CSE210 Programming with Classes

Jacob Chan Ka Shing

Encapsulation Articulate

Due Date: 22 Oct, 2023

Meaning of encapsulation:

The meaning of encapsulation in C# is grouping up some data and code into a class. Class is like a capsule or container to contain functions in the program. When data, information and functions are wrapped into a class, the details will be hide. This will play a protective role to the code.

Benefit:

One primary benefit is data hiding. When we use ‘private’ and ‘public’, the data will be controlled by us so that we can control who can access the data from a class. ‘public’ means the public members can access the data. ‘private’ means private members can access the code in the class. Unless it is necessary to show the code, it is a good habit not to expose the inside variables and data. That can help secure our data.

An application example:

namespace Demo

{

class PersonalInfo

{

public string Name;

public int Age;

public string Gender;

public string Address;

}

}

The above is a simple encapsulation example. The data Name, Age, Gender and Address are wrapped by a class called PersonalInfo.

Code wrote by me:

using System;

public class PromptGenerator

{

private static string[] \_randomQuestions =

{

"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?",

"Who was the most interesting person I interacted with today?",

"What did you eat today?",

"What do you want to improve today if you can do it again?",

"Where have you been today?",

"How was the weather today?"

};

public static string GetRandomQuestion()

{

var rand = new Random();

string randomQuestion = \_randomQuestions [rand.Next(\_randomQuestions.Length)];

return randomQuestion;

}

}

Principle explanation:

The above code was written by me from the last assignment in the Journal program. This is a prompt generator to generate a prompt randomly. The random prompt question finally saves to the public static string GetRandomQuestion() and is all wrapped by a class PromptGenerator. When the main cs file accesses this function, just need to call the class, so that all the long data of prompt questions no need to show in the main cs file. The advantage is to maintain the main program organized and neat.