2. Written Responses

Submit one PDF document in which you respond directly to each prompt. Clearly label your responses **2a** – **2d in order**. **Your response to all prompts combined must not exceed 750 words, exclusive of the Program Code.**

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.

(Approximately 150 words)

In my program, Card-Wars, I built a card game similar to blackjack in the programming language Flash Actionscript 3.0 in the game developer Adobe Animate. The purpose of the game is to calculate and add the randomized cards a player has and match their current value to meet the requirements in an algorithm. Once the program runs the results through the algorithm the program displays a message for the reader telling them the current status of the game.The end result of this is the player can play against another player and the program can tell them which player won. The video illustrates two players playing against each other to win all the other players cash. In the video you can see the messaging system that displays the game's status on the bottom left corner, and the cash value system that displays each player's cash value on the right side. The programs purpose to the user is to act as an enjoyable game that incorporates useful software design strategies to make the game a fun experience.

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. *(Approximately 200 words)*

In developing my program I faced challenges, difficulties, and opportunities. One challenge I faced in designing my program was working with global frames and symbol frames embedded in movie clips. This was difficult because my program required me to call frames from the main timeline to the timeline in a movieclip. After working with the coding language I discovered how to successfully call a frame in the movieclip from a frame in the main timeline. This also created a challenge on how I would determine the correct frame in the movie clip's timeline. In solving this I created a randomization algorithm and set the return value to a variable that was equal to the frame that the movieclip went to. This project was also full of opportunities. It gave me a chance to learn more about working with Adobe Animate and expanded my understanding in the Actionscript 3.0 language. Another challenge I faced in programming this project was declaring a label in flash as dynamic text and then declaring a variable to display as the value of the label, so the character could see the score they got. I conquered this challenge by embedding a font into the label or dynamic text box and setting the text value to the declared variable.

2c. Capture and paste the program code segment that implements an algorithm (marked with an **oval in section 3** below) that is fundamental for your program to achieve its intended purpose. Your code segment must include an algorithm that integrates other algorithms and integrates 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. *(Approximately 200 words)*

One of the most important parts and key component of my program is the use of mathematical algorithms and logical concepts. These algorithms function independently and in correlation with others to form a working and reliable program that accomplishes its intended purpose. Without algorithms the program has a 100% rate of not working because they are the only reason the cards appear, your score is shown, or your cash is calculated. Most algorithms in my program run on logical concepts and if statements because the program has to deal with multiple variables and each of these variables has to meet certain requirements for the program to accomplish something. The algorithm shown in my example incorporates the use of mathematical data achieved from my randomization function and cycles it through a if else statement at each enterframe condition. This algorithm works with each different player's score to determine which score is the best and how to correctly display the outcome of playing the game to the player. Without this function the player would not know who won the game or what the status of the hand is. The several if else statements work together to display the correct status and read the scores so they are accurately displayed. The hardest part of making this algorithm was the use of several different logical operators on one variable and then determining out of all the conclusions acquired through these operators which is the correct solution.

2d. Capture and paste the program code segment that contains an abstraction you developed (marked with a **rectangle in section 3** below). Your abstraction should integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. *(Approximately 200 words)*

This detailed example of abstraction that I developed in creating my program aided in my success in comprehending the program's complexity. It is important to use abstraction when coding in order to manage and organize the program. These are valuable effects of abstraction because they allow for better understanding from an outside viewer and help me, the programmer, to determine how to fix a problem if one occurred. Keeping your code organized and manageable allows for a more efficient and useable program. The use of abstraction in programming also is valuable to the planning and developing process of writing the code. Without abstraction writing the code correctly would be increasingly difficult. Abstraction allows for the programmer to determine the correct steps in writing the program. This example of abstraction, displayed in a graphical image, is used to determine which function the program starts with so I can start adding code within them and continuing to edit the program. It was important to start my program with the method of abstraction because it is important for my coding design t be organized and manageable. The example of abstraction was taken from frame three of my program in the main timeline. This frame is the most important in my program and without the use of abstraction it could have created many errors.