1. What does the program do?

The program helps the users in journaling. It helps them in recording their experiences. It provides them with prompts for guidance. It can also be used for viewing previously written journal entries.

1. What user inputs does it have?

The journal program will have the following user inputs:

* Writing a new journal entry, as well as responding to a prompt.
* Displaying the journal entries.
* Saving the journal to a file.
* Loading a journal from a file.
* Displaying the above menu options.

1. What output does it produce?

It produces the following outputs:

* Displaying the prompts.
* Displaying the journal entries.
* Saving as well as loading a journal.
* Displaying the menu options for the user to select from.

1. How does the program end?

The program can end anytime the user wishes. He can choose to exit the application before or after completing the desired tasks.

1. What are good candidates for classes in this program?

The classes for this program are:

* Journal
* Entry
* PromptGenerator

1. What are the primary responsibilities of each class?

* Journal- This class will manage entries (i.e., perform functions like displaying entries, saving as well as loading entries, to and from files respectively.
* Entry- This class will contain the individual journal entries, along with the user’s responses to the prompts, and the dates on these entries.
* PromptGenerator- This class will generate random prompts for the new journal entries.

1. What are the behaviours this class will have in order to fulfil its responsibilities? (In other words, what things should this class *do*?)

The behaviours of the various classes are:

* Journal- It should have methods for adding and displaying entries, and, also for saving as well as loading entries to and from a file, respectively.
* Entry- It should have a method for displaying the entry.
* PromptGenerator- It should have a method for generating random prompts.

1. What attributes does this class need to fulfil its behaviours? (In other words, what variables should this class *store*?)

* Journal needs an attribute to store a list of Entry objects.
* Entry needs attributes to store the prompt, response, and date.
* PromptGenerator may not need specific attributes for this simplified design.

1. What are the data types of these member variables?

* Journal- It should have an attribute of type List<Entry> for storing entries.
* Entry- It should have attributes like string for prompts and responses and DateTime for dates.

|  |
| --- |
| **Journal** |
| \_entries: List<Entry> |
| \_writeEntry(prompt: Prompt, response: string)  \_displayJournal( )  \_saveJournal(filename: string)  \_loadJournal(filename: string)  \_displayMenu( ) |

|  |
| --- |
| **Entry** |
| \_prompt: string  \_response: string  \_date: DateTime |
| Display( ): void |

|  |
| --- |
| **PromptGenerator** |
| \_generatePrompt( ): string |

In the above diagram the main classes and their relationships in the program is described. The Journal class manages entries, the Entry class represents individual entries, and the PromptGenerator class generates prompts. The program’s functionality is driven by the methods within these classes. The Journal class, for instance, handles writing new entries, displaying the journal, and saving/loading entries to/from files.