# Create — Applications from Ideas
# Written Response Submission Template

## Submission Requirements

### 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.** Identify the programming language and identify the purpose of your program. Explain your video using one of the following:

- A written summary
- of what the video illustrates OR
- An audio narration in your video. If you choose this option, your response to the written summary should read, "The explanation is located in the video."

(Approximately 150 words)

Insert response for 2a in the text box below.

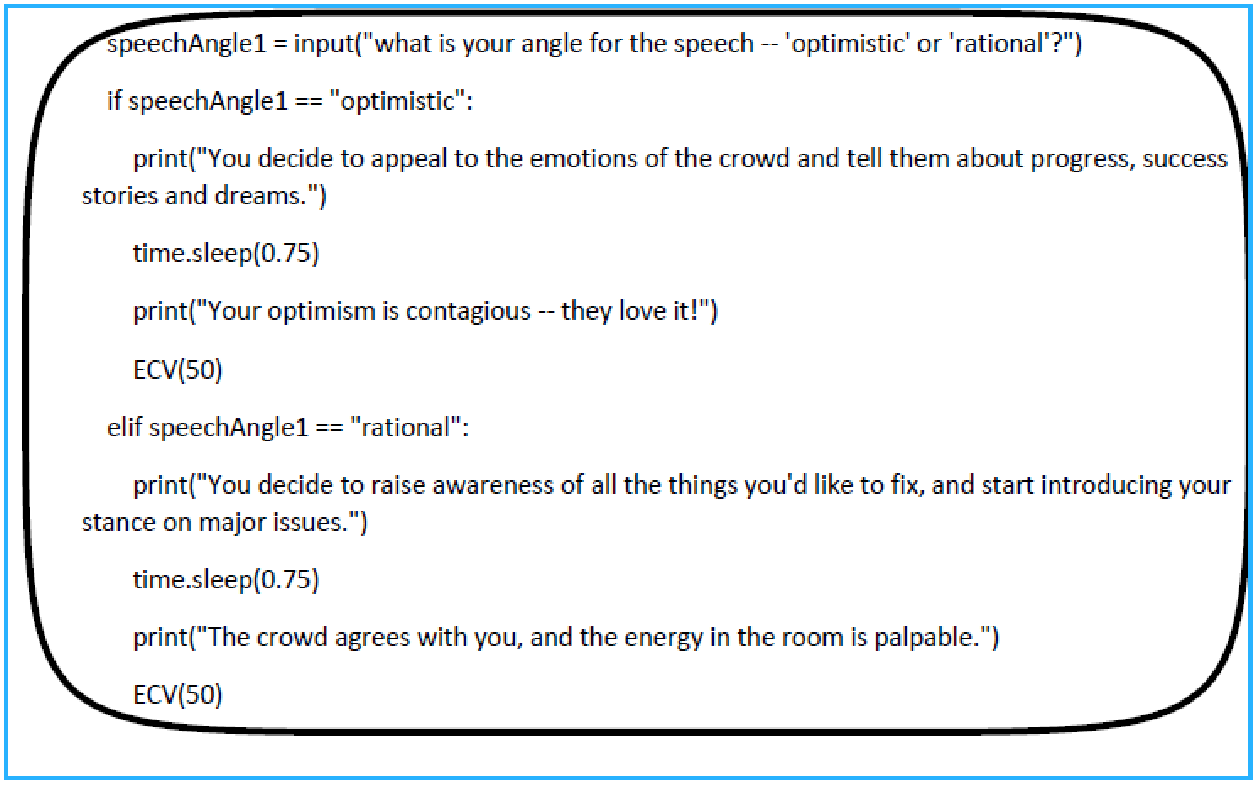
My program is a text-based simulation that is meant to represent a presidential campaign. It is a choose-your-own-adventure story, and incorporates the input of the user to determine how the situation plays out and how the storyline continues. Its purpose was solely to present a story in an interesting way; instead of just reading something, the user actually has to interact with the plotline and their decisions impact how the narrative continues. The video shows almost the entire program running, including the places where the user's input is necessary to determine what comes next. The video starts a few seconds into the program running, and ends just moments before, as there was no way to make the entire thing fit the time constraints. I wrote my program using Python, because I am very familiar with it and it allowed me to best execute the ideas I had for this 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; the second could refer to either collaborative or independent program development. *(Approximately 200 words)*

Insert response for 2b in the text box below.

