Lab 2 – Assignment Statements

The heart of programming is based on the IPO model – that is, Input, Processing, Output. In this lab, we will be implementing this model by writing a syntactical Java program that will perform some processing.

For the first two parts of this exercise, we will work with a square. A square is the simplest quadrilateral, having all four sides equal and each internal angle is 90^o. For this exercise, the length of the side of the square will be entered via the terminal and we will calculate the perimeter and the area of a square based on this entered value. Once these are calculated, the perimeter and area will be printed as the output.

(Note: perimeter is 4 times the side and area is the square of the side)

To start, create a new BlueJ project and name it `Lab2_lastname_firstname`.

Terminal Input

The main way that the user interacts with the program is via terminal input. To allow the user to input via the terminal (or keyboard), Java uses what is known as the Scanner class. You will learn how to use the Scanner to input data into the program.

Before you can use the Scanner, you need to import it.

Part 1 – Using normal arithmetic operations

Create a new Driver class and name it "`Part1`". In this class, write a Java program that will do the following:

1. At the top of the program, before everything else, import the Scanner by entering the following:

```
import java.util.Scanner;
```

2. Inside the main method, declare the double variable `side` as well as the Scanner object to allow you to read the value of the side via the terminal. You also need to declare the variables `perimeter` and `area`. To do this, enter the following:

```
double side;
double perimeter;
double area;
Scanner in = new Scanner(System.in);
```

3. Ask the user to enter the length of the side of a square. You will need to prompt the user as to what you want the user to enter and use the Scanner to accept what the user will enter. The user prompt is important. Otherwise, the user will not know what to do. Enter the following:

```
System.out.println("Enter the length of the side");
side = in.nextDouble();
```

   The first line informs the user what is expected to be entered. The second line accepts what the user enters and assigns this to the variable `side`.

4. Calculate the perimeter and the area using the standard formulas for these. Assign the formula to respective variables. For this part, use (`side * side`) as the formula for the area of the square.

5. Print the results of the calculation. For this part, make sure that the printout is understandable and not just numbers. Hence, you might want to consider the following as an example of informative output:

   "`A square of side 2.0 has a perimeter of 8.0 and an area of 4.0.`"

6. Compile your program and fix whatever errors have been identified. When the program compiles cleanly, run the program as you have done in the introductory lab.

Part 2 – Using Java built in Math methods

The above program works and does what you expect it to do. However, we were fortunate that the formula for the area involves just squaring the side and we were able to use the simple assignment statement. However, when the power is higher than 2, using the normal arithmetic operations becomes quite cumbersome. Fortunately, Java comes with "built-in" Math Library that have methods which will allow us a much simpler way to deal with complex mathematical operations.

To gain experience with this, create a new Driver class and copy the main program from the previous part. Edit the program by replacing the formula for calculating the area with:

```
area = Math.pow(side,2)
```

`Math.pow` calculates the value of the first argument raised to the second argument. In this case, this calculates the square of the value of the variable `side`.

Once done, compile the program and run it using the data used in part 1.

Lab 2 – Assignment Statements

Additional Information

There are many other methods included in the Math Library of Java. For example,

```
result = Math.sqrt(value)
```

calculates the square root of `value` and assigns this to the variable `result`. You can check all the different methods contained in the Math Library by consulting the Java documentation (or Java API) by accessing the following website:

https://docs.oracle.com/javase/8/docs/api/index.html

Choose `Math` from the lower panel on the left side of the website. It is recommended that you familiarize yourself with this library by trying out some more familiar mathematical operations.

Part 3 – Using Integer Variables

In a recent tutorial, we developed an algorithm that uses integer variables. The algorithm determines the number of fruit per person as well the number returned given the total number of fruits available and the number of persons sharing the fruits.

Create a third Driver class and translate the algorithm developed in the tutorial into a Java program. Run the program with different inputs to see the results.

Submission

When you have completed this exercise, submit the entire folder into the submit drive by:

1. Compress (zip) the Java folder.
2. Submit the compressed folder to the submit drive under you section.