LAB 1

JANITH LAHIRU IM/2019/081

OBJECT OREANTED PROGRAMMING

INTE 12213– Object Oriented Programming

Lab Exercises 01

You are given the following tasks. Perform all of them and observe the compiler and run time system responses along with the changes in your working directory.

You have to use a text editor and the terminal in order to complete the below tasks and you are not permitted to use an IDE.

Prepare a report with the experiment, your observations on each task and conclusions. Submit the files by 11.59 pm on the lab session day to the CAL.

No marks will be given to any late submissions.

Lab01_Task01

Enter the following Java programming code using text editor and save it as "Task1.java".

```
class Task1{ }
```

To compile the source code: give the following command at the terminal

Your_Working_Directory$ javac name of the class with .java extension

```
/Documents/OOP2020_Practicals/Lab1$ javac Task1.java
```

I. What are the observations at the terminal?
   - **Nothing observed.**

II. What are the observations at your working directory?
   - **A byte code was created.**

To run the program: give the following command at the terminal.

Your_Working_Directory$ java name of the class

```
/Documents/OOP2020_Practicals/Lab1$ java Task1
```

III. What are the observations at the terminal?
   - **Give an error message ('Main method not found in class Task1)**

IV. What are the observations at your working directory?
   - **No changes were observed**

V. What are your conclusions on Lab01_Task 01?

- **javac – compiles the program creates the byte code**
- **java – opens and runs the byte code**
- **class need to contain a ‘Main’ method.**

Lab01_Task02

Enter the following Java programming code using text editor and save it as “Task2.java”.

```
class Task2{ }
```

To compile the source code: give the following command at the terminal.

```
Your_Working_Directory$ javac name of the class with .java extension
```

```
/Documents/OOP2020_Practicals/Lab1$ javac Task2.java
```

I. What are the observations at the terminal?
   - **No changes observed**

II. What are the observations at your working directory?
   - **Class file is created**

To run the program: give the following command at the terminal.

```
Your_Working_Directory$ java name of the class
```

```
/Documents/OOP2020_Practicals/Lab1$ java Task2
```

III. What are the observations at the terminal?

- **Give an error message (‘Main method not found in class Task1)**

IV. What are the observations at your working directory

- **No observations.**

V. What are your conclusions on Lab01_Task 02?
   - **javac – compiles the program creates the byte code**
   - **java – opens and runs the byte code**
   - **class need to contain a ‘Main’ method.**

Lab01_Task03

Change the Task1 class source code as below.

```
Class Task1{ }
```

To compile the source code: give the following command at the terminal.

Your_Working_Directory$ javac name of the class with .java extension

```
/Documents/OOP2020_Practicals/Lab1$ javac Task1.java
```

I. What are the observations at the terminal?

```
Task01.java:1: error: class, interface, enum, or record expected
Class Task1{ }
^
1 error
```

II. What are the observations at your working directory?

No changes observed

To run the program: give the following command at the terminal.

Your_Working_Directory$ java name of the class

```
/Documents/OOP2020_Practicals/Lab1$ java Task1
```

III. What are the observations at the terminal?

```
Error: Could not find or load main class Task03
Caused by: java.lang.ClassNotFoundException: Task03
```

IV. What are the observations at your working directory?

No changes

V. What are your conclusions on Lab01_Task 03?

You need to use ‘class’ keyword instead ‘Class’ to declare a variable.

Lab01_Task04

Enter the following Java programming code using text editor and save it as "Task4.java".

```
class Task4{

        public static void main (String[] args){

        }

 }
```

To compile the source code: give the following command at the terminal.

```
Your_Working_Directory$ javac name of the class with .java extension

/Documents/OOP2020_Practicals/Lab1$ javac Task4.java
```

I. What are the observations at the terminal?

   Complies without errors

II. What are the observations at your working directory?

   Class file was created

To run the program: give the following command at the terminal.

```
Your_Working_Directory$ java name of the class

/Documents/OOP2020_Practicals/Lab1$ java Task4
```

III. What are the observations at the terminal?

   Runs without errors

IV. What are the observations at your working directory?

   No changes

V. What are your conclusions on Lab01_Task 04?

   Run without any issue (because it contain main mehod)

