Langara.
THE COLLEGE OF HIGHER LEARNING.

OBJECT-ORIENTED PROGRAMMING(CPSC 1811)

Assignment #1

Due Date: September 25, 2022

Mourad Bouguerra
mbouguerra@langara.ca

I Using Bitwise Operators

The most basic unit of computer storage is the **bit**. **Java** provides six (6) **bitwise** operators that allow manipulating the individual bits of integer type: `byte`, `char`,`short`,`int long`. More efficiency can be achieved by using bitwise operations instead of arithmetic operations. We can distinguish two types of bitwise operators:

① Bitwise Shift Operators

② Bitwise Logical Operators

I.1 Bitwise Shift Operators

Java provides two (2) binary bitwise shift operators, which shift the bits in an integer type variable a specified number of positions:

❒ Right Shift, denoted by `x>>n`

❒ Zero Fill Right Shift, denoted by `x>>>n`

❒ Left Shift, denoted by `x<<n`

Java Bitwise Shift Operators			
Operator	Symbol	Example	Description
Right Shift	`>>`	`x>>n`	shift right the bits of x by n positions shifted positions filled by sign bit
Zero Fill Right Shift	`>>>`	`x>>>n`	shift right the bits of x by n positions shifted positions filled by zero
Left Shift	`<<`	`x<<n`	shift left the bits of x by n positions

I.2 Bitwise Logical Operators

Java provides four (4) bitwise logical operators. Unlike, logical operators which operate on Boolean expressions, bitwise logical operators operate on the individual bits of the binary representation of the integer data types.

Java Bitwise Logical Operators			
Operator	Symbol	Example	Description

AND	`&`	`x&y`	compare the bits of x and y
			`0&0=0, 1&0=0`
			`1&0=0, 1&1=1`

OR	`|`	`x|y`	compare the bits of x and y
			`0|0=0, 1|0=1`
			`1|0=1, 1|1=1`

XOR	`^`	`x^y`	compare the bits of x and y
			`0^0=0, 1^0=1`
			`1^0=1, 1^1=0`

NOT (Complement)	`~`	`~x`	reverse the bits of x

II Requirements

In this assignment you are required to use bitwise operators in order to compute the minimum, maximum of the following integer data types: `byte`, `short`, `int`, `long`.

Part-I
Creating The Program Menu

(12 marks)

1. Open a text editor, and create `Assignment1.java`

```java
import java.util.Scanner;
public class Assignment1
{
  final static int QUIT=9;
    public static void displayMenu()
  {
        //Enter your code here
  }
  public static int getMenuChoice()
  {
    //Enter your code here
  }
  public static byte getMinimumByte()
  {
    //Enter your code here
  }
  public static byte getMaximumByte()
  {
    //Enter your code here
  }
  public static short getMinimumShort()
  {
    //Enter your code here
  }
  public static short getMaximumShort()
  {
    //Enter your code here
  }
  public static int getMinimumInt()
```

```
  {
    //Enter your code here
  }
  public static int getMaximumInt()
  {
    //Enter your code here
  }
  public static long getMinimumLong()
  {
    //Enter your code here
  }
  public static long getMaximumLong()
  {
    //Enter your code here
  }

  public static void main(String [] args)
  {
    //Enter your code
  }
}
```

2. Write down the code of the method `displayMenu()` that produces the following menu (see figure 1 on page 5): (2 marks)

