Homework 3

Due: Oct. 3, 11:59 PM

Name:

Student ID:

Submission Instructions: This PDF contains fillable fields where you can input your answers. (For example, you can input your name and ID in the fields above.) Please save the document regularly so your answers are not lost. After you complete the assignment, upload a copy to Gradescope. A link to Gradescope can be found on the left side of the course Canvas page.

Important Note on Academic Integrity: This assignment should be completed individually. In recent semesters, improper collaboration on homework has led to multiple cases of plagiarism, where we receive identical or nearly identical submissions from two or more students. If you decide to discuss this assignment with other students before the deadline, make sure you first read the section of the syllabus on proper and improper collaboration. Additionally, you must include the names of these students below.

Names of Collaborators (if any):

Question 1 (15 points; 3 per part): Trace the execution of the loops below by filling in the table to the right of each code fragment. If a loop executes infinitely, trace at least three iterations and then write "infinite loop" on the next row. Be sure to store the initial and final values in the table. Note that the tables may contain more rows than you need.

Example:
```
int count = 1;
while (count > 0)
{
    count = count + 1;
}
```

count
1
2
3
4
inf. loop

a)
```
int total = 0;
int count = 1;
while (count <= 3)
{
    total = total + 2 * count;
    count = count + 1;
}
```

total	count

b)
```
total = 0;
int count1 = 5;
int count2 = 0;
while (count1 > count2)
{
    total = count1 + count2;
    count1 = count1 - 1;
    count2 = count2 + 2;
}
```

total	count1	count2

c)
```
int total = 0;
for (int x = 1; x != 18; x = x + 6)
{
    total = total - x;
}
```

total	x

d)
```
int cycle = 0;
for (int idx = 5; idx > 0; --idx)
{
    cycle = (cycle + 1) % 3;
}
```

cycle	idx

e)
```
int huge = 54321;
for (int count = 0; count > 0; ++count)
{
    huge = huge * huge * huge;
}
```

huge	count

Question 2 (10 points): The program shown below has comments at the end of some lines. On these lines, there is a variable that also appears in a row of the table under the program. For each row, fill in the empty columns. See the first row of the table for an example.

In the first empty column, write the role that the variable plays in the program. There are three possible answers: local variable, parameter, or argument. Note that any variable that is passed to a method as an argument will also be either a local variable or parameter of the calling method. For example, consider the following method:

```
public static int exampleMethod(int param)                    // Line 1
{
    int localVar = 42;                                        // Line 2
    return Math.max(param, localVar);                         // Line 3
}
```

On line 1, the variable param is a parameter of the method exampleMethod. On line 2, the variable localVar is a local variable. On line 3, both param and localVar are passed to the method Math.max as arguments. Thus, although the variables param and localVar are a parameter and local variable respectively, on line 3 they both play the role of arguments.

In the second empty column, give the scope of the variable. The scope is the part of the program where the variable can be used. If the variable can only be used in a loop, write the type of loop (e.g., “for loop”). If it can only be used in a conditional statement, write “conditional.” Otherwise, write the name of the method where the variable can be used. In the method above, for example, the scope of both param and localVar are the method exampleMethod.

```
import java.util.Scanner;

public class WordGame
{
    public static void main(String[] args)                    // Line 1
    {
        Scanner keyboard = new Scanner(System.in);            // Line 2

        String playAgain = "YES";
        while (playAgain.equalsIgnoreCase("YES"))
        {
            boolean won = playTheGame(keyboard);              // Line 3
            if (won)
                System.out.println("You word guesser won!");
            else
                System.out.println("The word guesser lost!");

            System.out.println("Do you want to play again?");
            playAgain = keyboard.nextLine();
        }
    }
```

```
    public static boolean playTheGame(Scanner keyboard)              // Line 4
    {
        String word;        // word to be guessed
        String guess = "";  // current guess

        System.out.println("Enter a word for your opponent to guess");
        word = keyboard.nextLine();                                  // Line 5

        String shuffled = shuffle(word);

        // Show the shuffled word one character at a time
        for (int index = 1; index < shuffled.length() + 1            // Line 6
                && guess.equalsIgnoreCase(word) == false; ++index)
        {
            // This shows one more character each time through
            System.out.println("Some characters are: " +
                    shuffled.substring(0, index));                   // Line 7
            System.out.println("What is your guess?");
            guess = keyboard.nextLine();                             // Line 8
        }

        // If the guess is the word, the guesser won
        return guess.equalsIgnoreCase(word);
    }

    public static String shuffle(String letters)
    {
        // StringBuilders are like Strings,except we can edit characters
        StringBuilder result = new StringBuilder(letters);           // Line 9

        // for each letter, swap it with a randomly chosen letter
        for (int index = 0; index < result.length(); ++index)
        {
            // Find the random location--all parentheses are necessary
            int rand = (int) (Math.random() * result.length());

            // Find the letter at the next location
            char c = result.charAt(index);                           // Line 10

            // Swap the letter at the random location with the next one
            result.setCharAt(index, result.charAt(rand));
            result.setCharAt(rand,  c);                              // Line 11
        }

        // Turn the StringBuidler back into a String object
        return result.toString();
    }
}
```

Line #	Identifier	Role	Scope
1	`args`	parameter	main method
2	`keyboard`		
3	`keyboard`		
4	`keyboard`		
5	`keyboard`		
6	`index`		
7	`index`		
8	`keyboard`		
9	`letters`		
10	`c`		
11	`c`		

Question 3 (10 points; 2 per part): The first table below shows the signatures of an overloaded method. For each row in the second table, determine which of the methods in the first table is actually called. If it is impossible to determine the method, write "impossible." If the assignment is illegal, write "illegal."

Label	Method Signature
A	`void mix(int color1, int color2)`
B	`void mix(int color1, int color2, boolean opaque)`
C	`int mix(int color1, int color2)`
D	`int mix(int color1, double color2)`
E	`double mix(double color1, int color2)`
F	`double mix(double color1, double color2)`

Method Call	Label (A to F) or Impossible or Illegal
`mix(127, 63, true);`	
`int result = mix(127, 63);`	
`int result = mix(127.0, 63);`	
`double result = mix(127, 63.0);`	
`double result = mix(127.0, 63.0);`	

Question 4 (15 points; 3 per part): Write the signature for a method that will perform each of the computations described below. **You do not need to write the body of the method.**

Example: Find the average of two grades given as integers.

```
double average(int grade1, int grade2)
```

a) Generate a random integer that is greater than a given double.

b) Determine if two characters are equal.

c) Perform the logical AND operation.

d) Print a given message to the console.

e) Find the sum of a sequence of double values entered from the keyboard that ends with a given sentinel value. (Note: the Scanner used to read the values is not constructed in the method.)