

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



<b>Total score</b>	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6	Row 7	Row 8	<i>This document combines student sample, scoring guidelines and scoring commentary from: <a href="#">Create PT Sample F</a></i>
<b>Sample: F</b>	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
 <p><i>My program is called "Nyan Cat Clicker". The programming language that was used was JavaScript in a program called AppLab. <b>This program is a clicker game,</b></i></p>	<p><b>Row 1</b> <b>Response 2A</b></p> <p>The video demonstrates the running of at least one feature of the program submitted.</p> <p><b>AND</b></p> <p>The response (audio narration or written response) identifies the purpose of the program (what the program is attempting to do).</p>	<p>Response earns the point if it explains the function of the program instead of identifying the purpose.</p> <p>Response earns the point if the illustrated feature runs, even if it does not function as intended.</p> <p>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.</p> <p><b>Do NOT award a point if any one of the following is true:</b></p> <ul style="list-style-type: none"> <li>• a video is not submitted;</li> <li>• the video does not illustrate the feature mentioned in the response; or</li> <li>• the video does not illustrate the running of the feature (screen shots or storyboards are not acceptable and would not be credited).</li> </ul>

<p><i>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.</i></p>	<p><b>The response earned the point for this row.</b> The video illustrates features of the program, and the response states the purpose of the program.</p>
--	--

**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	
<p><i>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</i></p>	Row and Task	Decision Rules
	<p><b>Row 2 - Response 2B</b></p> <p>Describes or outlines steps used in the incremental and iterative development process to create the entire program.</p>	<p><b>Do NOT award a point if any one of the following is true:</b></p> <ul style="list-style-type: none"> <li>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</li> <li>the response does not indicate iterative development;</li> <li>refinement and revision are not connected to feedback, testing, or reflection; or</li> <li>the response only describes the development at two specific points in time.</li> </ul>
<p><b>The response DID NOT earn the point for this row.</b> The response does not describe the incremental and iterative design process for the entire program.</p>		

<p>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.</p>	<p><b>Code.org Commentary:</b> 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.</p>	
	<p><b>Row 3 - Response 2B</b></p> <p>Specifically identifies at least two program development difficulties or opportunities.</p> <p style="text-align: center;"><b>AND</b></p> <p>Describes how the two identified difficulties or opportunities are resolved or incorporated.</p>	<p>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.</p> <p><b>Do NOT award a point if any one of the following is true:</b></p> <ul style="list-style-type: none"> <li>• only one distinct difficulty or opportunity in the process is identified and described; or</li> <li>• the response does not describe how the difficulties or opportunities were resolved or incorporated.</li> </ul>
	<p><b>The response earned the point for this row.</b></p> <p>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."</p>	

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
<pre>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) {</pre>	<p><b>Row 4</b></p> <p><b>Response 2C</b></p> <p>Selected code segment implements an algorithm.</p>	<p><b>Do NOT award a point if any one of the following is true:</b></p> <ul style="list-style-type: none"> <li>• the algorithm consists of a single instruction;</li> <li>• 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</li> <li>• the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code</li> </ul>

```

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 reimplementaion 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;

	<p style="text-align: center;"><b>AND</b></p> <p>At least one of the included algorithms uses mathematical or logical concepts.</p> <p style="text-align: center;"><b>AND</b></p> <p>Explains how one of the included algorithms functions independently..</p>	<ul style="list-style-type: none"> <li>• neither of the included algorithms nor the selected algorithm that includes two or more algorithms uses mathematical or logical concepts;</li> <li>• 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</li> <li>• <b>the algorithm is not explicitly identified</b> (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).</li> </ul>
<p><b>The response DID NOT earn the point for this row.</b> The response <b>does not clearly identify two algorithms used by the selected algorithm.</b></p> <p><b>Code.org Commentary:</b> To receive this point, the two included algorithms must be defined and explained in addition to the selected algorithm</p>		

**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	
<pre>function Time() {   myInterval = setInterval(function() {     seconds = seconds - 1;     setText("Time",seconds);     if (seconds === 0){       if (Score &gt; 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); }</pre> <p><i>In order to use the , such as certain lines of code are not repeated, I made my</i></p>	Row and Task	Decision Rules
	<p><b>Row 7</b> <b>Response 2D</b></p> <p>Selected code segment is a student-developed abstraction.</p>	<p>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.</p> <p><b>Do NOT award a point if any one of the following is true:</b></p> <ul style="list-style-type: none"> <li>• the response is an existing abstraction such as variables, existing control structures, event handlers, APIs;</li> <li>• 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</li> <li>• 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).</li> </ul>
<p><b>The response earned the point for this row.</b> The code segment given represents an abstraction (a procedure or <b>function</b>).</p>		

<p><i>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.</i></p>	<p><b>Row 8 Response 2D</b></p> <p>Explains how the selected abstraction manages the complexity of the program.</p>	<p>Responses should not be penalized for explanations of abstractions that are not developed by the student.</p> <p><b>Do NOT award a point if any one of the following is true:</b></p> <ul style="list-style-type: none"> <li>• the explanation does not apply to the selected abstraction; or</li> <li>• 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).</li> </ul>
	<p><b>The response earned the point for this row.</b> 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."</p>	

### 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.