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)

Hello, today I will be showcasing to you my app and its 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).
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)

<table>
<thead>
<tr>
<th>Student Response</th>
<th>Scoring Guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td>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</td>
<td>Do NOT award a point if any one of the following is true:</td>
</tr>
<tr>
<td></td>
<td>Describes or outlines steps used in the incremental and iterative development process to create the entire program.</td>
</tr>
<tr>
<td>Row 2 - Response 2B</td>
<td></td>
</tr>
<tr>
<td>The response DID NOT earn the point for this row.</td>
<td></td>
</tr>
<tr>
<td>The response does not describe the overall interactive or iterative development of the entire program.</td>
<td></td>
</tr>
<tr>
<td>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).</td>
<td></td>
</tr>
</tbody>
</table>
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.

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)

<table>
<thead>
<tr>
<th>Student Response</th>
<th>Scoring Guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row 4</strong></td>
<td><strong>Decision Rules</strong></td>
</tr>
<tr>
<td>Response 2C</td>
<td>Do NOT award a point if any one of the following is true:</td>
</tr>
<tr>
<td></td>
<td>● the algorithm consists of a single instruction;</td>
</tr>
<tr>
<td></td>
<td>● 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</td>
</tr>
<tr>
<td></td>
<td>● 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>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</td>
<td>The response earned the point for this row. The selected block of code represents an algorithm.</td>
</tr>
<tr>
<td></td>
<td><strong>Code.org Commentary:</strong> The block of code in this response includes both sequencing and selection thus qualifying as an algorithm.</td>
</tr>
</tbody>
</table>
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.

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

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 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.

Row 6
Response 2C

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.
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**

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

**Selected code segment** is a student-developed abstraction.

**Scoring Guidelines**

<table>
<thead>
<tr>
<th>Row and Task</th>
<th>Decision Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row 7</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.</td>
</tr>
<tr>
<td>Response 2D</td>
<td>Do NOT award a point if any one of the following is true:</td>
</tr>
<tr>
<td></td>
<td>● the response is an existing abstraction such as variables, existing control structures, event handlers, APIs;</td>
</tr>
<tr>
<td></td>
<td>● 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</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>The response DID NOT earn the point for this row.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The boxed code fragment does not represent an abstraction.</td>
</tr>
</tbody>
</table>

**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.

**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.