentary:** The response details how the game works and design challenges faced instead of explaining the process of creating the program for the app step by step (incremental development) and how parts were improved through testing, reflection, or feedback (iterative development).	
	Row 3 - Response 2B Specifically identifies at least two program development difficulties or opportunities. **AND** Describes how the two identified difficulties or opportunities are resolved or incorporated.	Response earns the point if it identifies two opportunities, or two difficulties, or one opportunity and one difficulty AND describes how each is resolved or incorporated. **Do NOT award a point if any one of the following is true:** ● only one distinct difficulty or opportunity in the process is identified and described; or ● the response does not describe how the difficulties or opportunities were resolved or incorporated.
	The response DID NOT earn the point for this row.	

	The response does not identify two difficulties and/or opportunities related to the program development process and how these programming issues were resolved or handled. The issues identified have to do with "what particular questions my partner and I would come up with" and "what genre our game would be based on." **Code.org Commentary:** To receive the point, two distinct opportunities or problems must be defined and an explanation given for how they were solved using feedback, testing, or revision. These two problems must be related to program development (i.e. writing the code) and NOT design development (i.e. deciding on a screen layout).

2c. Capture and paste a program code segment that implements an algorithm (marked with an **oval** in **section 3** below) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. *(Must not exceed 200 words)*

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**

onEvent("WelcomeScreen", "click", function(event) {
console.log("WelcomeScreen clicked!");
});
onEvent("football", "click", function(event) {
console.log("football clicked!");
});
onEvent("football", "click", function(event) {
setScreen("Footballq1");
});
onEvent("Footballq1", "click", function(event) {
console.log("Footballq1 clicked!");
});
onEvent("Footballq2", "click", function(event) {
console.log("Footballq2 clicked!");
});
onEvent("abbutton", "click", function(event) {
setScreen("Footballq2");
});
onEvent("Gobackbutton", "click", function(event) {
console.log("Gobackbutton clicked!");
});
onEvent("Gobackbutton", "click", function(event) {
setScreen("WelcomeScreen");
});
onEvent("Baseballq1", "click", function(event) {
console.log("Baseballq1 clicked!");
});
onEvent("baseball", "click", function(event) {
setScreen("Baseballq1");
});
	Row 4
	Response 2C
	Selected code segment implements an algorithm.
		Do NOT award a point if any one of the following is true:
		● the algorithm consists of a single instruction;
		● the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or
		● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).

	The response DID NOT earn the point for this row. The code given in the response is not considered an algorithm because each event handler consists of a single instruction that does not involve sequencing, selection, or iteration. **Code.org Commentary:** In this case, the AP Scorer is looking to see if an onEvent can count as an algorithm. A collection of onEvents does not count as a single algorithm. Each individual onEvent only contains a single instruction which disqualifies it as an algorithm. To receive the point, the algorithm must contain sequencing (more than one step), selection (if-statement), or iteration (for loop).	
	Row 5 **Response 2C** Selected code segment	The algorithm being described can utilize existing language functionality, or library calls. Response earns the point even if the algorithm was not newly developed. (i.e., a student's reimplementation of the algorithm to find the minimum value). Mathematical and logical concepts can be a

<table>
<tr>
<td rowspan="4">

```
onEvent("gobackbutton2", "click", function(event) {
    console.log("gobackbutton2 clicked!");
});
onEvent("gobackbutton2", "click", function(event) {
    setScreen("WelcomeScreen");
});
onEvent("nybutton", "click", function(event) {
    setScreen("Baseballq2");
});
onEvent("basketball", "click", function(event) {
    setScreen("Basketballq1");
});
onEvent("button11", "click", function(event) {
    setScreen("Basketballq2");
});
onEvent("button17", "click", function(event) {
    setScreen("WelcomeScreen");
});
onEvent("Soccer", "click", function(event) {
    setScreen("Soccerq1");
});
onEvent("button5", "click", function(event) {
    setScreen("Soccerq2");
});
onEvent("button20", "click", function(event) {
    setScreen("WelcomeScreen");
});
```

Within the code, the algorithm is essential to the program by each sport having it's own command or action. it controls which sport you want to go to. The code represents the amount of time it takes you to answer a question, taking too long might make the game go back to the welcome screen so there's limited time to play. Picking the right answer will get you to the next question automatically. From there it's just repetitive for the rest of game, for all sports. All buttons are on click actions, as well, just know whatever sport you pick you must have a little background knowledge on it.
</td>
<td>implements an algorithm that uses mathematical or logical concepts.
AND
Explains how the selected algorithm functions.
AND
Describes what the selected algorithm does in relation to the overall purpose of the program.</td>
<td>part of the selected algorithm or part of either of the included algorithms.

Do NOT award a point if any one of the following is true:
● the selected algorithm consists of a single instruction;
● the selected algorithm consists solely of library calls to existing language functionality;
● the selected algorithm does not include mathematical or logical concepts;
● the response only describes what the selected algorithm does without explaining how it does it;
● the response does not explicitly address the program's purpose;
● the code segment consisting of the selected algorithm is not included in the written responses section or is not explicitly identified in the program code section; or
● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).</td>
</tr>
<tr>
<td colspan="2">**The response DID NOT earn the point for this row.**
The provided code does not use math or logic, and the code segment is not considered an algorithm.</td>
</tr>
<tr>
<td>**Row 6**
Response 2C

Selected code segment implements an algorithm that includes at least two or more algorithms.
AND
At least one of the included algorithms uses mathematical or logical concepts.
AND
Explains how one of the included algorithms functions independently..</td>
<td>Responses are still eligible to earn this row, even if they do not earn row 5. The included algorithms can be sub-parts of the algorithm in row 5.

Do NOT award a point if any one of the following is true:
● the selected algorithm consists of a single instruction;
● the selected algorithm consists solely of library calls to existing language functionality;
● neither of the included algorithms nor the selected algorithm that includes two or more algorithms uses mathematical or logical concepts;
● the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or
● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).</td>
</tr>
<tr>
<td colspan="2">**The response DID NOT earn the point for this row.**
The response does not clearly identify two additional algorithms that are used by the selected algorithm. This cannot be done because the selected code is not considered an algorithm.</td>
</tr>
</table>

2d. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a **rectangle** in **section 3** below). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. *(Must not exceed 200 words)*

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="5">


This manages the app so you get where you want to be questioned on. Allowing you to go to the home screen, after you finish your question. The game has a go back button which allows you to go home after you finish. But it was complex making the game move automatically to the next question after you answer it correctly. This was great experience making this computational artifact.
</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>**Row 7**
Response 2D

Selected code segment is a student-developed abstraction.</td><td>Responses that use existing abstractions to create a new abstraction, such as creating a list to represent a collection (e.g., a classroom, an inventory), would earn this point.

Do NOT award a point if any one of the following is true:
● the response is an existing abstraction such as variables, existing control structures, event handlers, APIs;
● the code segment consisting of the abstraction is not included in the written responses section or is not explicitly identified in the program code section; or
● the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction).</td></tr>
<tr><td colspan="2">**The response DID NOT earn the point for this row.**
The code segment given does not represent a student-defined abstraction. An event handler is built-in to this language.</td></tr>
<tr><td>**Row 8**
Response 2D

Explains how the selected abstraction manages the complexity of the program.</td><td>Responses should not be penalized for explanations of abstractions that are not developed by the student.

Do NOT award a point if any one of the following is true:
● the explanation does not apply to the selected abstraction; or
● the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction).</td></tr>
<tr><td colspan="2">**The response DID NOT earn the point for this row.**
The response does not explain how this abstraction chosen manages complexity of the program. The response states "It was complex making the game move automatically to the next question after you answer it correctly." This is not the correct meaning of complexity for this row.

Code.org Commentary: The response demonstrates a misunderstanding of the word “complexity” in the</td></tr>
</table>

	context of abstraction. Students need to explain how their abstraction manages or reduces the complexity of their program.

3. Program Code

Capture and paste your entire program code in this section.

- ➢ Mark with an oval the segment of program code that implements the algorithm you created for your program that integrates other algorithms and integrates mathematical and/or logical concepts.
- ➢ Mark with a rectangle the segment of program code that represents an abstraction you developed.
- ➢ Include comments or acknowledgments for program code that has been written by someone else.

Unit 7 Lesson 2

Create PT - Make a Plan

Resources

Create PT Survival Guide 2019/2020[1]

Create PT Overview

Goal of the Task: Create a programming project of your own design and then explain the **purpose, process, algorithms and abstractions** used to build it. You have **12 hours** to complete this task.

