Assignment03-Q3:

PhoneNumber
- phoneNumber: String
+ PhoneNumber(number: String) - alphaToNumerical(digit: String): String + getNumber(): String

Assignment03-Q5:

ComputerClient
- guess: int - correct: Boolean
+ ComputerClient() - generateGuess(): int

The ComputerClient class has two private instance variables called guess and correct. The guess variable stores the computer's randomly generated guess, and the correct variable is whether or not the guess was correct. The ComputerClient method would have a while loop that could run as long the correct variable is false. The method would generate a guess using the generateGuess method, and then call the play method of the GuessingGame class. The compare class would return whether the guess was correct or not, and if it was correct or if the limit of tries has been reached, the while loop will terminate.

GuessingGame

- random: int - tries: int
+ play(guess: int): Boolean - countTries(): Boolean

The GuessingGame class would consist of the random instance variable, which would be the random integer that is trying to be guessed by the computer client. The play method would be called by the ComputerClient class. If the guess is correct, the play method would return true to the ComputerClient class. If the guess isn’t correct, then the play method would call the countTries method, which would keep a count of the number of tries taken, and would return true if the limit has not been reached. If the limit has been reached, then it would output false, and play would output that the limit has been reached and return true in order to signal to the ComputerClient class that the game has ended.

Assignment03-Q6:

Encapsulation Principle: The encapsulation principle is applied through the use of private instance variables for the GuessingGame class and the ComputerClient class. For the GuessingGame class, this allows each method within the class to have access to the information it needs, while preventing other classes from accessing the same information. The benefit of utilizing the Encapsulation Principle is that data can stay safe and private while also ensuring that each method that needs the data has access to it.

Information Hiding Principle: The Information Hiding Principle is applied through the use of private methods and variables. The ComputerClient class has no access to the countTries method or the instance variable tries, so it can not directly change that variable. It also has no access to the variable it is trying to guess. The GuessingGame class has no access to the guess of the ComputerClient before the ComputerClient passes the guess into the GuessingGame class. The benefit of applying this principle is that information will stay safe and private, and there will be clearly defined uses for each method.

Interface Principle: The ComputerClient interacts with the GuessingGame class using the play method, which is a public method. This is the only method it has access to. It has no access to the countTries method, which is a private method. The benefit of the Interface Principle is that there are clear ways for classes to communicate, which allows them to better implement the Information Hiding Principle and the Encapsulation Principle.