Introduction to Computer Science 1 – Homework 4 (Loops)

1. (On paper) Write down the execution trace of the following program. Write down all the variables and their intermediate values. Also clearly present what will be printed.

```
public class OddStuff {
public static void main(String[] args) {
     int number = 32;
     for (int count = 1; count <= number; count++) {
          System.out.println(number);
          number = number / 2;
     }
  }
}
```

2. (On paper) What is the output of this code (what is printed)? How many times is the while code executed (i.e. how many loops we have)? What is the value of x after this code executes?

```
int x = 1;
System.out.print(x);
while (x < 100) {
    x = x + x;
    System.out.print(", " + x);
}
```

3. (On paper) Below you can see the code of a method called mystery. What is the type of the method? What are its parameters and their type(s)? What is printed if I call `mystery(6)`? What about `mystery(1)`?

```
public static void mystery(int x) {
    int y = 1;
    int z = 0;
    while (2 * y <= x) {
        y = y * 2;
        z++;
    }
    System.out.println(y + " " + z);
}
```

4. (On paper, Reminder!) Find the syntax errors in the following program.

```
public class Parameters {
    public static void main() {
        double bubble = 867.5309;
        double x = 10.01;
        printer(double x, double y);
        printer(x);
        printer("barack", "obama");
        System.out.println("z = " + z);
    }

    public static void printer(x, y double) {
        int z = 5;
        System.out.println("x = " + double x + " and y = " + y);
        System.out.println("The value from main is: " + bubble);
    }
}
```

5. (We normally cover this during lecture. If you were not there, do it now). Formulate correct for loops (or while or do-while) that will print the following sequences of numbers. We did this in class (if you were not in, try it now):
a. 1,2,3,4,…,1000
b. 2,4,6,8,10,…100
c. 50,45,40,…,-5

6. Write a for loop that produces the song *Bottles of Beer on the Wall*:

```
10 bottles of beer on the wall, 10 bottles of beer
Take one down, pass it around, 9 bottles of beer on the wall
9 bottles of beer on the wall, 9 bottles of beer
Take one down, pass it around, 8 bottles of beer on the wall  ...
…
1 bottles of beer on the wall, 1 bottles of beer
take one down, pass it around, 0 bottles of beer on the wall
```

7. Validation loop is called a loop that keeps asking the user to enter a number until that is within a valid range (it can be a number within a range or a number in specific format or a specific character, etc.). For example, can you write a validation loop that will prompt the user to give a number between 0 and 12 (included)? That means that if the user gives a number outside this range, then the loop should keep asking the user for a new number.

8. Write a method `isSingleDigit` that accepts an integer and returns true if that integer has only one digit or false if that integer has more digits. Pay attention to the type of the method (i.e. what it returns) and what are its parameters (i.e. what it takes as input). How would you call that method in your main method? Try it with different examples.

9. Write a method named `printLetters` that takes a String as its parameter and that prints the letters of the String, separated by dashes. For example, the call of `printLetters("Rabbit")` should print:

```
R-a-b-b-i-t
```

Hint: You need a loop to go over all the characters of the String. How do you know how many characters the string has? How do you get each character? If you don't remember, check String API for methods on Strings.

10. How do I print the following pattern using nested loops?

```
*******
*****
***
*
```

11. What is the output of this program?

```
public class StrangeStuff {
public static void main(String[] args) {
    for (int i = 1; i <= 10; i++) {
        for (int j = 1; j <= i; j++) {
            System.out.print("*");
        }
        for (int j = 1; j <= 20 - 2 * i; j++) {
            System.out.print(" ");
        }
        for (int j = 1; j <= i; j++) {
            System.out.print("*");
        }
        System.out.println();
    }}
```

12. Write a method named `printGrid` that accepts two integers representing a number of rows and columns and prints a grid of integers from 1 to (rows*columns) in column-major order.
For example, the call: `printGrid(4, 6);`
should produce the following output:

```
1 5 9 13 17 21
2 6 10 14 18 22
3 7 11 15 19 23
4 8 12 16 20 24
```

13. Write a method named `repl` that accepts a String and a number of repetitions as parameters and returns the String concatenated that many times. For example, the call `repl("die",3)` returns `"diediedie"`. If the number of repetitions is 0 or less, an empty string is returned.

14. Write a program that first asks the user for a word and then asks for a number, and prints out the word as many times as indicated by the number.

15. Read in an unspecified number of integers, typed in on 1 line, separated by spaces and terminated with a letter. When the line ends, write out the maximum and minimum of the numbers.

16. Write a Java program that prints a pyramid like this:

```
    []
  [][][]
[][][][][]
```

Let your program ask the user for the height of the pyramid.
(*Hint:* you will need multiple nested loops for this exercise.)

If you are having difficulties, try having your program print a staircase first:

```
[]
[][]
[][][]
```

then adapt it to print the pyramid.

17. Now that you know how to use loops, return to *Exercise 5* from module 3 an implement, this time, a completely customizable dice. Write a method that receives as input parameter an integer and use this number to calculate the probabilities per combination and to allow any (in the range of an `int` variable) number of sides dynamically. You will need to use this value in the for loop.