Lab Assignment Number 3
CS1140

In this lab we will again make use of simple IO (input/output) and arithmetic expressions to create simple applications. It never hurts to review your previous labs in preparation for doing today's lab. Create only 1 project called **Lab3** and include the required classes for each part described below.

PART 1 – Equation Tester

Create a Java application called **EqTester** that will allow you to test the values for a specific expression. Assume **integer** inputs and **integer** results.

The expression you are to test is as follows:

$$\frac{4a-2}{a(b+1)} + 4b(ac - b) - \frac{2a + 3 - b}{b + c}$$

Prompt the user to enter the values for a,b and c then compute the expression and display the result.

Example Run:

Equation Tester

```
Enter the value for the variable a: 2
Enter the value for the variable b: 3
Enter the value for the variable c: 4

The value of the expression is    : 60
```

PART 2 – What's my speed

This past year you have started cycling and keeping track of your results. You would like to know how fast you were going in meters per second, unfortunately the cheap bike computer you bought only gives you the total distance you travelled and the time it took to complete your ride. Create a Java application called **Speedy** that lets you calculate your speed as m/sec. **Speed needs to be displayed with only two digits after the decimal point.**

Hint - there is a slide in the notes that shows how to reduce the number of digits

Example Run:

```
Bike computer helper

Enter the number of meters travelled   :  25000
Enter the number of hours you took     :  1
Enter the number of minutes you took   :  12
Enter the number of seconds you took   :  48
Your average speed was                 :  5.72 m/sec
```

PART 3 – Fun with numbers

Just for fun you have decided to write a program that will let you calculate the sum of numbers in the following sequence:

sum = n + nn + nnn + nnnn ⋯ Where n is the number that you enter.

In other words, if you enter a single digit number 5 then you need to find the sum of : 5 + 55 + 555 + 5555

Create a Java application called **FunNums**

Example Run:

```
Fun with numbers

Enter a single digit number : 5

The sum of the sequence is  : 6170
```

Up until this point, the only way to get data to your program was by interacting with the user via the console window (where your output appears). The following example program will show you how to read data from a file rather than from the keyboard. Download the txt file **Lab3_example.txt** to your desktop. Create a Java application called **Lab3_example** and copy the code from the text file. Next download the data file **Lab3_ExampleData.txt** into your project folder. Run the program. You should be able to trace the program and figure out what is happening. I have highlighted two lines of code which you will need to incorporate into your own code when you wish to read data from a text file.

```
import java.util.Scanner;
import java.io.File;

public class Lab3_example {

public static void main( String args[] ) throws Exception {

// create Scanner object to obtain input from file
Scanner inputFile = new Scanner( new File ("Lab3_ExampleData.txt") );

int a,b,c,d;    // reserve space for variables

a = inputFile.nextInt();   // get the number from file
b = inputFile.nextInt();   // get the number from file
inputFile.nextLine();      // skip the rest of the line from file

c = inputFile.nextInt();   // get the number from file
inputFile.nextLine();      // skip the rest of the line from file
inputFile.nextLine();      // skip the rest of the line from file

d = inputFile.nextInt();   // get the number from file
d = inputFile.nextInt();   // get the number from file

System.out.println( "First  number: " + a);
System.out.println( "Second number: " + b);
System.out.println( "Third number : " + c);
System.out.println( "Fourth number: " + d);

} // end method main
} // end of the class
```

Your output should be as follows:

```
First  number : 13
Second number : 5
Third  number : 52
Fourth number : 78
```

PART 4 – Pay Check

Now that you know how to use a data file with your program, I want you to create a Java application called **PayCheck** that will simulate a simple weekly payroll program. The program will read data from an employee data file which contains the data for a single employee. The data file will contain 8 numbers, one per line. The first line is the wage rate per hour whereas the next 7 numbers (also one per line) are the number of hours worked for each of the 7 days of the week.

Write a program, which will make use of data in a file to compute a pay stub for that employee. Your program should read data from a file and print out the pay stub to the console. When you print out the salaries don't worry about the number of decimal places in your output.

Example Run:

```
Employee Paystub

Hourly wage                : 12.50

Hours worked on Monday     : 4
Hours worked on Tuesday    : 8
Hours worked on Wednesday  : 6.5
Hours worked on Thursday   : 0
Hours worked on Friday     : 0
Hours worked on Saturday   : 8
Hours worked on Sunday     : 4.5

Total Hours worked         : 31.0

Monday    earnings         : $ 50.00
Tuesday   earnings         : $100.00
Wednesday earnings         : $ 81.25
Thursday  earnings         : $  0.00
Friday    earnings         : $  0.00
Saturday  earnings         : $100.00
Sunday    earnings         : $ 56.25

Weekday earnings           : $231.25
Weekend earnings           : $156.25

Total Salary               : $387.50

Make sure to download the data file which corresponds to
this pay stub and include in the project directory.
```

The data in the file is as follows:

```
Employee hourly wage rate
Hours worked on Monday
...
Hours worked on Sunday
```