Lab01_Task05

Change the source code of Task 04 as below.

```
class Task4{

        public static void main (String[] args){

                System.out.println(“Hello MIT”);

        }

 }
```

To compile the source code: give the following command at the terminal.

```
Your_Working_Directory$ javac name of the class with .java extension

/Documents/OOP2020_Practicals/Lab1$ javac Task4.java
```

I. What are the observations at the terminal?
No observations

II. What are the observations at your working directory?
Class file was created

To run the program: give the following command at the terminal.

```
Your_Working_Directory$ java name of the class

/Documents/OOP2020_Practicals/Lab1$ java Task4
```

III. What are the observations at the terminal?
Print → Hello MIT

IV. What are the observations at your working directory?
No changes

V. What are your conclusions on Lab01_Task 05?
‘System.out.println’ prints the words within the brackets and the double quotations marks

Lab01_Task06

Change the source code of Task 04 as below.

```
class Task4{

        public static void main (String[] args){

                System.out.print(“Hello MIT”);

                System.out.println(“Hello DIM”);

        }

 }
```

We faced issues relating to the the double quotes used inthis code . so we changed the double quotes and used valid programming double quotes.

To compile the source code: give the following command at the terminal.

```
Your_Working_Directory$ javac name of the class with .java extension

/Documents/OOP2020_Practicals/Lab1$ javac Task4.java
```

I. What are the observations at the terminal?

No changes were observed

II. What are the observations at your working directory?

Class file was created

To run the program: give the following command at the terminal.

```
Your_Working_Directory$ java name of the class

/Documents/OOP2020_Practicals/Lab1$ java Task4
```

III. What are the observations at the terminal?

Hello MIT

Hello DIM

IV. What are the observations at your working directory?

No changes were observed

V. What are your conclusions on Lab01_Task 06?

When the program was compiled and run; Hello MIT and Hello DIM were shown

Lab01_Task07

Enter the following Java programming code using text editor and save it as “Task7.java”.

```
class Task7{
        public static void main (String[] args){
                System.out.println(args.length);
                for (String s: args){
                        System.out.println(s);
                }
        }
 }
```

To compile the source code: give the following command at the terminal.

```
Your_Working_Directory$ javac name of the class with .java extension
```

```
/Documents/OOP2020_Practicals/Lab1$ javac Task7.java
```

I. What are the observations at the terminal?
No changes were observed

II. What are the observations at your working directory?
Class file was created (byte code)

To run the program: give the following command at the terminal.

```
Your_Working_Directory$ java name of the class
```

```
/Documents/OOP2020_Practicals/Lab1$ java Task7
```

III. What are the observations at the terminal?
0   (we can see zero)

IV. What are the observations at your working directory?
No changes

Re-run the program by giving the following command at the terminal.

```
Your_Working_Directory$ java name of the class with a set of strings
```

```
/Documents/OOP2020_Practicals/Lab1$ java Task7 Hello MIT
```

V. What are the observations at the terminal?

```
2
Hello
MIT
```

VI. What are the observations at your working directory?
No changes

Re-run the program by giving the following command at the terminal.

Your_Working_Directory$ java name of the class with a set of strings

/Documents/OOP2020_Practicals/Lab1$ java Task7 “Hello MIT”

VII. What are the observations at the terminal?

1
Hello MIT

VIII. What are the observations at your working directory?

No changes were observed

IX. What are your conclusions on Lab01_Task 07?

Output the number of string passing into the main method and print them out line by line using for loop.

Lab01_Task_08

Do some experiments and identify at least three things which can be changed in the given main method in Task8 class.

```
class Task8 {

        public static void main (String args[])   {

                System.out.println("Hello World");

        }

}
```

```
public class Task8 {

        public static void main (String [] args)   {

                System.out.print("Hello World");

        }

}
```

- **We can add public access modifier to our code .**
- **Can change parameters in main method.**
- **Can use print instead println.**

Lab01_Task_09

Use escape characters in your program and identify the purpose of each. Write down the purpose of each escape character based on your experience.

