**Python Journal Template**

**Directions:** Follow the directions for each part of the journal template. Include in your response all the elements listed under the Requirements section. Prompts in the Inspiration section are not required; however, they may help you to fully think through your response.

Remember to review the Touchstone page for entry requirements, examples, and grading specifics.

**Name: Anthony Mascari**

**Date: 08/13/2023**

**Final Replit Program Share Link:** https://github.com/Nice-Take/sophia\_python.git

Complete the following template. Fill out all entries using complete sentences.

## PART 1: Defining Your Problem

|  |
| --- |
| **Task**  State the problem you are planning to solve.  **Requirements**   * Describe the problem you are trying to solve for. * Describe any input data you expect to use. * Describe what the program will do to solve the problem. * Describe any outputs or results the program will provide.   **Inspiration**  When writing your entry below ask yourself the following questions:   * Why do you want to solve this particular problem? * What source(s) of data do you believe you will need? Will the user need to supply that data, or will you get it from an external file or another source? * Will you need to interact with the user throughout the program? Will users continually need to enter data in and see something to continue? * What are your expected results or what will be the end product? What will you need to tell a user of your program when it is complete? |
| **Problem:**  When it comes time for a family to choose a house pet, things can get complicated quickly. Tensions can rise and conflict can cloud the decision-making process even further. Often there is a need to organize thoughts, qualify and quantify numerous factors. These factors can become exceedingly difficult to keep track of mentally when emotions and tensions are high so a program to help a family solve this issue would be incredibly helpful.  We will need the user to input information that pertains to their particular circumstances. There will be initial user input then possibly a second round of required user input in order to provide the best recommendation possible. At the end of the program the user can expect to receive a single recommendation of what pet they should consider for their family.  **Required Input:**  We will need several entries from the user in order to help them best choose a pet.   1. How many people are in the family? 2. How much weight does that persons’ preference carry? 3. Does this person prefer a mammal or reptile pet? 4. Does this person prefer to cuddle or spectate? 5. Is anyone in the family allergic to pet dander? 6. Do you have noise restrictions where you live? 7. Annual pet budget? 8. How many days will you be away from your pet at a time and how often per year? 9. Do you have a yard?   **Program Solution:**  The program will operate by having a number of different pets hardcoded with appropriate attributes in a module called “pet\_options.py”. It will then take input for each member of the family, account for that user’s decision power and match the family with best fit pet.  **Desired Output:**  The program will provide the best fit pet solution for the current family and their set of circumstances. |

## PART 2: Working Through Specific Examples

|  |
| --- |
| **Task**  Write down clear and specific steps to solve a simple version of your problem you identified in Part 1.  **Requirements**  Complete the three steps below **for at least two distinct examples/scenarios**.   * State any necessary input data for your simplified problem. * Write clear and specific steps in English (not Python) detailing what the program will do to solve the problem. * Describe the specific result of your example/scenario.   **Inspiration**  When writing your entry below ask yourself the following questions:   * Are there any steps that you don’t fully understand? These are places to spend more time working out the details. Consider adding additional smaller steps in these spots. * Remember that a computer program is very literal. Are there any steps that are unclear? Try giving the steps of your example/scenario to a friend or family member to read through and ask you questions about parts they don’t understand. Rewrite these parts as clearly as you can. * Are there interesting edge cases for your program? Try to start one of your examples/scenarios with input that matches this edge case. How does it change how your program might work? |
| The first thing to appear in the program will be a welcome message and a brief explanation of what the program does, and what the user can expect to receive at the end of the program.  The user will then be prompted to answer a number of questions.  **Required Input:**   1. How many people are in the family?   *For each family member they will be asked:*   1. How much weight does this persons’ preference carry? 2. Does this person prefer a mammal or reptile pet? 3. Does this person prefer to cuddle or spectate?   *The remaining questions will only be asked of the user one time.*   1. Is anyone in the family allergic to pet dander? 2. Do you have noise restrictions where you live? 3. Annual pet budget? 4. How many days will you be away from your pet at a time and how often per year? 5. Do you have a yard?   **Steps:**   1. Receive input 2. Create objects containing information for each family/member 3. Compare the user input to the hardcoded “pet\_options.py” 4. Numerically score matching options between the pet\_option classes and the user\_input 5. Return the pet option with the highest score to the user with a brief explanation   **Scenario 1:**  User Input:   1. How many people are in the family? **1** 2. How much weight does that persons’ preference carry (1-10)? **10** 3. Does this person prefer a mammal or reptile pet? **Mammal** 4. Does this person prefer to cuddle or spectate? **Cuddle** 5. Is anyone in the family allergic to pet dander? **No** 6. Do you have noise restrictions where you live? **No** 7. Annual pet budget? $**1000** 8. How many days will you be away from your pet at a time? **2** 9. Do you have a yard? **Yes**   RESULT: Dog because they like to cuddle, are not allergic to dander, have enough budget and a fitting lifestyle to comfortably own a dog as a house-pet.  **Scenario 1:**  User Input:   1. How many people are in the family? **4** 2. How much weight does PERSON1’s preference carry (1-10)? **5** 3. Does this person prefer a mammal or reptile pet? **Mammal** 4. Does this person prefer to cuddle or spectate? **Cuddle** 5. How much weight does PERSON2’s preference carry (1-10)? **5** 6. Does this person prefer a mammal or reptile pet? **Mammal** 7. Does this person prefer to cuddle or spectate? **Spectate** 8. How much weight does PERSON3’s preference carry (1-10)? **10** 9. Does this person prefer a mammal or reptile pet? **Mammal** 10. Does this person prefer to cuddle or spectate? **Spectate** 11. How much weight does PERSON4’s preference carry (1-10)? **10** 12. Does this person prefer a mammal or reptile pet? **Reptile** 13. Does this person prefer to cuddle or spectate? **Spectate** 14. Is anyone in the family allergic to pet dander? **No** 15. Do you have noise restrictions where you live? **No** 16. Annual pet budget? $**1000** 17. How many days will you be away from your pet at a time? **2** 18. Do you have a yard? **Yes**   RESULT: Cat because they like on average prefer to spectate, are not allergic to dander, have enough budget and a fitting lifestyle to comfortably own a cat as a house-pet. |
| Edge Case Possibilities:  We may encounter a situation where everything is weighted evenly with an even number of family member where we have two identical match scores. This is a scenario we will have to keep an eye on as we are solving and test for in order to ensure we don’t encounter an error at runtime or a recommendation that is not useful to the end-user. |
|  |