What you Submit: (1) Video of running program (2) Written Responses to prompts 2a-d (3) PDF of program code

How you get a good score: The AP committee wants to see that you can:

- Design and write the code for a computer program with a topic of your choosing
- Describe how you identified and solved problems as you developed your program
- Write code for an algorithm and describe its purpose within your program
- Write code for an abstraction and describe its purpose within your program

Suggested Process in a Nutshell (see also the Sample Timeline on following pages)**:**

Pick your project… Something small enough that you can design and write in only a few hours. You should basically know ahead of time what the core algorithm will be.

Hours 1-2 **Develop a "good enough" program / prototype within 2 hours**

- ❏ Evaluate whether you can finish what you set out to do, and adjust as necessary.
- ❏ Check progress for responses 2c and 2d - algorithms and abstraction.
- ❏ You should **know** what **your algorithm and abstraction** will be at this point.

Hours 3-7 **Keep coding and get to a stopping point with ~5 hours to go**

- ❏ Your video and written responses will take time to make — you'll want to make last minute improvements!

Hour 8 **Record your video and respond to 2a**

Hours 9-10 **Write responses to 2b, 2c, 2d**

Hours 11-12 **Prepare to submit**

- ❏ Finalize program code and written responses and submit on the digital portfolio as three separate files.
  - ❏ Video of running program (.mp4, .wmv, .avi, or .mov)
  - ❏ Written Response to prompts 2a-2d (PDF)
  - ❏ Program Code (PDF)

[1] *Much of the content of this this guide was inspired by Jill Westerlund at the Abstracting CS blog. We are grateful for Jill's ingenuity and generosity.*

Algorithms on the Create PT (20 mins)

Is it a Good Algorithm?

Algorithm - College Board Definitions: Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages. Algorithms make use of sequencing, selection or iteration. People write programs to execute algorithms. Your algorithm must include two or more algorithms that work in combination to achieve some desired result. Finally it must integrate mathematical and/or logical concepts.

What it means: Algorithms are chunks of code that accomplish a task. To make sure your algorithm is a little complex and that you demonstrate your programming abilities the College Board has a few specific requirements.

- **Something You Wrote:** The entirety of the code you submit as your algorithm should be something that you wrote entirely on your own. You cannot submit code that a partner helped you write.

- **Mathematical and/or Logical Concepts:** Mathematical concepts means code where your app does some sort of mathematical computation (+, -, *, /, %). These are typically used when your program is making a calculation.

  Logical concepts means code where you use logical or comparison operators (&&, ||, !, ==, !=, <, > <= >=). These are typically used in combination with if-statements where your program is making a decision.

- **A Parent and Two Children:** Your **selected** algorithm must have two **included** algorithms that work in combination to achieve some result. The best way to think about this is that you have a “parent” algorithm that requires two or more components that can stand on their own as independent “child” algorithms.

  You should NOT submit three separate algorithms. Instead you should find one large task in your program that can actually be divided into two or more sub-tasks (for example: a large task might be making a user login screen; subtasks are (1) checking their login information (2) updating the screen appropriately). The code to accomplish each sub-task are the **included** “child” algorithms that can work independently to solve each sub-task, but are combined to work together as the **selected** “parent” algorithm to solve the overall task.

- **Talking About Your Algorithm:** It is recommended that you place your **selected** algorithm and **included** algorithms in separate functions. This will make it easier for you to talk about your algorithms in your responses. You may also, however, simply place rectangles around them and then use line numbers to identify your algorithms.

Does it Count? - Algorithm Edition

The AP reader has to judge strictly from the code you include in your response to 2c whether it meets the requirements. They will assume that all of the screen elements and variables the code refers to exist, and that the code is the working code from your video.

Row 6 - Response 2C: The points for Row 6 of the scoring guidelines are awarded strictly for the code segment selected.

Criteria	Decision Rules	Scoring Notes
Selected code segment implements an algorithm that **includes** at least two or more algorithms. **AND** At least one of the **included** algorithms uses mathematical or logical concepts. **AND** Explains how one of the **included** algorithms functions independently.	Responses are still eligible to earn this row, even if they do not earn row 5. The **included** algorithms can be sub-parts of the algorithm in row 5. **Do NOT award a point if any one of the following is true:** • the **selected** algorithm consists of a single instruction; • the **selected** algorithm consists solely of library calls to existing language functionality; • neither of the **included** algorithms nor the **selected** algorithm that includes two or more algorithms uses mathematical or logical concepts; • the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or • the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).	• Algorithms make use of sequencing, selection or iteration. • Mathematical concepts include mathematical expressions using arithmetic operators and mathematical functions. • Logical concepts include Boolean algebra and compound expressions. • Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times. • Selection uses a Boolean condition to determine which of two parts of an algorithm is used.

You be the AP Reader! You are are the AP reader trying to determine if they get the point for Row 6. Assume each algorithm snippet below was submitted as part of written response 2c. For each, select yes or no - should the point be awarded, and explain why. (It doesn't have to be a written explanation. You can draw an arrow or circle something in the code with a brief word or label about why.)

Example Algorithm 1
`onEvent("button1", "click", function() {`
`  setProperty("screen1","background-color","blue");`
`});`
`onEvent("id", "click", function(event) {`
`  setProperty("screen1","background-color","red");`
`});`
	Earn Point? Yes / No
	
	Why?

Example Algorithm 2
`onEvent("button1", "click", function() {`
`  setProperty("label1","background-color","blue");`
`  setProperty("label1","font-size",randomNumber(20,40));`
`  playSound("sound://default.mp3",false);`
`});`
	Earn Point? Yes / No
	
	Why?

Example Algorithm 3
`function manageLogIn() {`
`  if (loginStatus=="loggedIn") {`
`  showHomeScreen();`
`  } else {`
`  resetLogIn();`
`  }`
`}`
`function showHomeScreen() {`
`  setScreen("homeScreen");`
`  setText("id", "text");`
`}`
`function resetLogIn() {`
`  loginStatus = "loggedOut";`
`  setScreen("signInScreen");`
`}`
	Earn Point? Yes / No
	
	Why?

Example Algorithm 4

```
function increaseScore(){
  score = score + 1;
  checkIfEndGame();
  if(endGame){
    setText("timeText", "Time:" + totalTime + " secs!")
  } else {
    updateScreen();
  }
}

function updateScreen(){
  setPosition("button1",randomNumber(0,320), 100);
  setText("scoreText", score);
  playSound("sound://default.mp3", false);
}

function checkIfEndGame(){
  var endTime = getTime();
  totalTime = (startTime - endTime)/1000;
  endGame = false;
  if(score > 10 || totalTime > 20){
      endGame=true;
  }
}
```

Earn Point? Yes / No

Why?

Example Algorithm 5

```
while (done == false){
  var n = promptNum("Enter the secret number");

  if(n == 97){
    done = true;
    console.log("You did it! Begin countdown!");
    for(var i=97; i>=0; i--){
      console.log(i);
    }
    console.log("You're in!");
  } else {
    console.log("You've guessed incorrectly);
    if (n < 97){
      console.log("Guess higher");
    else {
      console.log("Guess lower");
    }
  }
}
```

Earn Point? Yes / No

Why?

Abstraction on the Create PT (20 mins)

Is it a Good Abstraction?

Abstraction - College Board Definitions The AP course framework includes the following statements related to abstraction and programming that are relevant for the Create PT:

- *Multiple levels of abstraction are used to write programs (EU 2.2)*
- *The process of developing an abstraction involves removing detail and generalizing functionality. (EK 2.2.1A)*
- *(Student can) Use abstraction to manage complexity in programs. (LO 5.3.1 [P3])*

What it means: Abstractions manage complexity. Programming abstractions make it easier to write complex programs. The AP reader is checking that you can **create, identify, and describe an abstraction** that manages complexity in your code.

- **Something You Wrote:** The code you submit as your abstraction should be something that you wrote entirely on your own. You cannot submit code that a partner helped you write.

- **Good Abstraction Choice - Functions that you wrote:** A function that you wrote means a function that you defined, named, and wrote code for (i.e code that starts `function myFunc(){ ... })`. Your function can demonstrate *managing complexity* if it either (or both):
  - gets called from multiple different places in your code
  - has a parameter that generalizes some behavior (and also probably called from several places)

- **Bad Abstraction Choice: `onEvent` and other built-in programming language/environment features**
  - `onEvent` does NOT count as a "student-developed abstraction". OnEvent is provided by the programming environment and the code contained within it is used in a single location in your code. Therefore it does not in any way manage complexity.
  - Identifying `onEvent` as an abstraction is the most common way to **lose** credit on this question.
  - **Variables** by themselves are not a data abstraction
  - Anything that is an existing abstraction provided by the language that you are simply using. For example: loops and logical structures are not student developed abstractions

*`onEvent` **IS NOT** an abstraction*

Does it Count? - Abstraction Edition

The AP reader has to judge strictly from the code you include in your response to 2d whether it meets the requirements. They will assume that all of the screen elements and variables the code refers to exist, and that the code is the working code from your video.

