GUESS The Number Game Class

For this assignment you are asked to write a class which allows you to create a game object to play some specified number of rounds of the random number guessing game. Essentially you are being asked to convert the random number class from last week's practicum into an object class. Given the main method provided, your game should work as follows:

```
In this game, you will be playing 3 rounds!
Get ready to play round 1!
Enter your guess: 57
Too high
Enter your guess: 49
Too Low!
Enter your guess: 50
got it! You guessed it in 3 guesses!!

Get ready to play round 2!
Enter your guess: 76
Too high
Enter your guess: 67
Too Low!
Enter your guess: 70
Too Low!
Enter your guess: 73
Too Low!
Enter your guess: 74
Too Low!
Enter your guess: 76
Too high
Enter your guess: 75
got it! You guessed it in 7 guesses!!

Get ready to play round 3!
Enter your guess: 27
Too Low!
Enter your guess: 37
Too Low!
Enter your guess: 67
Too high
Enter your guess: 66
got it! You guessed it in 4 guesses!!

In this game, you will be playing 2 rounds!
Get ready to play round 1!
Enter your guess: 87
Too high
Enter your guess: 77
Too high
Enter your guess: 67
Too high
Enter your guess: 54
Too high
Enter your guess: 47
Too high
Enter your guess: 27
Too Low!
Enter your guess: 37
Too high
Enter your guess: 33
got it! You guessed it in 8 guesses!!

Get ready to play round 2!
Enter your guess: 44
Too high
Enter your guess: 33
Too high
Enter your guess: 28
got it! You guessed it in 3 guesses!!

In this game, you will be playing 1 round!
Get ready to play round 1!
Enter your guess: 98
Too high
Enter your guess: 54
got it! You guessed it in 2 guesses!

The least number of guesses was 2 and the greatest number of guesses was 8!
```

```
public static void main( String[] args ) {

   // Start the first game, default 3 rounds to be played
   GuessMyNumber game = new GuessMyNumber();
   game.playGame();

   // Start a second game, 2 rounds to be played
   game = new GuessMyNumber(2);
   game.playGame();

   // Start a third game, 1 round to be played
   game = new GuessMyNumber(1);
   game.playGame();

   // We should be able to see the
   // LEAST and MOST number of guesses
   // of all the three games played here

}
```

Name:_____ **BU Email:** _____

Note: Be sure to follow proper programming standards including proper use of comments, descriptive variable names and consistent alignment and indentation techniques as presented in lecture1

GUESS The Number Game Class

As you can see from the main method, each instance (i.e. object) of the game should allow for *some number of rounds* of the guessing game to be played. The number of rounds to be played is determined when each instance of the game is created. If the number of rounds is not specified when the game is created, some default value can be assigned to the number of rounds to be played.

Each *round* played in the game should give the player unlimited attempts to guess a number which is randomly generated for that round. You do not need to worry about how to generate the random number, a method to do so is provided with the skeleton code.

With each guess the user enters, your program should inform them if the guess is too high or too low and continue until the correct number is entered.

Each *game* should maintain a list of the random number generated for each round as well as the number of attempts it took the player to guess that number. Additionally, you need to keep track of the greatest ***and*** the least number of guesses it took to guess the number of all the **games** played.

Using the skeleton code provided, complete the class called GuessMyNumber that allows this guessing game to be played following the rules specified and shown in the example above. Your code should employ *proper use of encapsulation* when declaring the necessary data members and defining the required methods.

The skeleton code for your class is provided on the next page. *You are asked to fill in the:*

- **fields (i.e. data members) needed** to accomplish the objectives stated above.
- **required constructor(s)** *to allow different instances of the game to be created (as shown in the main method provided above).*
- **the following two instance methods:**
    - `public void playGame()`
      this method controls the number of rounds of the game to be played by calling the playRound method (as many times as needed) and passing to the method the required inputs.
    - `public int playRound(Scanner scan, int numToGuess, int round)`
      this method gives the user unlimited attempts to guess the number passed (by playGame) to the parameter numToGuess.

Name:_____ **BU Email:** _____

Note: Be sure to follow proper programming standards including proper use of comments, descriptive variable names and consistent alignment and indentation techniques as presented in lecture

GUESS The Number Game Class

```
public class GuessMyNumber {
    public static final int MIN = 1;    // MIN and MAX is the specified range
    public static final int MAX = 100;  // for the random number

    // define remaining data members as needed









    // You can assume this method works and generates a random number between min/max
    private static int randomNumber(int min, int max) {
        return( // some random number );
    }

    // define the required Constructor(s)
```

Name:_____ **BU Email:** _____

Note: Be sure to follow proper programming standards including proper use of comments, descriptive variable names and consistent alignment and indentation techniques as presented in lecture

GUESS The Number Game Class

```
    // Complete the method playGame()
    //
    // This method allows the user to play the required number of rounds
    // of the guessing game, generating a new number for each round.
    //
    public void playGame() {
        Scanner scan = new Scanner( System.in );
```

```
}
```

Name:_____ BU Email: _____

Note: Be sure to follow proper programming standards including proper use of comments, descriptive variable names and consistent alignment and indentation techniques as presented in lecture

GUESS The Number Game Class

```
    // complete the method playRound()
    //
    // This method should give the player unlimited attempts to guess the number in
    // numToGuess which is passed to the method. Once the player enters the
    // correct guess, no more guesses are needed, and the method should return
    // the number of attempts the player made in that round to guess the number.
    //
    // With each guess that is entered, your method should tell the player if the
    // guess is too high or too low until the correct number is entered.
    //
    public int playRound( Scanner scan, int numToGuess, int round ) {

        System.out.println( "Get ready to play round " + round + "!" );









    }
} // end of class
```

Name:_____ **BU Email:** _____

Note: Be sure to follow proper programming standards including proper use of comments, descriptive variable names and consistent alignment and indentation techniques as presented in lecture

Extra Page

Name:_____ BU Email: _____

Note: Be sure to follow proper programming standards including proper use of comments, descriptive variable names and consistent alignment and indentation techniques as presented in lecture