This program must contain the following features:

1. Write a new entry - Show the user a random prompt (from a list that you create), and save their response, the prompt, and the date as an Entry.
2. Display the journal - Iterate through all entries in the journal and display them to the screen.
3. Load the journal from a file - Prompt the user for a filename and then save the current journal (the complete list of entries) to that file location.
4. Save the journal to a file - Prompt the user for a filename and then load the journal (a complete list of entries) from that file. This should replace any entries currently stored in memory.
5. Provide a menu that allows the user choose these options
6. Your list of prompts must contain at least five different prompts. Make sure to add your own prompts to the list, but the following are examples to help get you started:
   * Who was the most interesting person I interacted with today?
   * What was the best part of my day?
   * How did I see the hand of the Lord in my life today?
   * What was the strongest emotion I felt today?
   * If I had one thing I could do over today, what would it be?
7. Your interface should generally follow the pattern shown in the video demo below.

**Design Requirements**

In addition, your program must:

1. Contain classes for the major components in the program.
2. Contain at least two classes in addition to the Program class.
3. Demonstrate the principle of abstraction by using member variables and methods appropriately.

**Showing Creativity and Exceeding Requirements**

Meeting the core requirements makes your program eligible to receive a 93%. To receive 100% on the assignment, you need to show creativity and exceed these requirements.

Here are some ideas you might consider:

* Think of other problems that keep people from writing in their journal and address one of those.
* Save other information in the journal entry.
* Improve the process of saving and loading to save as a .csv file that could be opened in Excel (make sure to account for quotation marks and commas correctly in your content.
* Save or load your document to a database or use a different library or format such as JSON for storage.

Report on what you have done to exceed requirements by adding a description of it in a comment in the Program.cs file.

WRITE

DISPLAY

LOAD

SAVE

QUIT

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Geri Rusk |  |  |  |  |  |  |  |  |
| CSE 210, Section 12 |  |  |  |  |  |  |  |  |
| Winter 2023 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **Journal Project - Abstraction Design Activity** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **Journal** |  |  |  |  |  |  |  |  |
| Responsibility: to hold and display information |  |  |  |  |  |  |  |  |
| Behaviors: adding an entry, displaying all the entries, saving to a file, loading from a file; calls Entry display method   |  | | --- | | **Prompt** | | List<string>: List | | Get(): string |  |  | | --- | |  | |  | |  | | | | | | | | | |
| **Journal** | | |  |  |  |  |  |  |
| \_entries: List<entryDisplay()> | | |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |
| Display(): void | | |  |  |  |  |  |  |
| AddEntry(): void | | |  |  |  |  |  |  |
| SaveToFile(): void | | |  |  |  |  |  |  |
| LoadToFile(): void | | |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **Entry** |  |  |  |  |  |  |  |  |
| Responsibility: hold data |  |  |  |  |  |  |  |  |
| Behaviors: show all Entry objects (prompt, user response, date); needs own display method | | | | | | | | |
| **Entry** | | |  |  |  |  |  |  |
| \_dailyPrompt: List<PromptGenerator> | | |  |  |  |  |  |  |
| \_userResponse: string | | |  |  |  |  |  |  |
| \_dateWritten: string | | |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |
| entryDisplay(): void | | |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **PromptGenerator** |  |  |  |  |  |  |  |  |
| Responsibility: call prompt from sources |  |  |  |  |  |  |  |  |
| Behavior: generate random prompt response |  |  |  |  |  |  |  |  |
| **PromptGenerator** | | |  |  |  |  |  |  |
| \_prompt: string | | |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |
| Display(): void | | |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **SaveToFile** |  |  |  |  |  |  |  |  |
| Responsibility: save all data to specified filename |  |  |  |  |  |  |  |  |
| Behaviors: save all data and allow to be opened again | | | | | | | | |
| **SaveToFile** | | |  |  |  |  |  |  |
| \_filename: string | | |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |
| SaveToFile(): void | | |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **LoadToFile** |  |  |  |  |  |  |  |  |
| Responsibility: load specified filename |  |  |  |  |  |  |  |  |
| Behaviors: open and display all Entry objects from specified filename | | | | | | | | |
| **LoadToFile** | | |  |  |  |  |  |  |
| \_filenames: List<SaveToFile> | | |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |
| LoadToFile(): void | | |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

CSE210Thursday@7pm Team Meeting

Last edited: 7m ago

**Brenton**

class write    
    -Responsibilities   
        -Generate random prompt for user to write about.    
        -Provide storage for journal entries, potentially in a list variable.   
        -Be able to save the journal entries with a timestamp in a local file.    
    -Input   
        -Journal entry

class display   
    -Responsibilities   
        -Iterates through the file and displays the entries for recollection with timestamp.

class load   
    -Responsibilities   
        -Calls for filename and retrieves file for processing.    
    -Input   
        -filename and extension

class save   
    -Responsibilities   
        -Takes text input and save it to local storage.   
    -Input   
        -Directory, filename, and extension

class main   
    -Responsibilities   
        -Display menu options   
        -Receive input and then determine class based on Responsibilities   
        -Defines variables for storage   
    -Input    
        -Integer correlating with menu choices 