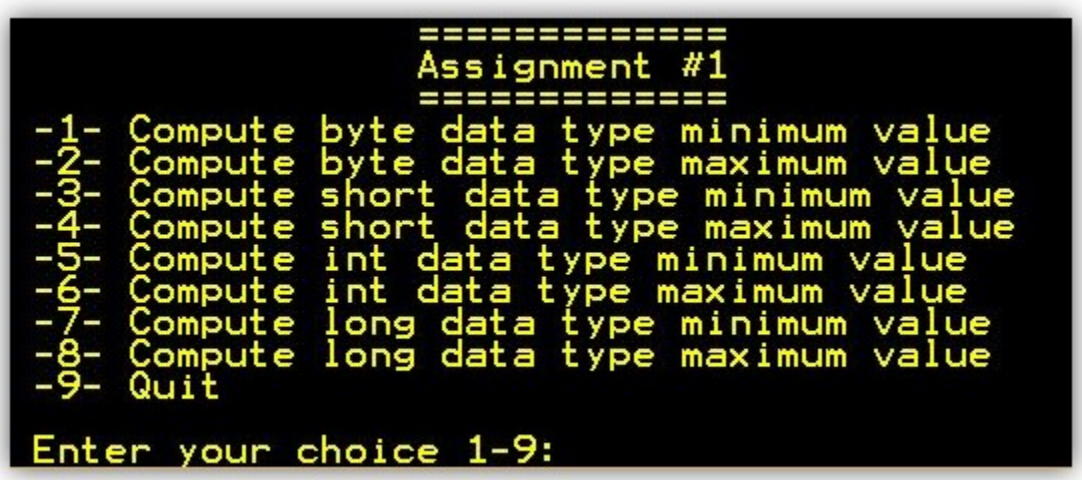

Figure 1: Program Menu

3. Write down the code of the method `getMenuChoice()` that returns the user choice a number between 1 and 9. The method should validate user input, and

keeps displaying the menu until the user enter a valid choice. Use appropriate control structures (see figure 2 on page 6). (5 marks)

```
                ==============
                Assignment #1
                ==============
-1- Compute byte data type minimum value
-2- Compute byte data type maximum value
-3- Compute short data type minimum value
-4- Compute short data type maximum value
-5- Compute int data type minimum value
-6- Compute int data type maximum value
-7- Compute long data type minimum value
-8- Compute long data type maximum value
-9- Quit

Enter your choice 1-9: 16667
                ==============
                Assignment #1
                ==============
-1- Compute byte data type minimum value
-2- Compute byte data type maximum value
-3- Compute short data type minimum value
-4- Compute short data type maximum value
-5- Compute int data type minimum value
-6- Compute int data type maximum value
-7- Compute long data type minimum value
-8- Compute long data type maximum value
-9- Quit

Enter your choice 1-9:
```

Figure 2: User Choice Validation

4. Write down the code of the method `main()` that keeps running the program until the user chooses 9 to quit the application . (5 marks)

Part-II
Computing Minumum & Maximum Using Bitwise Operators

(25 marks)

✎ Write down the code of the following 6 methods (see table 1 on page 8 to test your code)

① `getMinimumByte()` that returns the minimum value of the `byte` data type. (2 marks)

② `getMaximumByte()` that returns the maximum value of the `byte` data type. (3 marks)

③ `getMinimumShort()` that returns the minimum value of the `short` data type. (2 marks)

④ `getMaximumShort()` that returns the maximum value of the `short` data type. (3 marks)

⑤ `getMinimumInt()` that returns the minimum value of the `int` data type. (2 marks)

⑥ `getMaximumInt()` that returns the maximum value of the `int` data type. (3 marks)

⑦ `getMinimumLong()` that returns the minimum value of the `long` data type. (5 marks)

⑧ `getMaximumLong()` that returns the maximum value of the `long` data type. (5 marks)

Java Primitive Integer Data Types			
Data Type	Java	Size (Bytes)	Value Range
Byte	byte	1	$-2^7 = -128$ To $2^7 - 1 = 127$
Short Integer	short	2	$-2^{15} = -32,768$ To $2^{15} - 1 = 32767$
Integer	int	4	$-2^{31} = -2,147,483,648$ To $2^{31} - 1 = 2,147,483,647$
Long integer	long	8	$-2^{63} = -9,223,372,036,854,775,808$ To $2^{63} - 1 = 9,223,372,036,854,775,807$

Table 1: Range of Integer Types

III Marking Scheme

Task	Marks
Coding	37
Coding Style	5
Using javadoc to generate documentation	4
Using jar to archive all assignment files	4
Task	**50**

IV submission

Submission

✎ Upload your **Your-Name-ID.jar** archive file of your **Java** folder project to the submission Page on Brightspace.