CS 5000 – Spring 2020
Assignment #1

Simple Java Programs – Chapters 1 and 2

Develop a complete Java program for each of the following problems. Please name the programs as indicated and add proper program headers and output labels as shown below. ***Please use <u>only</u> concepts and programming constructs/syntax we discuss to date.***

<u>Program #1 (10 points):</u> Create a new Java program named `HelloWorld` and type the following code. Compile the code and run it. Make sure you **replace the dots in the header section (red comments) with your full name and the name of your IDE, such as Jgrasp.**

```
// Class:      CS 5000
// Term:       Spring 2020
// Name:       ...
// Instructor: Dr. Haddad
// Assignment: 1
// IDE Name:    ...

import java.util.Scanner; //import scanner class

public class HelloWorld
{
    public static void main(String args[]) //program main method
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Please enter your name: "); //prompt user for input
        String name = s.nextLine();  //read user input
        System.out.println("Hello " + name + "!"); //print out your message
    }
}
```

<u>Program #2 (10 points):</u> Write a java program to display the following pattern three letters (B E X) using Java *print* statements. Name the program *Letter*. Compile the code and run it. Make sure you add program header similar to that in program #1 above.

```
BBBB    EEEEE  X   X
B   B   E       X X
BBBB    EEEE     X
B   B   E       X X
BBBB    EEEEE  X   X
```

<u>Program #3 (10 points):</u> Write a java program to read from the user an integer number (as numeric value, using integer variable and method `nextInt()` of the scanner class). Limit the input value to be between 0 and 9999. Then display the sum of the digit in the number (**Do not use loops or treat the entered number as string of characters**). Manipulate the input number **<u>mathematically</u>** using proper math operators (division and remainder) to determine the sum of its digits. For example, if the user enters the integer value1234, the program would determine the sum and display the output as shown below. Sample test data below does not show the input prompts.

<u>First test:</u>

```
The input number is:   1234
The sum of digits is:  10
```

<u>Second test:</u>

```
The input number is:   1111
The sum of digits is:  4
```

Third test:

```
The input number is:   2233
The sum of digits is:  10
```

Name the program *SumDigits*. Again, limit the input integer to no more than 4 digits. Make sure the program has a proper header and inline comments, see program #1 above. Use proper labels for the input prompt message (such as *Please enter 4-digits number:*), and label and line-up the outputs as shown above. Use tabs (\t) to line-up the outputs after the labels.

Program #4 (10 points): Write a java program to read the coordinates of 2 points, (x1,y1) and (x2,y2), as double values, and then compute and display the distance between the points using the following formula:

Distance = Sqrt [(x2-x1)^2 + (y2-y1)^2]

Note: The square root function in Java is **`MATH.sqrt(…)`** and the power function is **`MATH.pow(…)`**.

Name the program *Distance*. Make sure the program has a proper header and inline comments, see program #1 above. Use proper labels for the input prompts; and for the output as shown below. Sample test data below does not show the input prompts.

Test data:

```
First point coordinates:      (2.0, 2.0)
Second point coordinates:     (3.0, 3.0)
The distance is:              1.4142135623730951
```

Program #5 (10 points): Write a java program to compute the driving cost of a road trip. Prompt the user to enter the distance to drive (in miles) as double type value, the fuel efficiency of the car (mpg) as double type value, and the price per gallon (price) as double type value. Then the program computes the cost of the trip and displays the output as shown below. Sample test data below does not show the input prompts.

Test data:

```
The distance (miles):         100.0
Fuel efficiency (mpg):        20.0
Price per gallon (dollars):   $2.00
The trip cost (dollars):      $10.00
```

Name the program *DrivingCost*. Make sure the program has a proper header and inline comments, see program #1 above. Use proper labels for the inputs prompts; and use outputs labels as shown above. Use tab (\t) characters to line-up the outputs after the labels as shown above.

Submission:

1. Before submitting your programs, make sure you review the assignment submission requirements and grading guidelines on the course webpage. The grading guidelines explain some of the common errors found in programming assignments.
2. The assignment is due no later than **5:00pm** on the due day posted in D2L.
3. Please compile and run your java files (only the .java files) right before you upload to the assignment submission folder in D2L.