Enter the following Java programming code using text editor and save it as “Task9.java”.

```
class Task9 {

        public static void main(String args[])  {

                System.out.println(“UOK, \n BSc in MIT");

                System.out.println("Whatever you are, \n be a good one.");

        }

}
```

Continue the experiments replacing \n with \t, \b, \r, \f, \’, \”, \\

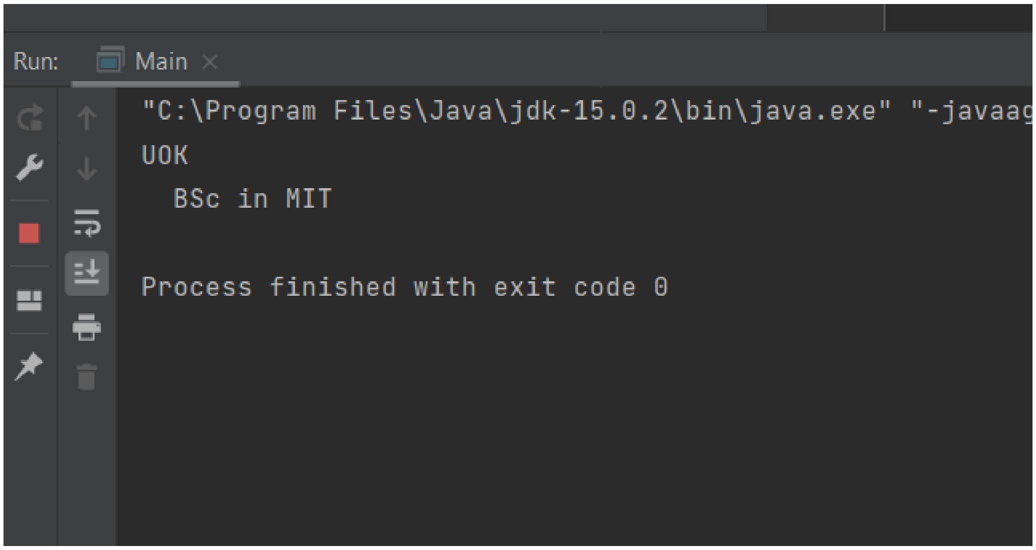
- **\n – New line**
- **\t – New tab**
- **\b – Backspace**
- **\r – The segment within the same System.out.println before the \r gets deleted**
- **\f – We can insert a quote at this point in the terminal**
- **\’ – Can print a single quote**
- **\” – Can print a double quote**
- **\\ - Can type an extra ‘\’ symbol in our code.**

Lab01_Task_10

Use printf in your program and identify the purpose of it. Write down the purpose printf and %s based on your experience.

Enter the following Java programming code using text editor and save it as “Task10.java”.

```
class Task10 {

        public static void main(String args[])  {

                System.out.printf(“%s\n %s\n “, “UOK”, “ BSc in MIT");

        }

}
```



```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-javaag
UOK
  BSc in MIT

Process finished with exit code 0
```

Printf – can use to print text.

String literels are parse into ‘%s’

Lab01_Task_11

Enter the following Java programming code using text editor and save it as “Task11.java”. Explain how the assignment operator works and purpose of any other symbols used in the program.

