**What is abstraction and why is it important?**

* **Explain the meaning of Abstraction**

This week I learned through the exercises that abstration is a form of coding that simplify certain tasks a programmer would have when writing a program.

* **Highlight a benefit of Abstraction**

One of benefits of abstraction that I can think of is how easy is to fix code errors. A program that has a variety of tasks, by using abstraction and assigning every single task to a class, the program becomes simple to approach and easy to read and if there will be an error, we can solve the error without having to change the whole code but only the class of the task that has the error, and if will no affect anywhere else in the program.

* **Provide an application of Abstraction**

A clear example of an abstraction is the program I wrote that helps users to write their journal in a simple way and in order to write the program I divided it in five (5) objects.

* **Use a code example of Abstraction from the program you wrote**

To put that in code we can divide it in 5 classes of code:

Public class Journal

Public class PromptGenerator

Public class Menu

Public class LuchyPhases

Public class Entry

This division made the program easier to write and to debug as well.

* **Thoroughly explain these concepts (this likely cannot be done in less than 100 words)**

Then after dividing into classes, we can now write its purposes:

Class Journal – this class holds the structure of the Journal it has a member variable with custom Data type that gets in all the entries and then through some member functions it can Display in a set pattern, save into a file or load a file with past information.

Class PromptGenerator – This simple function holds a member list of strings that has prompt phrases and through a member function Called GetRandomPrompt, it returns only one prompt randomly.

Class Menu – The menu class holds a list of options that then through a member function it allows this class to focus only on displaying the menu option for the users

Class LuchyPhases – just like the PromptGenerator class, LuchyPhases class holds a number of phrases and through a member function it returns a random phrase with the user’s name in it.

Class Entry – This class is uncharged with all the entries collecting through some member classes the date, the prompts, the user text input and the lucky phrase and trough a member function it displays the information in a set way.

With all these functions working well separately then in the Program file we can put then to work together in just one program.