Row 7 - Response 2D: The points for Row 7 of the scoring guidelines are awarded strictly for the code segment selected.

Criteria	Decision Rules	Scoring Notes

• Selected code segment is a student-developed abstraction.	Responses that use existing abstractions to create a new abstraction, such as creating a list to represent a collection (e.g., a classroom, an inventory), would earn this point.
	
	Do NOT award a point if any one of the following is true:
	• the response is an **existing** abstraction such as variables, existing control structures, event handlers, APIs;
	• the code segment consisting of the abstraction is not included in the written responses section or is not explicitly identified in the program code section; or
	• the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly
		• The following are examples of abstractions (EK 5.3.1):
		  ○ Procedures
		  ○ Parameters
		  ○ Lists
		  ○ Application program interfaces (APIs)
		• Libraries
		• Lists and other collections can be treated as abstract data types (ADTs) in developing programs. (EK 5.5.1l)

	identifying the code segment containing the abstraction).	

Evaluate Abstractions! Each of the code segments below shows a portion of code with a rectangle around it. Sometimes additional code is included to help you understand the context of that abstraction (for example, to determine whether the abstraction helps manage complexity).

For each example below respond to 3 things:

- **Earn Point? Yes / No** - Would the selected code segment earn the point for row 7?
- **Why?** - note why it does or doesn't earn the point
- **Manages Complexity? Yes / No** - based on what you can see, are you able to argue that the abstraction manages complexity?

Example 1
`onEvent("btn1", "click", function(event) {`
`  setText("btn1", "I'm touched.");`
`  setProperty("btn1", "font-size", 100);`
`  showElement("lbl1");`
`});`
	Earn Point? Yes / No
	
	Why?
	
	Manages Complexity? Yes / No

Example 2
`1 onEvent("btn1", "click", function(event) {`
`2  if(score < 0){`
`3  setScreen("gameOverScreen")`
`4  }`
`5 });`
	Earn Point? Yes / No
	
	Why?
	
	Manages Complexity? Yes / No

Example 3	**Earn Point? Yes / No** **Why?** **Manages Complexity? Yes / No**

```
onEvent("btn1", "click", function(event) {
  updatePlayerTurn();
  updateScore(-5);

  if(score < 0){
    setScreen("gameOverScreen")
  }
  else{
    updateBoard();
  }
});
```

Example 4

```
onEvent("btn1", "click", function(event) {
  setButtonProps();
});

function setButtonProps(){
  setText("btn1", "I'm touched.");
  setProperty("btn1", "font-size", 100);
  showElement("lbl1");
}
```

Earn Point? Yes / No

Why?

Manages Complexity? Yes / No

Example 5

```
onEvent("btn1", "click", function(event) {
  setButtonProps();
});

onEvent("screen1", "click", function(event) {
  setButtonProps();
});

function setButtonProps(){
  setText("btn1", "I'm touched.");
  setProperty("btn1", "font-size", 100);
  showElement("lbl1");
}
```

Earn Point? Yes / No

Why?

Manages Complexity? Yes / No

Example 6

Earn Point? Yes / No

Why?

```
onEvent("btn1", "click", function(event) {
  updateScore(5);
});

onEvent("screen1", "click", function(event) {
  updateScore(-3);
});

function updateScore(amt){
  totalScore += amt;
  if( totalScore < 0 ){
     setScreen("gameOverScreen");
  }
}
```

Manages Complexity? **Yes / No**

“Narrow it Down” (15 mins)

You should assume that you’re not going to have enough time to complete the “perfect” project for the Create PT. **This is okay because you can get full credit for a programming project that feels incomplete as long as it has working elements.** While a large or more complete project is satisfying, your score is based mostly on the written responses, which is how you demonstrate that you understand the concepts covered on the Create PT.

All your project actually needs to include is:

- One working feature that you can demonstrate in your video
- An algorithm
- An abstraction

You will make the project much easier if you narrow down your original idea into a simpler target project. This will give you more time to complete the written responses or make smaller improvements.

How to Narrow it Down: Narrowing it down means identifying the sub-tasks or sub-problems that meet the PT requirements. Here’s a few strategies to help you do that:

- **Get to the Algorithm:** Split your project into individual components that will likely require an algorithm that meets the Create PT requirements. Each of these components individually could be your entire project.

- **Pick One Part of a Bigger Idea:** Think about your project as a set of programming tasks that each solve part of a larger problem. Some of these individual tasks might suffice for the Create PT. You can just pick one part of a big idea that meets the requirements that you think you can program in the time allotted.

- **Minimal Design Mode - looks don’t matter:** Complex visual design work in Design Mode (setting colors, fonts, spacing, etc.) will likely NOT meet any of the requirements for the Create PT. Don’t worry about how your app looks until *after* you already have code that will let you complete the written responses.

Practice Narrowing it Down

Below are three descriptions of potential projects that another CS Principles student is considering. For each write:

- Two or three ways they could narrow down the project using the tips above
- Opportunities to write an algorithm in their project even after it’s been narrowed down.

Project 1: Tic-Tac-Toe

“Here’s my idea: I want to build a tic-tac-toe game. The user creates an account if they don’t already have one and are taken to the main game board. From there the player will play against the computer in either easy, intermediate, or advanced mode, so I will need to write the code for the computer player. When the game is over their lifetime win total is updated. I will also keep track of how long the game took.”

Ways to narrow down the project (2 or 3)	Algorithm opportunities

Project 2: Health App
"I volunteer at my local health clinic so I want to build a health app. The user can record information about what they eat, how much they sleep, how much they exercise, and information like their blood pressure and weight. Based on the information provided the app will provide recommendations to the user about how they can improve their health for both diet and exercise. Users can also personalize the look of the app with different theme colors."

Ways to narrow down the project (2 or 3)	Algorithm opportunities

Project 3: Sports Stats
"I think that I'll build an app that allows the user to quickly record stats during a basketball game. The app will show a picture of the court. The user taps on the court to indicate something happened there. They are presented with a quick menu of options like: shot attempt, foul, steal, rebound, etc. then they select from another list which player did it. At the end of the game it displays a stat sheet for all of the players and the stats for that game."

Ways to narrow down the project (2 or 3)	Algorithm opportunities

Bring it All Together (15 mins)

With an understanding of the major components of the Create PT, you are ready to start brainstorming projects. While you have at least 12 class hours to complete the task, keep in mind that those 12 hours also must account for time to make your video and complete the written responses. We recommend budgeting at least 5 hours to complete the video and written responses, and so it is highly recommended that you prepare to do a project in which the programming / coding can be completed in 6-7 hours. You want projects with the following features.

- **Personally Relevant:** Pick projects you are actually interested in building.
- **Clear Purpose:** Aim for a simple program whose purpose can be stated in one sentence. For example:
  - The purpose of my program is _____.
  - My app does _____.
  - My program lets a user _____.
  - My app is a _____.
- **Narrowed Down:** Repeat the "Narrow it Down" process with your own ideas. A good rule of thumb is that you'll want to be able to have a first draft of your algorithm within two hours of starting to program.
- **No New Programming Skills:** Make sure you *already* have the programming skills necessary to complete the project. Be flexible. With some creativity you can likely use the skills you've already learned to make many different types of projects. Avoid taking on new programming environments or concepts as part of the Create PT.

Brainstorm Project Ideas

Brainstorm one or two project ideas for the Create PT. Afterwards you'll share ideas with a classmate for feedback.

Project Idea	Classmate Feedback
Purpose: *Ways to Narrow it Down:* *Algorithm Opportunities:* *Confidence you have skills to do this it in time allotted?:*	*Use the list above to give feedback on the idea.*
Purpose: *Ways to Narrow it Down:* *Algorithm Opportunities:* *Confidence you have skills to do this it in time allotted?:*	*Use the list above to give feedback on the idea.*

Create PT Progress / Check-in organizer

2a

Program **purpose**
(remember: your video should show this)

*Row **1***

2b

Development **process** overall

*Row **2***

Difficulty/opportunity #1
Circle one: feedback · testing · reflection
(how it was resolved, incorporated into project)

*Row **3***

Difficulty/opportunity #2
Circle one: feedback · testing · reflection
(how it was resolved, incorporated into project)

*Row **3***

2c

Identify main **algorithm** (name of function, location in code, etc.)

*Rows **4, 5***

How does it function? (explain any mathematical or logical concepts used).

*Row **5***

How does it relate to overall purpose?

*Row **5***

Algorithm Includes two or more algorithms....

Sub-algorithm 1

Sub-algorithm 2

Explain how at least one of the two (1) uses Mathematical or Logical concepts (2) can explain how it functions independently.

*Row **6***

2d

Your developed **abstraction**
(name of function(s) *you wrote*)

How does it manage complexity?