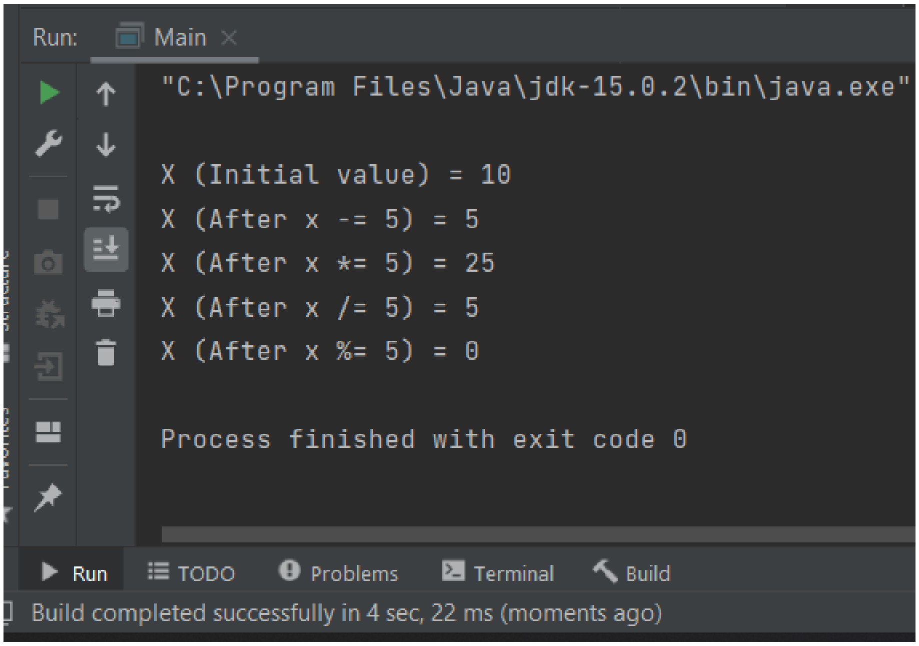
```
/**  This program shows how the assignment operator works. */

class Task11 {

            public static void main(String args[]) {

                        int    x = 10;

                        System.out.println("\n X (Initial value) = "+x);

                         x += 25;

                         System.out.println("X (After x += 25) = "+ x);

            }

}
```

- **X = X + 25 can be use as x += 25**
- **combines the two segments (concatenates)**
- **/**   */ - To type comments**

Lab01_Task_12

Enter the following Java programming code using text editor and save it as “Task12.java”. Explain what you learn when developing the task.

```
 //This program shows how the assignment operators work.

class Task12 {

        public static void main(String args[])  {

                int    x = 10;

                System.out.println("\nX (Initial value) = "+x);

                x -= 5;

                // x = x - 5;

                System.out.println("X (After x -= 5) = "+ x);

                x *= 5; // x = x * 5;

                System.out.println("X (After x *= 5) = "+ x);

                x /= 5; // x = x / 5;

                System.out.println("X (After x /= 5) = "+ x);

                x %= 5; // x = x % 5;

                System.out.println("X (After x %= 5) = "+ x);

        }

}
```



```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe"

X (Initial value) = 10
X (After x -= 5) = 5
X (After x *= 5) = 25
X (After x /= 5) = 5
X (After x %= 5) = 0

Process finished with exit code 0
```

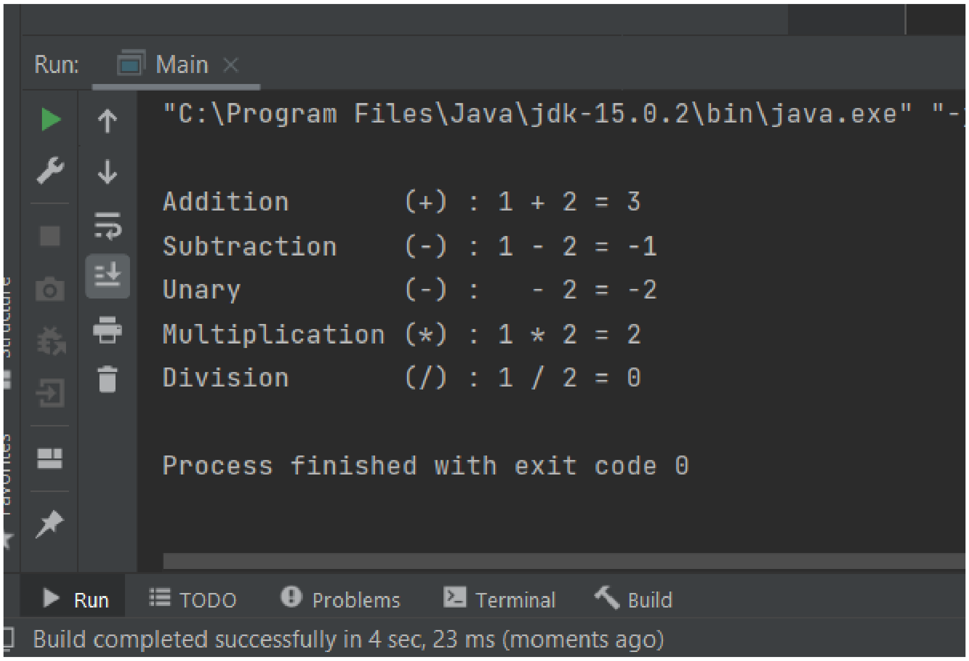
Build completed successfully in 4 sec, 22 ms (moments ago)