**Journal\_Entry Class**

Journal\_Entry Class   
------------------------------------   
 journal\_Entry   
------------------------------------   
        \_date : date   
        \_prompt: string   
        \_response: string   
------------------------------------   
ShowEntry: void   
------------------------------------ 

**Classes:**

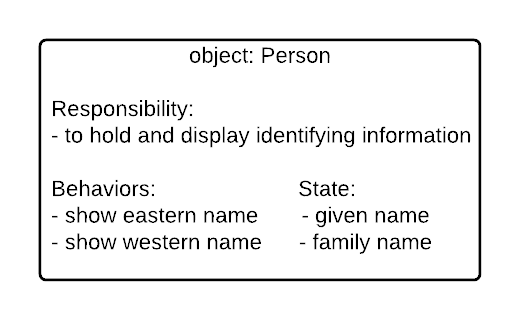
1. Write: prompts to display (5 are required); store prompt given, the user response, and the date
2. Display: all entries with prompt, response and date; calls from the write class
3. Load: ask for filename; loads all entries in the journal
4. Save: ask for filename; should replace all entries stored in memory

**Main:**

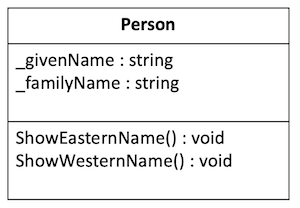
Print menu - get response - decide what class to use

When Write is selected - get returned object to list

template called a class. The object's state is translated to variables called attributes. The object's behaviors are translated to functions called methods.



|  |
| --- |
|  |
|  |
|  |



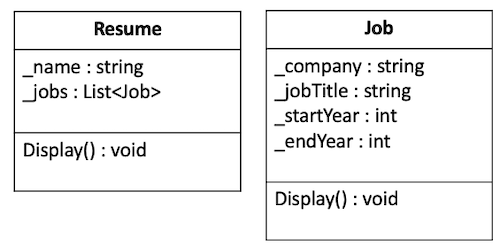
|  |
| --- |
| **Prompt** |
| List<string>: List |
| Get(): string |

prompt class a little different, so it would return the prompt as a string that can be displayed by AddEntry and be saved in the JournalEntry.

Prompt class has a List<string> in it with a bunch of prompts. The Get method returns a random item from that list. So the AddEntry can prompt.Get() a string prompt. Does that make sense?

|  |
| --- |
| **Prompt** |
| List<string>: List |
| Get(): string |

|  |
| --- |
| **Prompt** |
| List<string>: List |
| Get(): string |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classes | |  |  |  | | --- | --- | --- | | 1 pts  Complete  A class diagram is included that defines all classes needed for this assignment. | 0.7 pts  Developing  A class diagram is included that defines at least one class. | 0 pts  Incomplete  A class diagram is not included. | | 1 pts |
| This criterion is linked to a Learning Outcome Journal - Behaviors | |  |  |  | | --- | --- | --- | | 2 pts  Complete  All Journal behaviors needed for this assignment are defined. | 1.4 pts  Developing  Some Journal behaviors are defined. | 0 pts  Incomplete  No Journal behaviors are defined. | | 2 pts |
| This criterion is linked to a Learning Outcome Journal - Attributes | |  |  |  | | --- | --- | --- | | 1 pts  Complete  All Journal attributes needed for this assignment are defined. | 0.7 pts  Developing  Some Journal behaviors are defined. | 0 pts  Incomplete  No Journal behaviors are defined. | | 1 pts |
| This criterion is linked to a Learning Outcome Entry - Behaviors | |  |  |  | | --- | --- | --- | | 2 pts  Complete  All entry behaviors needed for this assignment are defined. | 1.4 pts  Developing  Some entry behaviors are defined. | 0 pts  Incomplete  No entry behaviors are defined. | | 2 pts |
| This criterion is linked to a Learning Outcome Entry - Attributes | |  |  |  | | --- | --- | --- | | 1 pts  Complete  All entry attributes needed for this assignment are defined. | 0.7 pts  Developing  Some entry attributes are defined. | 0 pts  Incomplete  No entry attributes are defined. | | 1 pts |
| This criterion is linked to a Learning Outcome Saving/Loading | |  |  |  | | --- | --- | --- | | 1 pts  Complete  Saving/Loading is defined sufficiently for this assignment. | 0.7 pts  Developing  Saving/Loading is mentioned. | 0 pts  Incomplete  Saving/Loading is not defined. | | 1 pts |
| This criterion is linked to a Learning Outcome Prompt Generation | |  |  |  | | --- | --- | --- | | 1 pts  Complete  Prompt generation is defined sufficiently for this assignment. | 0.7 pts  Developing  Prompt generation is mentioned. | 0 pts  Incomplete  Prompt generation is not defined. | | 1 pts |
| This criterion is linked to a Learning Outcome Interaction | |  |  | | --- | --- | | 1 pts  Complete  A diagram or description explains the interaction among the elements of the design. | 0 pts  Incomplete  No interaction is explained. | | 1 pts |