Examples:
- Able to reuse functionality?
- Problem broken down into functions and sub-functions?
- Generalizes specific behavior?

*Rows **7,8***

This organizer is inspired by the work of Jill Westerlund at abstractingCS.com. Recreated and modified with permission.

Create PT Completion Timeline

Before you start, you should think about how you are going to allocate your time for the 12 hours provided for the task. Below is a sample timeline that you can use to plan out how you will complete the Create Performance Task.

Hour	Suggested Activity	Your Plan
1 - 2	Begin building a program for a project you brainstormed. Carefully monitor whether you will finish enough of your project in time. Write down one opportunity or difficulty you've encountered for prompt **2b**. **Goal:** you should be confident after this first round of development that you'll be able to meet the requirements for algorithms (2c) and abstraction (2d). **Use the *Create PT Progress / Check-in organizer* to test yourself about whether you are on track.**	
3 - 4	Keep working. Check in after hour 4 once again on whether you are on track to complete responses. You should ideally know: ● The abstraction you will write about ● The algorithm you will write about Write down a second opportunity or difficulty you've encountered for prompt **2b**. **Use the *Create PT Progress / Check-in organizer* to test yourself about whether you are on track.**	
5 - 7	Finalize all programming. After this point you shouldn't be writing more code (beyond simple touch ups)	
8	Complete prompt **2b**, using the notes you took as you were programming.	
9	Record **video** of your program running and complete response **2a**	
10	Complete **2c** describing your algorithm	
11	Complete **2d** describing your abstraction	
12	Complete the task ● Review the submission materials ● Check your responses against the scoring guidelines ● Upload your video, written response, and program code to the digital portfolio **Goal:** At the end of this day, your Create PT is submitted!	

Note: The timeline above is just a guideline. You may complete the performance task on a different schedule. Make sure to leave enough time to complete your computational artifact and write-up.

Create PT Guidelines

Video Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. Your video must not exceed 1 minute in length and must not exceed 30MB in size

Prompt 2a. Provide a written response or audio narration in your video that:
- identifies the programming language;
- identifies the purpose of your program; and
- explains what the video illustrates.

(Must not exceed 150 words)

Advice: For resources on how to make your video, head to https://studio.code.org/s/csp-create-2019/stage/1/puzzle/2. Here are the most important things to remember for your video and prompt 2a.

- **Video Runs Continuously:** Your video must run continuously and show your actual code running. It can't just be a series of screenshots.
- **Show One Feature:** Your program does NOT need to be complete so long as you can demonstrate one major feature that's running.
- **Describe the Purpose:** The purpose of your program is the intended goal or objective of the program. In other words, it's "what" the program is supposed to do. If you made a game, an app, or some other kind of project, just quickly describe "what" kind of program it is and how it would be used / played.
- **Connection to Video:** Make sure that you can connect the purpose of your program to what is shown in the video. If you only have one feature working then describe the purpose of the feature.

Response 2a Checklist

Video
- ❏ Video runs continuously (it cannot be a series of screenshots)
- ❏ Video is less than 60 seconds long and less than 30MB in size
- ❏ Video demonstrates one running feature of the program

Written Response
- ❏ Response identifies the programming language used
- ❏ Identifies the purpose of the program
- ❏ Describes the feature(s) shown in the video and their connection to the purpose of the program
- ❏ May be audio commentary in your video. Carefully follow this checklist even if you use audio commentary.

2b. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and / or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development.
(Must not exceed 200 words)

Advice: There are *many* individual pieces of information you need to fit in this response. Use the checklist at the bottom carefully to make sure you don't miss any. Here are the most important pieces to remember:

- **Don't Forget the Overall Process:** You should reference your development process as a whole, NOT just two points in time. Your response should reflect an iterative process of identifying and solving problems focusing on programming decisions (i.e. writing a function to display score) vs design decisions (i.e. how to layout a screen).
- **Difficulties / Opportunities:** You need to describe *two* distinct difficulties / opportunities you encountered
  - A **difficulty** is likely either a bug in your code or a difficult design problem you needed to work out.
  - An **opportunity** is likely an idea or realization you had as you developed your code that led to new

ideas.

- **Feedback / Testing / Reflection:** You should clearly describe HOW you identified the two difficulties / opportunities - was it through testing your program out? Personal reflection? Through feedback from a peer?
- **Independent or Collaborative:** If you developed your program completely independently, you need to say so - don't assume the reader will know. If you developed your program collaboratively, some parts need to be done on your own. For this response *make sure*:
    - You clearly indicate which parts were done collaboratively
    - No identifying information is in your response. It's ok to say "I worked with a partner on this part of the program…". Don't say "I worked with Jesse S to create this part of the program...."
    - You clearly state which of the difficulties/opportunities described here was done *independently on your own* (could be both, but at least one).

Response 2b Checklist

Overall Development

- ❏ Response describes the *overall* development process, *not only* two key points.
- ❏ Response indicates whether you completed the project independently or with a partner. (note: this indication can be incorporated throughout your response *and* in comments within your code as well).

First Difficulty / Opportunity

- ❏ Response describes one difficulty / opportunity encountered early in the development process
- ❏ Response describes source of difficulty / opportunity as either feedback, testing, or reflection
- ❏ Response indicates how it was incorporated / solved, including whether you wrote the code independently.

Second Difficulty / Opportunity

- ❏ Response describes one difficulty / opportunity encountered later in the development process
- ❏ Response describes source of difficulty / opportunity as either feedback, testing, or reflection
- ❏ Response indicates how it was incorporated / solved, including whether you wrote the code independently.
- ❏ If first Difficulty / Opportunity WAS NOT solved independently, then this one must be

2c. Capture and paste a program code segment that implements an algorithm (marked with an **oval** in **section 3** below) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. *(Must not exceed 200 words)*

Advice: Review the "Is it a Good Algorithm" section above for lots of helpful tips on how to choose your algorithm. Here's the most important points.

- **You Wrote it:** You need to have written the code of your algorithm entirely on your own (not with a partner)
- **Copy and Paste it:** You must paste your actual algorithm code as part of this response.
- **A Parent and Two Children:** Your **selected** algorithm (the "parent") needs to have two **included** algorithms (the "children"). See Example Algorithms 4 and 5 for ideas about how this might look.
- **Mathematical / Logic Concepts:** At least one **included** algorithm needs to use mathematical and/or logical concepts.
- **Break Into Functions:** To make it easier to refer to individual parts of your algorithm give the **selected** and **included** algorithms their own functions. Example Algorithm 4 is written in this way.
- **Describe "how", not just "what":** You need to talk about how your code works, not just what the user will see when it runs. Do this by referring to the actual variables names, programming constructs, strings, and so on, that are visible in your code snippet. For example:

    "The algorithm I selected is signInUser() which handles the user login process in my app which has two key parts: checkName() and startHomeScreen(). checkName has an

if-statement that checks to see whether the name entered in the usernameTxt textbox is equal to 'MrSillyMan' or 'MsFunnyGal'. If it is then it sets the accessGranted variable to true, otherwise false. The startHomeScreen() function checks the accessGranted variable and returns the login screen if false, otherwise it proceeds to show the home screen for the user."

Response 2c Checklist

Overall

- ❏ You wrote all algorithm code yourself
- ❏ Response includes copy-pasted versions of code for main and sub-algorithms with ovals around them
- ❏ Response identifies the **selected** algorithm (parent) and at least two **included** algorithms (children).

Included algorithm 1

- ❏ Clearly identifies the code for the algorithm (where in the code, function name, line numbers, etc)
- ❏ Explains what the algorithm does independently
- ❏ Describes how the code of the algorithm works
- ❏ Uses mathematical or logical concepts

Included algorithm 2

- ❏ Clearly identifies the code for the algorithm (where in the code, function name, line numbers, etc).
- ❏ Explains what the algorithm does independently
- ❏ Describes how the code of the algorithm works
- ❏ Uses mathematical or logical concepts

Selected Algorithm

- ❏ Clearly identifies the code for the **selected** algorithm (where in the code, function name, line numbers, etc).
- ❏ Describes how **selected** algorithm combines **included** algorithms.
- ❏ Explains how **selected** algorithm helps to achieve the overall purpose of the program

2d. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a **rectangle** in **section 3** below). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. *(Must not exceed 200 words)*

Advice: Review the "*Is it a Good Abstraction?*" section above for tips on how to choose your abstraction. Here's the most important points.

- **Choose a Function:** Unless you feel confident about another abstraction, choose a function - not an onEvent, but a function you defined and named yourself.
- **You Wrote it:** You need to have written the code of your abstraction entirely on your own (not with a partner)
- **Copy and Paste it:** You must paste your actual abstraction code as part of this question submission.
- **Manages Complexity:** Make sure you can describe how your abstraction helps manage complexity in your program.
- **Make a contrasting argument:** explain how your program would be *more* complex to read, write, or reason about if you had *not* created your abstraction.
- **Mathematical and Logical Concepts:** While you should aim to include these in your abstraction, this is NOT explicitly assessed by the Scoring Guidelines for 2019.