- **We can use '//' to commented out the informations that we can use later to read the code for any purpose.**

Lab01_Task_13

Enter the following Java programming code using text editor and save it as “Task13.java”. Explain what you learn when developing the task.

```
 //This program shows how the basic arithmetic operators work for integer.

class Task13 {

          public static void main(String args[])  {

                    int x, y;

                    x= 1;  y =2;

                    System.out.println("\nAddition       (+) : "+ x + " + "+y + " = "+ (x + y));

                    System.out.println("Subtraction    (-) : "+ x + " - "+y + " = "+ (x - y));

                    System.out.println("Unary          (-) :   - "+y + " = "+ -y);

                    System.out.println("Multiplication (*) : "+ x + " * "+y + " = "+ (x * y));

                    System.out.println("Division       (/) : "+ x + " / "+y + " = "+ (x / y));

          }

}
```



```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-

Addition       (+) : 1 + 2 = 3
Subtraction    (-) : 1 - 2 = -1
Unary          (-) :   - 2 = -2
Multiplication (*) : 1 * 2 = 2
Division       (/) : 1 / 2 = 0

Process finished with exit code 0
```

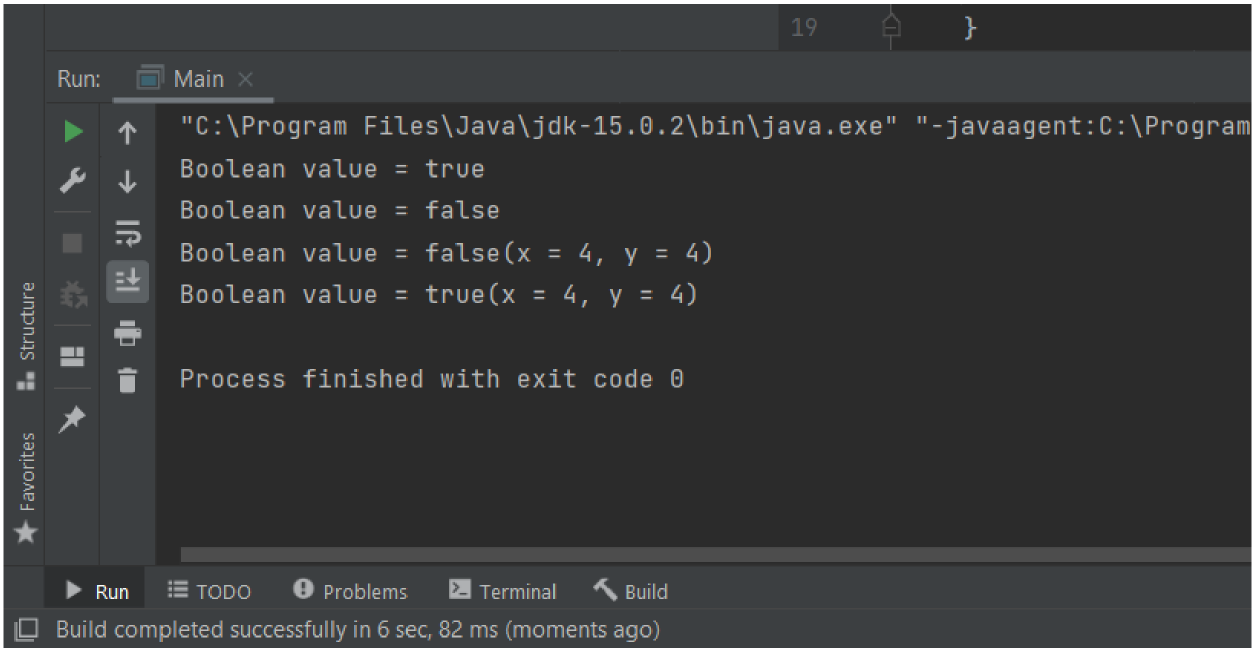
Build completed successfully in 4 sec, 23 ms (moments ago)

- **We learned how to perform mathamatical operations in java. (addition,substraction,multiplication,division)**