The first difficulty I had to overcome was with deciding how to make the plotline work. I devised the storyline for my text-based adventure independently, and resolved my difficulties the same way. I couldn't decide how best to make an interesting story that I could program and incorporate complex logic at the same time. The "political campaign" storyline created another challenge in that I needed to make it unbiased and unaffiliated. I had to use very neutral terminology and be very cautious to avoid bringing in my own views. The second difficulty in the development process was working out the scoring system. I wrote the program individually, so this was something I needed to resolve on my own. This program simulated a political campaign, and therefore required a way to win or lose. I decided that the user's score had to be determined by their choices, so I created a function that took a parameter, and the value given as the parameter would increase the user's final score (shown as "electoralCollegeVotes") by that amount.

**2c.** **Capture and paste an image or images of your program code segment** that implements the most complex algorithm you wrote. (marked with a **color border** below)



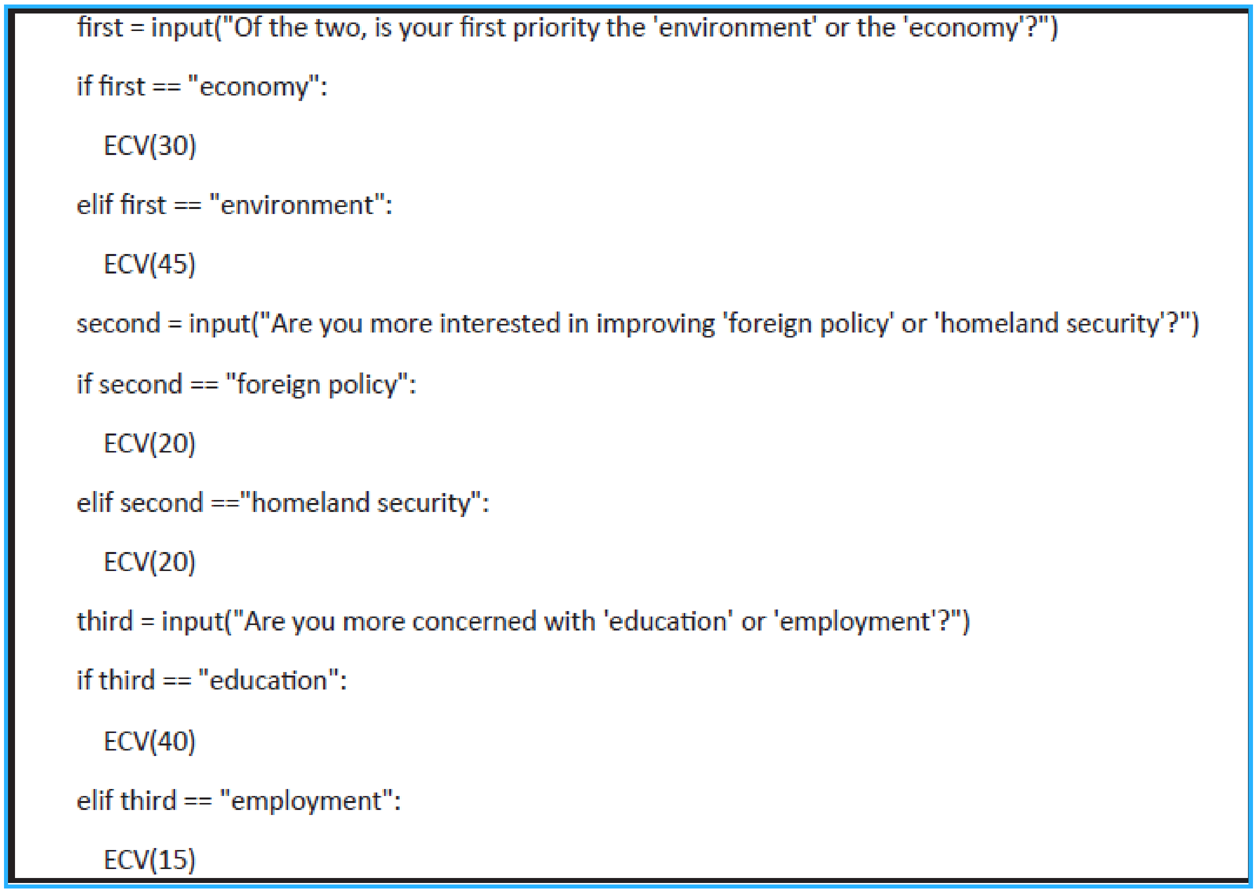
```
 speechAngle1 = input("what is your angle for the speech -- 'optimistic' or 'rational'?")
 if speechAngle1 == "optimistic":
   print("You decide to appeal to the emotions of the crowd and tell them about progress, success
stories and dreams.")
   time.sleep(0.75)
   print("Your optimism is contagious -- they love it!")
   ECV(50)
 elif speechAngle1 == "rational":
   print("You decide to raise awareness of all the things you'd like to fix, and start introducing your
stance on major issues.")
   time.sleep(0.75)
   print("The crowd agrees with you, and the energy in the room is palpable.")
   ECV(50)
```