Response 2d Checklist

Overall

- ❏ You wrote all abstraction code yourself (it's not an onEvent block, but a function you defined and named)
- ❏ Response includes copy-pasted versions of code for abstraction with a rectangle around it
- ❏ Response identifies the abstraction by name
- ❏ You explicitly describe HOW the abstraction manages complexity (e.g. by explaining how your code would be more complex to write or reason about without the abstraction)

3. Program Code

Capture and paste your entire program code in this section.

› Mark with an oval the segment of program code that implements the algorithm you created for your program that integrates other algorithms and integrates mathematical and/or logical concepts.

› Mark with a rectangle the segment of program code that represents an abstraction you developed.

› Include comments or acknowledgments for program code that has been written by someone else.

Advice: For resources on how to make a PDF of your program code head to https://studio.code.org/s/csp-create-2019/stage/1/puzzle/2. Here's the most important things to remember:

- **Making Your PDF:** Use CodePrint to make a PDF of your program. It's designed specifically for the Create PT. You can find it from the link above.

- **Marking Your Algorithm:** Make sure you place an oval around all parts of your algorithm (**selected** and **included**).

- **Marking Your Abstraction:** Place your rectangle around the code where you create your abstraction (e.g. define a function) not where you use the abstraction (e.g. call a function).

- **Commenting and Collaboration:** You may work with a partner on the Create PT, but you must clearly indicate which parts you completed independently and which you completed together by using comments. For example:

  ```
  // I completed the section below with my collaborative partner
  // I completed this section independently
  // I have extended code found at [URL]. The code below is my additions
  ```

  Remember that your algorithm and abstraction need to be created entirely independently.

- **Citing Images:** If you use code or images made by someone else (for example that you found online) then you should cite those resources as well. Again you can use comments.

  ```
  // The images used in this app came from:
  // [1] bird image - http://name-of-site.com/path/to/image.jpg
  // [2] flower image - http://site.com/path/to/flower.jpg
  ```

Unit 7 Lesson 3

Create PT - Complete the Task (12 hrs)

Resources

Create PT Survival Guide 2019/2020[1]

Create PT Overview

Goal of the Task: Create a programming project of your own design and then explain the **purpose, process, algorithms and abstractions** used to build it. You have **12 hours** to complete this task.

What you Submit: (1) Video of running program (2) Written Responses to prompts 2a-d (3) PDF of program code

How you get a good score: The AP committee wants to see that you can:

- Design and write the code for a computer program with a topic of your choosing
- Describe how you identified and solved problems as you developed your program
- Write code for an algorithm and describe its purpose within your program
- Write code for an abstraction and describe its purpose within your program

Suggested Process in a Nutshell (see also the Sample Timeline on following pages)**:**

Pick your project… Something small enough that you can design and write in only a few hours. You should basically know ahead of time what the core algorithm will be.

Hours 1-2 **Develop a "good enough" program / prototype within 2 hours**

- ❏ Evaluate whether you can finish what you set out to do, and adjust as necessary.
- ❏ Check progress for responses 2c and 2d - algorithms and abstraction.
- ❏ You should **know** what **your algorithm and abstraction** will be at this point.

Hours 3-7 **Keep coding and get to a stopping point with ~5 hours to go**

- ❏ Your video and written responses will take time to make — you'll want to make last minute improvements!

Hour 8 **Record your video and respond to 2a**

Hours 9-10 **Write responses to 2b, 2c, 2d**

Hours 11-12 **Prepare to submit**

- ❏ Finalize program code and written responses and submit on the digital portfolio as three separate files.
  - ❏ Video of running program (.mp4, .wmv, .avi, or .mov)
  - ❏ Written Response to prompts 2a-2d (PDF)
  - ❏ Program Code (PDF)

[1] *Much of the content of this this guide was inspired by Jill Westerlund at the Abstracting CS blog. We are grateful for Jill's ingenuity and generosity.*

Algorithms on the Create PT (20 mins)

Is it a Good Algorithm?

Algorithm - College Board Definitions: Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages. Algorithms make use of sequencing, selection or iteration. People write programs to execute algorithms. Your algorithm must include two or more algorithms that work in combination to achieve some desired result. Finally it must integrate mathematical and/or logical concepts.

What it means: Algorithms are chunks of code that accomplish a task. To make sure your algorithm is a little complex and that you demonstrate your programming abilities the College Board has a few specific requirements.

- **Something You Wrote:** The entirety of the code you submit as your algorithm should be something that you wrote entirely on your own. You cannot submit code that a partner helped you write.

- **Mathematical and/or Logical Concepts:** Mathematical concepts means code where your app does some sort of mathematical computation (+, -, *, /, %). These are typically used when your program is making a calculation.

  Logical concepts means code where you use logical or comparison operators (&&, ||, !, ==, !=, <, > <= >=). These are typically used in combination with if-statements where your program is making a decision.

- **A Parent and Two Children:** Your **selected** algorithm must have two **included** algorithms that work in combination to achieve some result. The best way to think about this is that you have a “parent” algorithm that requires two or more components that can stand on their own as independent “child” algorithms.

  You should NOT submit three separate algorithms. Instead you should find one large task in your program that can actually be divided into two or more sub-tasks (for example: a large task might be making a user login screen; subtasks are (1) checking their login information (2) updating the screen appropriately). The code to accomplish each sub-task are the **included** “child” algorithms that can work independently to solve each sub-task, but are combined to work together as the **selected** “parent” algorithm to solve the overall task.

- **Talking About Your Algorithm:** It is recommended that you place your **selected** algorithm and **included** algorithms in separate functions. This will make it easier for you to talk about your algorithms in your responses. You may also, however, simply place rectangles around them and then use line numbers to identify your algorithms.

Does it Count? - Algorithm Edition

The AP reader has to judge strictly from the code you include in your response to 2c whether it meets the requirements. They will assume that all of the screen elements and variables the code refers to exist, and that the code is the working code from your video.

Row 6 - Response 2C: The points for Row 6 of the scoring guidelines are awarded strictly for the code segment selected.

Criteria	Decision Rules	Scoring Notes
Selected code segment implements an algorithm that **includes** at least two or more algorithms. **AND** At least one of the **included** algorithms uses mathematical or logical concepts. **AND** Explains how one of the **included** algorithms functions independently.	Responses are still eligible to earn this row, even if they do not earn row 5. The **included** algorithms can be sub-parts of the algorithm in row 5. **Do NOT award a point if any one of the following is true:** • the **selected** algorithm consists of a single instruction; • the **selected** algorithm consists solely of library calls to existing language functionality; • neither of the **included** algorithms nor the **selected** algorithm that includes two or more algorithms uses mathematical or logical concepts; • the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or • the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).	• Algorithms make use of sequencing, selection or iteration. • Mathematical concepts include mathematical expressions using arithmetic operators and mathematical functions. • Logical concepts include Boolean algebra and compound expressions. • Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times. • Selection uses a Boolean condition to determine which of two parts of an algorithm is used.

You be the AP Reader! You are are the AP reader trying to determine if they get the point for Row 6. Assume each algorithm snippet below was submitted as part of written response 2c. For each, select yes or no - should the point be awarded, and explain why. (It doesn't have to be a written explanation. You can draw an arrow or circle something in the code with a brief word or label about why.)

Example Algorithm 1
`onEvent("button1", "click", function() {`
`  setProperty("screen1","background-color","blue");`
`});`
`onEvent("id", "click", function(event) {`
`  setProperty("screen1","background-color","red");`
`});`
	Earn Point? Yes / No
	
	Why?

Example Algorithm 2
`onEvent("button1", "click", function() {`
`  setProperty("label1","background-color","blue");`
`  setProperty("label1","font-size",randomNumber(20,40));`
`  playSound("sound://default.mp3",false);`
`});`
	Earn Point? Yes / No
	
	Why?

Example Algorithm 3
`function manageLogIn() {`
`  if (loginStatus=="loggedIn") {`
`  showHomeScreen();`
`  } else {`
`  resetLogIn();`
`  }`
`}`
`function showHomeScreen() {`
`  setScreen("homeScreen");`
`  setText("id", "text");`
`}`
`function resetLogIn() {`
`  loginStatus = "loggedOut";`
`  setScreen("signInScreen");`
`}`
	Earn Point? Yes / No
	
	Why?