Lab01_Task_14

Enter the following Java programming code using text editor and save it as “Task14.java”. Explain what you learn when developing the task.

```
 //This program shows how to use boolean type variables.

class Task14 {

        public static void main (String argv[]) {

                int x, y;

                boolean Boolean_variable;

                Boolean_variable = true;

                System.out.println("Boolean value = "+ Boolean_variable);

                Boolean_variable = false;

                System.out.println("Boolean value = "+ Boolean_variable);

                x = y = 4;

                Boolean_variable = (x != y);

                System.out.println("Boolean value = "+ Boolean_variable+"(x = "+x+", y = "+y+")");

                Boolean_variable = (x == y);

                System.out.println("Boolean value = "+ Boolean_variable+"(x = "+x+", y = "+y+")");

        }

}
```



```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-javaagent:C:\Program
Boolean value = true
Boolean value = false
Boolean value = false(x = 4, y = 4)
Boolean value = true(x = 4, y = 4)

Process finished with exit code 0
```

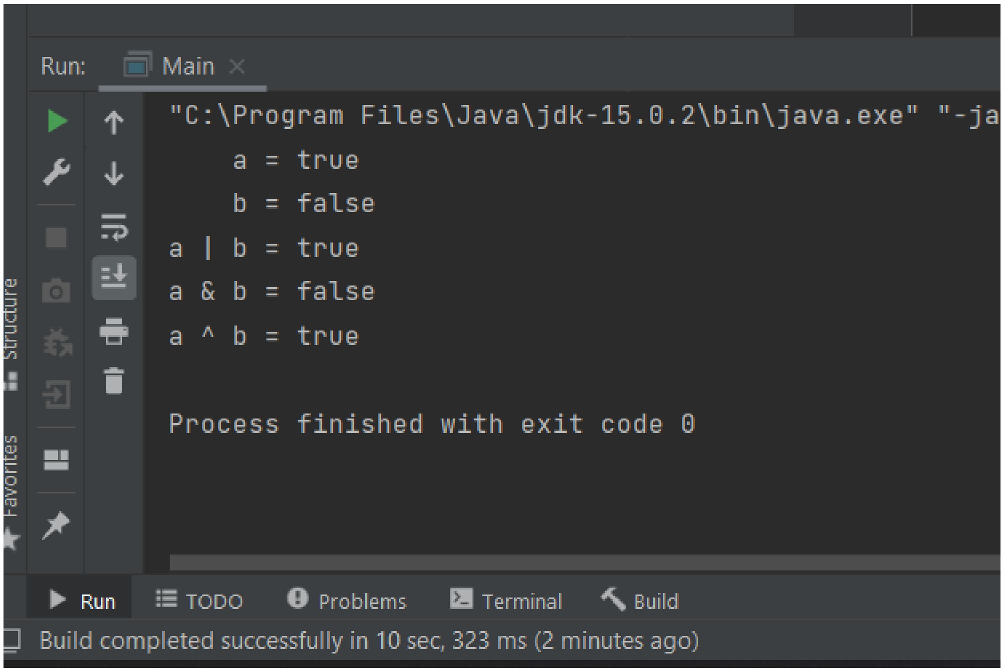
Build completed successfully in 6 sec, 82 ms (moments ago)

- We learn how to declare a boolean variable and how to properly use it.

Lab01_Task_15

Enter the following Java programming code using text editor and save it as “Task15.java”. Explain what you learn when developing the tasks.

```
 //This program shows how the Boolean logical operators work.

class Task15{

        public static void main(String argv[])  {

                boolean a = true;

                boolean b = false;

                boolean c = a | b;

                boolean d = a & b;

                boolean e = a ^ b;

                System.out.println("    a = "+a);

                System.out.println("    b = "+b);

                System.out.println("a | b = "+c);

                System.out.println("a & b = "+d);

                System.out.println("a ^ b = "+e);

        }

}
```



```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-ja
    a = true
    b = false
a | b = true
a & b = false
a ^ b = true

Process finished with exit code 0
```

Build completed successfully in 10 sec, 323 ms (2 minutes ago)

Observations

- **Learned things like bitwise or , and ,xor in java programming.**