**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.

(Boost not exceed 150 words)

<table>
<thead>
<tr>
<th>Student Response</th>
<th>Scoring Guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row and Task</strong></td>
<td><strong>Decision Rules</strong></td>
</tr>
<tr>
<td><strong>Row 1</strong></td>
<td>Response earns the point if it explains the function of the program instead of identifying the purpose.</td>
</tr>
<tr>
<td><strong>Response 2A</strong></td>
<td>Response earns the point if the illustrated feature runs, even if it does not function as intended.</td>
</tr>
<tr>
<td>The video demonstrates the running of at least one feature of the program submitted. <strong>AND</strong> The response (audio narration or written response) identifies the purpose of the program (what the program is attempting to do).</td>
<td>Response earns the point if the video includes a narration or some form of closed captioning that addresses the purpose of the program.</td>
</tr>
</tbody>
</table>

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.
flipping them over two at a time. The program itself creates 12 cards with 6 pairs of color dots to be matched randomly put on the 12 cards. It allows two users to take turns flipping two cards over and then either keeping them flipped over if they are a match or flipping them over once more to cover the color dot. The video illustrates that the cards can be flipped to play the game and it shows the random card placement. (121 words)

The video demonstrates the flipping of the cards and random card placement. The response matches the video and indicates that the purpose of the program overall is to be a memory game with matching of cards.

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

My project was created independently, so I had to break down the steps to create this in a manageable way for myself. I started with just creating the card which involved a cover, a background and a color dot. My main difficulty even at this early step was figuring out how to make the color dots random which started to take up the time I had to make this project, so I decided to make that a separate part of the program to not inhibit the creation process and first focused on making the blank card objects and covers. From there I had to go a step up and create the whole board of blank cards as a function utilizing the previous functions I made to draw a single card. At this point I was able to create a function to solve my earlier problem that shuffled the card order made in the board because the cards were stored as an array to be put on the screen rather than just directly put on the screen. (176 words)

Scoring Guidelines

Row and Task | Scoring Guidelines | 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 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 a point for this row.
The response describes the overall development process used to develop this program. The response includes how the process was incremental and iterative. The response states, that an aspect of the program was hindering the development, so it was “decided to make that a separate part of the program to not inhibit the creation process.” This problem was then resolved later in the development after other parts were developed.

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
The response DID NOT earn a point for this row.
The response only describes one difficulty and one resolution. The one difficulty described in the response is with making the color dots random. It is resolved by storing the cards in an array.

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)

```javascript
function drawCard(color, cardNumber, cardCoverNumber) {
    createCanvas(cardNumber, 86, 92);
    setActiveCanvas(cardNumber);
    setPosition(cardNumber, cardXPosition, cardYPosition);
    drawCardBase(cardXPosition, cardYPosition);
    drawSymbol(color);
    createCanvas(cardCoverNumber);
    setActiveCanvas(cardCoverNumber);
    setPosition(cardCoverNumber, cardXPosition, cardYPosition);
    drawCardCover(cardXPosition, cardYPosition);
}
```

Student Response

<table>
<thead>
<tr>
<th>Scoring Guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row and Task</td>
</tr>
<tr>
<td>Decision Rules</td>
</tr>
<tr>
<td>Row 4</td>
</tr>
<tr>
<td>Do NOT award a point if any one of the following is true:</td>
</tr>
</tbody>
</table>
aspects is created using its own function to make the individual part, but by combing all of these into a single function, I was able to later use this to create a usable entity instead of three different shapes. This function also allows the cards to be placed based on the inputted cardNumber so as to have an organized board. (80 words)

<table>
<thead>
<tr>
<th>Row 5</th>
<th>Response 2C</th>
<th>The response earned a point for this row.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Selected code segment implements an algorithm that uses mathematical or logical concepts.</td>
<td>The selected code segment, function drawCard(), is an algorithm.</td>
</tr>
</tbody>
</table>
|       | **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) **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 a point for this row. The selected code segment does not include mathematical or logical concepts. | The response DID NOT earn a point for this row. The selected code segment does not include mathematical or logical concepts. |
| Row 6 | Response 2C | Do NOT award a point if any one of the following is true: **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; |
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 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 a point for this row.
The selected code segment includes two or more algorithms. The selected code segment does not show the inner workings of the included algorithms, so it is unclear if the algorithms include mathematical or logical concepts.
The response does not explain how any of the included algorithms functions.

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)

```javascript
function drawBoard() {
    var cardColors = [];
    for (var card=0; card<6; card++) {
        cardColors[2*card] = 2*card;
        cardColors[2*card+1] = 2*card+1;
    }
    cardColors = shuffle(cardColors);
    //Shuffles the cards
    for (card=0; card<12; card++) {
        //sets twelve cards
        if (cardColors[card] === 0 || cardColors[card] == 10) {
            ...
    }
}
```

Note: This student submitted the code for a single function, drawBoard(), that spanned 3 pages. These are the first few lines. We've pasted screen shots of the rest of the code below.
<table>
<thead>
<tr>
<th>Student Response</th>
<th>Scoring Guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td>My main abstraction was splitting the board down into its component parts before using the drawBoard function to combine all of the abstracted parts such as the shuffling of the cards or the creation of the cards. This allowed me to focus on parts that would not directly interact at first, the making of the cards and the shuffling of the cards, before then making the program more complicated. (69 words)</td>
<td></td>
</tr>
</tbody>
</table>

<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>
<tr>
<td></td>
<td>● 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>
</tbody>
</table>

**The response DID NOT earn a point for this row.**
The selected code segment does not explicitly identify the abstraction.

\[**Code.org commentary:** The expectation from 2018 onward is that the student’s written response should explicitly state something along the lines of: “The main abstraction that I developed was the drawBoard function which...”. The response shown here is poorly worded because it leaves too much for the reader to infer, or assume that the student is talking about an abstraction they developed.\]

| Row 8 | Responses should not be penalized for explanations of abstractions that are not developed by the student. |
| Response 2D | 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 earned a point for this row.**
The response indicates that the splitting of the board into component parts allowed the programmer to “focus on parts that would not directly interact at first, the making of the cards and the shuffling of the cards, before then making the program more complicated.”