Example Algorithm 4

```
function increaseScore(){
  score = score + 1;
  checkIfEndGame();
  if(endGame){
    setText("timeText", "Time:" + totalTime + " secs!")
  } else {
    updateScreen();
  }
}

function updateScreen(){
  setPosition("button1",randomNumber(0,320), 100);
  setText("scoreText", score);
  playSound("sound://default.mp3", false);
}

function checkIfEndGame(){
  var endTime = getTime();
  totalTime = (startTime - endTime)/1000;
  endGame = false;
  if(score > 10 || totalTime > 20){
      endGame=true;
  }
}
```

Earn Point? Yes / No

Why?

Example Algorithm 5

```
while (done == false){
  var n = promptNum("Enter the secret number");

  if(n == 97){
    done = true;
    console.log("You did it! Begin countdown!");
    for(var i=97; i>=0; i--){
      console.log(i);
    }
    console.log("You're in!");
  } else {
    console.log("You've guessed incorrectly);
    if (n < 97){
      console.log("Guess higher");
    else {
      console.log("Guess lower");
    }
  }
}
```

Earn Point? Yes / No

Why?

Abstraction on the Create PT (20 mins)

Is it a Good Abstraction?

Abstraction - College Board Definitions The AP course framework includes the following statements related to abstraction and programming that are relevant for the Create PT:

- *Multiple levels of abstraction are used to write programs (EU 2.2)*
- *The process of developing an abstraction involves removing detail and generalizing functionality. (EK 2.2.1A)*
- *(Student can) Use abstraction to manage complexity in programs. (LO 5.3.1 [P3])*

What it means: Abstractions manage complexity. Programming abstractions make it easier to write complex programs. The AP reader is checking that you can **create, identify, and describe an abstraction** that manages complexity in your code.

- **Something You Wrote:** The code you submit as your abstraction should be something that you wrote entirely on your own. You cannot submit code that a partner helped you write.

- **Good Abstraction Choice - Functions that you wrote:** A function that you wrote means a function that you defined, named, and wrote code for (i.e code that starts `function myFunc(){ ... })`. Your function can demonstrate *managing complexity* if it either (or both):
  - gets called from multiple different places in your code
  - has a parameter that generalizes some behavior (and also probably called from several places)

- **Bad Abstraction Choice: `onEvent` and other built-in programming language/environment features**
  - `onEvent` does NOT count as a "student-developed abstraction". OnEvent is provided by the programming environment and the code contained within it is used in a single location in your code. Therefore it does not in any way manage complexity.
  - Identifying `onEvent` as an abstraction is the most common way to **lose** credit on this question.
  - **Variables** by themselves are not a data abstraction
  - Anything that is an existing abstraction provided by the language that you are simply using. For example: loops and logical structures are not student developed abstractions

*`onEvent` **IS NOT** an abstraction*

Does it Count? - Abstraction Edition

The AP reader has to judge strictly from the code you include in your response to 2d whether it meets the requirements. They will assume that all of the screen elements and variables the code refers to exist, and that the code is the working code from your video.

Row 7 - Response 2D: The points for Row 7 of the scoring guidelines are awarded strictly for the code segment selected.

Criteria	Decision Rules	Scoring Notes

• Selected code segment is a student-developed abstraction.	Responses that use existing abstractions to create a new abstraction, such as creating a list to represent a collection (e.g., a classroom, an inventory), would earn this point.
	
	Do NOT award a point if any one of the following is true:
	• the response is an **existing** abstraction such as variables, existing control structures, event handlers, APIs;
	• the code segment consisting of the abstraction is not included in the written responses section or is not explicitly identified in the program code section; or
	• the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly
		• The following are examples of abstractions (EK 5.3.1):
		  ○ Procedures
		  ○ Parameters
		  ○ Lists
		  ○ Application program interfaces (APIs)
		• Libraries
		• Lists and other collections can be treated as abstract data types (ADTs) in developing programs. (EK 5.5.1l)

	identifying the code segment containing the abstraction).	

Evaluate Abstractions! Each of the code segments below shows a portion of code with a rectangle around it. Sometimes additional code is included to help you understand the context of that abstraction (for example, to determine whether the abstraction helps manage complexity).

For each example below respond to 3 things:

- **Earn Point? Yes / No** - Would the selected code segment earn the point for row 7?
- **Why?** - note why it does or doesn't earn the point
- **Manages Complexity? Yes / No** - based on what you can see, are you able to argue that the abstraction manages complexity?

Example 1
`onEvent("btn1", "click", function(event) {`
`  setText("btn1", "I'm touched.");`
`  setProperty("btn1", "font-size", 100);`
`  showElement("lbl1");`
`});`
	Earn Point? Yes / No
	
	Why?
	
	Manages Complexity? Yes / No

Example 2
`1 onEvent("btn1", "click", function(event) {`
`2  if(score < 0){`
`3  setScreen("gameOverScreen")`
`4  }`
`5 });`
	Earn Point? Yes / No
	
	Why?
	
	Manages Complexity? Yes / No

Example 3	**Earn Point? Yes / No** **Why?** **Manages Complexity? Yes / No**

```
onEvent("btn1", "click", function(event) {
  updatePlayerTurn();
  updateScore(-5);

  if(score < 0){
    setScreen("gameOverScreen")
  }
  else{
    updateBoard();
  }
});
```

Example 4

```
onEvent("btn1", "click", function(event) {
  setButtonProps();
});

function setButtonProps(){
  setText("btn1", "I'm touched.");
  setProperty("btn1", "font-size", 100);
  showElement("lbl1");
}
```

Earn Point? Yes / No

Why?

Manages Complexity? Yes / No

Example 5

```
onEvent("btn1", "click", function(event) {
  setButtonProps();
});

onEvent("screen1", "click", function(event) {
  setButtonProps();
});

function setButtonProps(){
  setText("btn1", "I'm touched.");
  setProperty("btn1", "font-size", 100);
  showElement("lbl1");
}
```

Earn Point? Yes / No

Why?

Manages Complexity? Yes / No

Example 6

Earn Point? Yes / No

Why?

```
onEvent("btn1", "click", function(event) {
  updateScore(5);
});

onEvent("screen1", "click", function(event) {
  updateScore(-3);
});

function updateScore(amt){
  totalScore += amt;
  if( totalScore < 0 ){
     setScreen("gameOverScreen");
  }
}
```

Manages Complexity? **Yes / No**

“Narrow it Down” (15 mins)

You should assume that you’re not going to have enough time to complete the “perfect” project for the Create PT. **This is okay because you can get full credit for a programming project that feels incomplete as long as it has working elements.** While a large or more complete project is satisfying, your score is based mostly on the written responses, which is how you demonstrate that you understand the concepts covered on the Create PT.

All your project actually needs to include is:

- One working feature that you can demonstrate in your video
- An algorithm
- An abstraction

You will make the project much easier if you narrow down your original idea into a simpler target project. This will give you more time to complete the written responses or make smaller improvements.

How to Narrow it Down: Narrowing it down means identifying the sub-tasks or sub-problems that meet the PT requirements. Here’s a few strategies to help you do that:

- **Get to the Algorithm:** Split your project into individual components that will likely require an algorithm that meets the Create PT requirements. Each of these components individually could be your entire project.

- **Pick One Part of a Bigger Idea:** Think about your project as a set of programming tasks that each solve part of a larger problem. Some of these individual tasks might suffice for the Create PT. You can just pick one part of a big idea that meets the requirements that you think you can program in the time allotted.

- **Minimal Design Mode - looks don’t matter:** Complex visual design work in Design Mode (setting colors, fonts, spacing, etc.) will likely NOT meet any of the requirements for the Create PT. Don’t worry about how your app looks until *after* you already have code that will let you complete the written responses.

Practice Narrowing it Down

Below are three descriptions of potential projects that another CS Principles student is considering. For each write:

- Two or three ways they could narrow down the project using the tips above
- Opportunities to write an algorithm in their project even after it’s been narrowed down.

Project 1: Tic-Tac-Toe

“Here’s my idea: I want to build a tic-tac-toe game. The user creates an account if they don’t already have one and are taken to the main game board. From there the player will play against the computer in either easy, intermediate, or advanced mode, so I will need to write the code for the computer player. When the game is over their lifetime win total is updated. I will also keep track of how long the game took.”

Ways to narrow down the project (2 or 3)	Algorithm opportunities

Project 2: Health App
"I volunteer at my local health clinic so I want to build a health app. The user can record information about what they eat, how much they sleep, how much they exercise, and information like their blood pressure and weight. Based on the information provided the app will provide recommendations to the user about how they can improve their health for both diet and exercise. Users can also personalize the look of the app with different theme colors."

Ways to narrow down the project (2 or 3)	Algorithm opportunities

Project 3: Sports Stats
"I think that I'll build an app that allows the user to quickly record stats during a basketball game. The app will show a picture of the court. The user taps on the court to indicate something happened there. They are presented with a quick menu of options like: shot attempt, foul, steal, rebound, etc. then they select from another list which player did it. At the end of the game it displays a stat sheet for all of the players and the stats for that game."

