In this lab, you will practice using loops and also the Random class.

**General Instructions:**

1. Prepare your lab report in an essay format, answering every numbered question in the lab instructions below.

2. The report should have a cover page containing your name, the lab number, the date of the lab and the date that the report is submitted.

3. Evaluation will depend on the clarity of your writing, the correctness of your answers, and the adherence to the lab instructions.

Please submit your lab report by email to walkerel@hiram.edu by the deadline. Unless you have difficulty with the email submission, do not submit a hard copy in class. In addition to the written report (.doc file or equivalent), you should submit zip archives of all BlueJ projects that you create in this lab.

**Lab Instructions:**

1. Consider the following code fragments. What will each code fragment print? Write your answer before running the code. Check your answer by copying this code into the main() method of a java project. Since you cannot declare the same variable twice in the same method, you will need to remove “int” before num=20 in the second method. Run your code. In your laboratory report, indicate whether your initial guess for what would be printed was correct in each case. Find initialization values for num and max for which the result is identical for the two loops, and another set of initialization values for num and max for which the result is different for the two loops. Explain the difference.

**Fragment 1:**

```java
int num = 20, max = 20;
while (num < max)
{
    System.out.println(num);
    num +=4;
}
```
Fragment 2:

```java
int num = 20, max = 20;
do {
    System.out.println(num);
    num +=4;
} while (num < max);
```

2. Transform the following code fragment first into two equivalent code fragments. The first equivalent code fragment uses a do–while loop, second equivalent code fragment uses a for loop. Make sure that in both cases the code fragment produces the same input as the one produced by the code fragment which uses the while loop construct. Finally, discuss which of the previous fragments you would choose and why or when.

```java
int num = 1;
while (num < 20) {
    num++;
    System.out.println(num);
}
```

Which value of `num` causes the loop to terminate?

3. The “real” rules for Pig Latin don’t only move the first character to the end of the word, but actually move all the leading consonants together to the end of the word. So while “apple” still becomes “appleay” and “cat” still becomes “atcay”, “this” should really translate to “isthay” not “histay” as you did in lab 3. Modify your old Pig Latin program so that it correctly translates words that have more than one consonant before the first vowel. Be sure to carefully document your program. Write about the design, implementation, and testing in your lab report.

4. Your final program for this lab is to create a simple class that implements a guessing game. The class should have one constructor, which takes a single integer (the maximum value to guess). When the game is played (by calling a method `Play()` that takes no parameters) the following happens: A random number between 1 and max is generated for the user to guess. (This number is not printed, just saved in a variable). The user is asked to guess a number. If the number guessed is correct, then `Play` should return the number of guesses. If the number guessed is null (the player clicked Cancel), the user should get a “Thanks for trying” message, and `Play` should return -1. If the number is too high, the player should get a message telling them that the number is too high and should get to try again. If the number is too low, the player should get a message telling them that the number is too low, and they should get to try again. Feel free to write smaller methods for `Play()` to call to do part of its work for it. Be sure to completely document your program. In your lab report, discuss the design of your program, why you made it the way you did, and any problems or issues that you ran into while implementing it.
5. Write a tester class for your guessing. This class should have a main function that lets the user play the game repeatedly. After each game, the user should receive a message telling them how they did (like “Congratulations! You guessed the number in only 7 guesses!”) or an appropriate message if they quit the game instead of finishing it. (We will take off points if your program writes “Congratulations! You guessed the number in only –1 guesses!”) Then the user should be asked if they want to play again, and as long as they say “Yes” or “yes”, then they should get to play again. Discuss the design, implementation and testing of this class in your report.