Your algorithm should integrate several mathematical and logical concepts. Describe the mathematical and logical concepts used to develop the algorithm. Explain the complexity of the algorithm and how it functions in the program.
*(Approximately 200 words)*

Insert text response for 2c in the plain box below.

This algorithm uses logic to determine the outcome of this situation. The player's decision impacts what happens during this situation, and this occurs again at multiple points throughout the program. Input is crucial to how this program runs, and I used logic at many points to allow for different options. In this instance, the outcome of the program is based on what the user decides to do, and their decision affects the plotline at this point, and then again at the end. Here, the user determines whether they will take a more optimistic or pragmatic approach to the situation, and depending on which option they go with, the text that follows will be different. This uses logic and boolean operators to get the desired result, and although it is not incredibly complicated, this process is extremely crucial to how the program runs from start to finish. I use the same technique at many other points in the program, and the structure of this function was used multiple times afterwards.

**2d. Capture and paste an image or images** of the program code segment that contains an abstraction you developed (marked with a matching **blue color border** below)



```
first = input("Of the two, is your first priority the 'environment' or the 'economy'?")
if first == "economy":
   ECV(30)
elif first == "environment":
   ECV(45)
second = input("Are you more interested in improving 'foreign policy' or 'homeland security'?")
if second == "foreign policy":
   ECV(20)
elif second =="homeland security":
   ECV(20)
third = input("Are you more concerned with 'education' or 'employment'?")
if third == "education":
   ECV(40)
elif third == "employment":
   ECV(15)
```

Your abstraction should integrate mathematical and logical concepts.
Explain how your abstraction helped manage the complexity of your program.
*(Approximately 200 words)*

Insert text response for 2d in the plain box below.

This is an algorithm because it calls a function within another function, eliminating the need for rewriting the same code over and over. The smaller function, ECV, is called at the top of the prrogram and allows for a value to be added to the final point tally, the Electoral College Votes accumulated throughout the simulation. When it is called within a separate function, it allows for the decision made by the player to impact the outcome of the game, and display a more tangible indication of how the player is doing. Each time this ECV function is called, it requires a parameter, which determines the number of points added. It is shown here as a part of the function firstRally, and different numbers of points are added to the Electoral College score based on the decision made. I had originally struggled with developing a way to make the scoring system work, but I had not yet explored this method. It worked when I called the smaller function within a larger one, and it allowed me to reduce the complexity of this program and make it easier to keep adding onto the value and create the final score.