Ways to narrow down the project (2 or 3)	Algorithm opportunities

Bring it All Together (15 mins)

With an understanding of the major components of the Create PT, you are ready to start brainstorming projects. While you have at least 12 class hours to complete the task, keep in mind that those 12 hours also must account for time to make your video and complete the written responses. We recommend budgeting at least 5 hours to complete the video and written responses, and so it is highly recommended that you prepare to do a project in which the programming / coding can be completed in 6-7 hours. You want projects with the following features.

- **Personally Relevant:** Pick projects you are actually interested in building.
- **Clear Purpose:** Aim for a simple program whose purpose can be stated in one sentence. For example:
  - The purpose of my program is _____.
  - My app does _____.
  - My program lets a user _____.
  - My app is a _____.
- **Narrowed Down:** Repeat the "Narrow it Down" process with your own ideas. A good rule of thumb is that you'll want to be able to have a first draft of your algorithm within two hours of starting to program.
- **No New Programming Skills:** Make sure you *already* have the programming skills necessary to complete the project. Be flexible. With some creativity you can likely use the skills you've already learned to make many different types of projects. Avoid taking on new programming environments or concepts as part of the Create PT.

Brainstorm Project Ideas

Brainstorm one or two project ideas for the Create PT. Afterwards you'll share ideas with a classmate for feedback.

Project Idea	Classmate Feedback
Purpose: *Ways to Narrow it Down:* *Algorithm Opportunities:* *Confidence you have skills to do this it in time allotted?:*	*Use the list above to give feedback on the idea.*
Purpose: *Ways to Narrow it Down:* *Algorithm Opportunities:* *Confidence you have skills to do this it in time allotted?:*	*Use the list above to give feedback on the idea.*

Create PT Progress / Check-in organizer

2a

Program **purpose**
(remember: your video should show this)

*Row **1***

2b

Development **process** overall

*Row **2***

Difficulty/opportunity #1
Circle one: feedback · testing · reflection
(how it was resolved, incorporated into project)

*Row **3***

Difficulty/opportunity #2
Circle one: feedback · testing · reflection
(how it was resolved, incorporated into project)

*Row **3***

2c

Identify main **algorithm** (name of function, location in code, etc.)

*Rows **4, 5***

How does it function? (explain any mathematical or logical concepts used).

*Row **5***

How does it relate to overall purpose?

*Row **5***

Algorithm Includes two or more algorithms....

Sub-algorithm 1

Sub-algorithm 2

Explain how at least one of the two (1) uses Mathematical or Logical concepts (2) can explain how it functions independently.

*Row **6***

2d

Your developed **abstraction**
(name of function(s) *you wrote*)

How does it manage complexity?

Examples:
- Able to reuse functionality?
- Problem broken down into functions and sub-functions?
- Generalizes specific behavior?

*Rows **7,8***

This organizer is inspired by the work of Jill Westerlund at abstractingCS.com. Recreated and modified with permission.

Create PT Completion Timeline

Before you start, you should think about how you are going to allocate your time for the 12 hours provided for the task. Below is a sample timeline that you can use to plan out how you will complete the Create Performance Task.

Hour	Suggested Activity	Your Plan
1 - 2	Begin building a program for a project you brainstormed. Carefully monitor whether you will finish enough of your project in time. Write down one opportunity or difficulty you've encountered for prompt **2b**. **Goal:** you should be confident after this first round of development that you'll be able to meet the requirements for algorithms (2c) and abstraction (2d). **Use the *Create PT Progress / Check-in organizer* to test yourself about whether you are on track.**	
3 - 4	Keep working. Check in after hour 4 once again on whether you are on track to complete responses. You should ideally know: ● The abstraction you will write about ● The algorithm you will write about Write down a second opportunity or difficulty you've encountered for prompt **2b**. **Use the *Create PT Progress / Check-in organizer* to test yourself about whether you are on track.**	
5 - 7	Finalize all programming. After this point you shouldn't be writing more code (beyond simple touch ups)	
8	Complete prompt **2b**, using the notes you took as you were programming.	
9	Record **video** of your program running and complete response **2a**	
10	Complete **2c** describing your algorithm	
11	Complete **2d** describing your abstraction	
12	Complete the task ● Review the submission materials ● Check your responses against the scoring guidelines ● Upload your video, written response, and program code to the digital portfolio **Goal:** At the end of this day, your Create PT is submitted!	

Note: The timeline above is just a guideline. You may complete the performance task on a different schedule. Make sure to leave enough time to complete your computational artifact and write-up.

Create PT Guidelines

Video Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. Your video must not exceed 1 minute in length and must not exceed 30MB in size

Prompt 2a. Provide a written response or audio narration in your video that:
- identifies the programming language;
- identifies the purpose of your program; and
- explains what the video illustrates.

(Must not exceed 150 words)

Advice: For resources on how to make your video, head to https://studio.code.org/s/csp-create-2019/stage/1/puzzle/2. Here are the most important things to remember for your video and prompt 2a.

- **Video Runs Continuously:** Your video must run continuously and show your actual code running. It can't just be a series of screenshots.
- **Show One Feature:** Your program does NOT need to be complete so long as you can demonstrate one major feature that's running.
- **Describe the Purpose:** The purpose of your program is the intended goal or objective of the program. In other words, it's "what" the program is supposed to do. If you made a game, an app, or some other kind of project, just quickly describe "what" kind of program it is and how it would be used / played.
- **Connection to Video:** Make sure that you can connect the purpose of your program to what is shown in the video. If you only have one feature working then describe the purpose of the feature.

Response 2a Checklist

Video
- ❏ Video runs continuously (it cannot be a series of screenshots)
- ❏ Video is less than 60 seconds long and less than 30MB in size
- ❏ Video demonstrates one running feature of the program

Written Response
- ❏ Response identifies the programming language used
- ❏ Identifies the purpose of the program
- ❏ Describes the feature(s) shown in the video and their connection to the purpose of the program
- ❏ May be audio commentary in your video. Carefully follow this checklist even if you use audio commentary.

2b. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and / or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development.
(Must not exceed 200 words)

Advice: There are *many* individual pieces of information you need to fit in this response. Use the checklist at the bottom carefully to make sure you don't miss any. Here are the most important pieces to remember:

- **Don't Forget the Overall Process:** You should reference your development process as a whole, NOT just two points in time. Your response should reflect an iterative process of identifying and solving problems focusing on programming decisions (i.e. writing a function to display score) vs design decisions (i.e. how to layout a screen).
- **Difficulties / Opportunities:** You need to describe *two* distinct difficulties / opportunities you encountered
  - A **difficulty** is likely either a bug in your code or a difficult design problem you needed to work out.
  - An **opportunity** is likely an idea or realization you had as you developed your code that led to new

ideas.

- **Feedback / Testing / Reflection:** You should clearly describe HOW you identified the two difficulties / opportunities - was it through testing your program out? Personal reflection? Through feedback from a peer?
- **Independent or Collaborative:** If you developed your program completely independently, you need to say so - don't assume the reader will know. If you developed your program collaboratively, some parts need to be done on your own. For this response *make sure*:
    - You clearly indicate which parts were done collaboratively
    - No identifying information is in your response. It's ok to say "I worked with a partner on this part of the program…". Don't say "I worked with Jesse S to create this part of the program...."
    - You clearly state which of the difficulties/opportunities described here was done *independently on your own* (could be both, but at least one).

Response 2b Checklist

Overall Development

- ❏ Response describes the *overall* development process, *not only* two key points.
- ❏ Response indicates whether you completed the project independently or with a partner. (note: this indication can be incorporated throughout your response *and* in comments within your code as well).

First Difficulty / Opportunity

- ❏ Response describes one difficulty / opportunity encountered early in the development process
- ❏ Response describes source of difficulty / opportunity as either feedback, testing, or reflection
- ❏ Response indicates how it was incorporated / solved, including whether you wrote the code independently.

Second Difficulty / Opportunity

- ❏ Response describes one difficulty / opportunity encountered later in the development process
- ❏ Response describes source of difficulty / opportunity as either feedback, testing, or reflection
- ❏ Response indicates how it was incorporated / solved, including whether you wrote the code independently.
- ❏ If first Difficulty / Opportunity WAS NOT solved independently, then this one must be

2c. Capture and paste a program code segment that implements an algorithm (marked with an **oval** in **section 3** below) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. *(Must not exceed 200 words)*

Advice: Review the "Is it a Good Algorithm" section above for lots of helpful tips on how to choose your algorithm. Here's the most important points.