## PART 3: Generalizing Into Pseudocode

|  |
| --- |
| **Task**  Write out the general sequence your program will use, including all specific examples/scenarios you provided in Part 2.  **Requirements**   * Write pseudocode for the program in English but refer to Python program elements where they are appropriate. The pseudocode should represent the full functionality of the program, not just a simplified version. Pseudocode is broken down enough that the details of the program are no longer in any paragraph form. One statement per line is ideal.   **Help with writing pseudocode**   * Here are a few links that can help you write pseudocode with examples. Remember to check out part 3 of the Example Journal Template Submission if you have not already. Note: everyone will write pseudocode differently. There is no right or wrong way to write it other than to make sure you write it clearly and in as much detail as you can so that it should be easy to convert it to code later.   + <https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/>   + <https://www.wikihow.com/Write-Pseudocode>   **Inspiration**  When writing your entry below ask yourself the following questions:   * Do you see common program elements and patterns in your specific examples/scenarios in Part 2, like variables, conditionals, functions, loops, and classes? These should be part of your pseudocode for the general sequence as well. * Are there places where the steps for your examples/scenarios in Part 2 diverged? These may be places where errors may occur later in the project. Make note of them. * When you are finished with your pseudocode, does it make sense, even to a person that does not know Python? Aim for the clearest description of the steps, as this will make it easier to convert into program code later. |
| <Write your pseudocode here>  Print -> Welcome to my program!\nThis program exists to help with the decision of what type of   house-pet will best suit your family!  family\_size = Input(First we have a few questions:\nHow many people are in the family?) |
| 1. How many people are in the family? **1** 2. How much weight does that persons’ preference carry (1-10)? **10** 3. Does this person prefer a mammal or reptile pet? **Mammal** 4. Does this person prefer to cuddle or spectate? **Cuddle** 5. Is anyone in the family allergic to pet dander? **No** 6. Do you have noise restrictions where you live? **No** 7. Annual pet budget? $**1000** 8. How many days will you be away from your pet at a time? **2** 9. Do you have a yard? **Yes** |

## PART 4: Testing Your Program

|  |
| --- |
| **Task**  While writing and testing your program code, describe your tests, record any errors, and state your approach to fixing the errors.  **Requirements**   * For at least one of your test cases, describe how your choices for the test helped you understand whether the program was running correctly or not.   For each error that occurs while writing and testing your code:   * Record the details of the error from Replit. A screenshot or copy-and-paste of the text into the journal entry is acceptable. * Describe what you attempted in order to fix the error. Clearly identify what approach was the one that worked.   **Inspiration**  When writing your entry below ask yourself the following questions:   * Have you tested edge cases and special cases for the inputs of your program code? Often these unexpected values can cause errors in the operation of your program. * Have you tested opportunities for user error? If a user is asked to provide an input, what happens when they give the wrong type of input, like a letter instead of a number, or vice versa? * Did the outcome look the way you expected? Was it formatted correctly? * Does your output align with the solution to the problem you coded for? |
| <Record your errors and fixes here> |

## PART 5: Commenting Your Program

|  |
| --- |
| **Task**  Submit your full program code, including thorough comments describing what each portion of the program should do when working correctly.  **Requirements**   * The purpose of the program and each of its parts should be clear to a reader that does not know the Python programming language.   **Inspiration**  When writing your entry, you are encouraged to consider the following:   * Is each section or sub-section of your code commented to describe what the code is doing? * Give your code with comments to a friend or family member to review. Add additional comments to spots that confuse them to make it clearer. |
| <Copy your full program code here, including comments> |

## PART 6: Your Completed Program

|  |
| --- |
| **Task**  Provide the Replit link to your full program code.  **Requirements**   * The program must work correctly with all the comments included in the program.   **Inspiration**   * Check before submitting your touchstone that your final version of the program is running successfully. |
| https://github.com/Nice-Take/sophia\_python.git |