- **You Wrote it:** You need to have written the code of your algorithm entirely on your own (not with a partner)
- **Copy and Paste it:** You must paste your actual algorithm code as part of this response.
- **A Parent and Two Children:** Your **selected** algorithm (the "parent") needs to have two **included** algorithms (the "children"). See Example Algorithms 4 and 5 for ideas about how this might look.
- **Mathematical / Logic Concepts:** At least one **included** algorithm needs to use mathematical and/or logical concepts.
- **Break Into Functions:** To make it easier to refer to individual parts of your algorithm give the **selected** and **included** algorithms their own functions. Example Algorithm 4 is written in this way.
- **Describe "how", not just "what":** You need to talk about how your code works, not just what the user will see when it runs. Do this by referring to the actual variables names, programming constructs, strings, and so on, that are visible in your code snippet. For example:

    "The algorithm I selected is signInUser() which handles the user login process in my app which has two key parts: checkName() and startHomeScreen(). checkName has an

if-statement that checks to see whether the name entered in the usernameTxt textbox is equal to 'MrSillyMan' or 'MsFunnyGal'. If it is then it sets the accessGranted variable to true, otherwise false. The startHomeScreen() function checks the accessGranted variable and returns the login screen if false, otherwise it proceeds to show the home screen for the user."

Response 2c Checklist

Overall

- ❏ You wrote all algorithm code yourself
- ❏ Response includes copy-pasted versions of code for main and sub-algorithms with ovals around them
- ❏ Response identifies the **selected** algorithm (parent) and at least two **included** algorithms (children).

Included algorithm 1

- ❏ Clearly identifies the code for the algorithm (where in the code, function name, line numbers, etc)
- ❏ Explains what the algorithm does independently
- ❏ Describes how the code of the algorithm works
- ❏ Uses mathematical or logical concepts

Included algorithm 2

- ❏ Clearly identifies the code for the algorithm (where in the code, function name, line numbers, etc).
- ❏ Explains what the algorithm does independently
- ❏ Describes how the code of the algorithm works
- ❏ Uses mathematical or logical concepts

Selected Algorithm

- ❏ Clearly identifies the code for the **selected** algorithm (where in the code, function name, line numbers, etc).
- ❏ Describes how **selected** algorithm combines **included** algorithms.
- ❏ Explains how **selected** algorithm helps to achieve the overall purpose of the program

2d. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a **rectangle** in **section 3** below). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. *(Must not exceed 200 words)*

Advice: Review the "*Is it a Good Abstraction?*" section above for tips on how to choose your abstraction. Here's the most important points.

- **Choose a Function:** Unless you feel confident about another abstraction, choose a function - not an onEvent, but a function you defined and named yourself.
- **You Wrote it:** You need to have written the code of your abstraction entirely on your own (not with a partner)
- **Copy and Paste it:** You must paste your actual abstraction code as part of this question submission.
- **Manages Complexity:** Make sure you can describe how your abstraction helps manage complexity in your program.
- **Make a contrasting argument:** explain how your program would be *more* complex to read, write, or reason about if you had *not* created your abstraction.
- **Mathematical and Logical Concepts:** While you should aim to include these in your abstraction, this is NOT explicitly assessed by the Scoring Guidelines for 2019.

Response 2d Checklist

Overall

- ❏ You wrote all abstraction code yourself (it's not an onEvent block, but a function you defined and named)
- ❏ Response includes copy-pasted versions of code for abstraction with a rectangle around it
- ❏ Response identifies the abstraction by name
- ❏ You explicitly describe HOW the abstraction manages complexity (e.g. by explaining how your code would be more complex to write or reason about without the abstraction)

3. Program Code

Capture and paste your entire program code in this section.

› Mark with an oval the segment of program code that implements the algorithm you created for your program that integrates other algorithms and integrates mathematical and/or logical concepts.

› Mark with a rectangle the segment of program code that represents an abstraction you developed.

› Include comments or acknowledgments for program code that has been written by someone else.

Advice: For resources on how to make a PDF of your program code head to https://studio.code.org/s/csp-create-2019/stage/1/puzzle/2. Here's the most important things to remember:

- **Making Your PDF:** Use CodePrint to make a PDF of your program. It's designed specifically for the Create PT. You can find it from the link above.

- **Marking Your Algorithm:** Make sure you place an oval around all parts of your algorithm (**selected** and **included**).

- **Marking Your Abstraction:** Place your rectangle around the code where you create your abstraction (e.g. define a function) not where you use the abstraction (e.g. call a function).

- **Commenting and Collaboration:** You may work with a partner on the Create PT, but you must clearly indicate which parts you completed independently and which you completed together by using comments. For example:

  ```
  // I completed the section below with my collaborative partner
  // I completed this section independently
  // I have extended code found at [URL]. The code below is my additions
  ```

  Remember that your algorithm and abstraction need to be created entirely independently.

- **Citing Images:** If you use code or images made by someone else (for example that you found online) then you should cite those resources as well. Again you can use comments.

  ```
  // The images used in this app came from:
  // [1] bird image - http://name-of-site.com/path/to/image.jpg
  // [2] flower image - http://site.com/path/to/flower.jpg
  ```

Create PT - Written Response Template

Assessment Overview and Performance Task Directions for Students

Video Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. Your video must not exceed 1 minute in length and must not exceed 30MB in size **Prompt 2a.** Provide a written response or audio narration in your video that: ● identifies the programming language; ● identifies the purpose of your program; and ● explains what the video illustrates. *(Must not exceed 150 words)*

2b. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and / or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development. *(Must not exceed 200 words)*

2c. Capture and paste a program code segment that implements an algorithm (marked with an **oval** in **section 3**) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. *(Must not exceed 200 words)*
Code Segment
Written Response

2d. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a **rectangle** in **section 3**). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. *(Must not exceed 200 words)*
Code Segment
Written Response

Export or save this document as a PDF and turn in to the AP Digital Portfolio along with your **Video** and **Program Code** (separate files).

Name(s)_____ Period _____ Date _____

Create PT Programming Journal

Entry 1

Date:

What did I accomplish today?

Explain any code written today. What does it do?

Did anything go wrong? Explain in detail.

How did I solve it? Circle or highlight the technique you used and explain below:

testing **feedback** **reflection**

(Optional) Take a picture of your work today (code, Survival Guide, App Design, etc - whatever you were working on). Paste the picture here.

Name(s)_____ Period _____ Date _____

Entry 2

What did I accomplish today? How does today's work build upon previous work?

Explain any code written today. What does it do?

Did anything go wrong? Explain in detail.

How did I solve it? Explain below and circle or highlight the technique you used:

testing **feedback** **reflection**

(Optional) Take a picture of your work today (code, Survival Guide, App Design, etc - whatever you were working on). Paste the picture here.

Entry 3

What did I accomplish today? How does today's work build upon previous work?

Explain any code written today. What does it do?

Name(s)_____ Period _____ Date _____

Did anything go wrong? Explain in detail.

How did I solve it? Explain below and circle or highlight the technique you used:
testing feedback reflection

(Optional) Take a picture of your work today (code, Survival Guide, App Design, etc - whatever you were working on). Paste the picture here.

Entry 4

What did I accomplish today? How does today's work build upon previous work?

Explain any code written today. What does it do?

Did anything go wrong? Explain in detail.

Name(s)_____ Period _____ Date _____

How did I solve it? Explain below and circle or highlight the technique you used:
testing feedback reflection

(Optional) Take a picture of your work today (code, Survival Guide, App Design, etc - whatever you were working on). Paste the picture here.

Entry 5

What did I accomplish today? How does today's work build upon previous work?

Explain any code written today. What does it do?

Did anything go wrong? Explain in detail.

How did I solve it? Explain below and circle or highlight the technique you used:
testing feedback reflection

Name(s)_____ Period _____ Date _____

(Optional) Take a picture of your work today (code, Survival Guide, App Design, etc - whatever you were working on). Paste the picture here.

Entry 6

What did I accomplish today? How does today's work build upon previous work?

Explain any code written today. What does it do?

Did anything go wrong? Explain in detail.

How did I solve it? Explain below and circle or highlight the technique you used:
testing feedback reflection

(Optional) Take a picture of your work today (code, Survival Guide, App Design, etc - whatever you were working on). Paste the picture here.

Entry 7

What did I accomplish today? How does today's work build upon previous work?

Name(s)_____ Period _____ Date _____

Explain any code written today. What does it do?

Did anything go wrong? Explain in detail.

How did I solve it? Explain below and circle or highlight the technique you used:
testing feedback reflection

(Optional) Take a picture of your work today (code, Survival Guide, App Design, etc - whatever you were working on). Paste the picture here.

Entry 8

What did I accomplish today? How does today's work build upon previous work?

Explain any code written today. What does it do?

Did anything go wrong? Explain in detail.

Name(s)_____ Period _____ Date _____

How did I solve it? Explain below and circle or highlight the technique you used:

testing feedback reflection

(Optional) Take a picture of your work today (code, Survival Guide, App Design, etc - whatever you were working on). Paste the picture here.