Hands-on Experiment # 4 : Worksheet

Section_____2_____ Date_____08/02/2018_____

No more than 3 students per one submission of this worksheet.

Student ID _____6031851521_____ Name___Sarun Nuntaviriyakul_____

Student ID _____6031848721_____ Name___Watcharin Kriengwatana_____

Student ID _____6031847021_____ Name___Wasuthon Klyhirun_____

Part A: Using Objects and Classes

1. Use your program to fill up the table below. Prefix your answers with 0 so that all representations are shown using 32 bits.

Value	32-bit Binary Representation
0	0000 0000 0000 0000 0000 0000 0000 0000
1	0000 0000 0000 0000 0000 0000 0000 0001
-1	1111 1111 1111 1111 1111 1111 1111 1111
2	0000 0000 0000 0000 0000 0000 0000 0010
-2	1111 1111 1111 1111 1111 1111 1111 1110
3	0000 0000 0000 0000 0000 0000 0000 0011
-3	1111 1111 1111 1111 1111 1111 1111 1101
4	0000 0000 0000 0000 0000 0000 0000 0100
-4	1111 1111 1111 1111 1111 1111 1111 1100
5	0000 0000 0000 0000 0000 0000 0000 0101
-5	1111 1111 1111 1111 1111 1111 1111 1011
255	0000 0000 0000 0000 0000 1000 1010 0101
-255	1111 1111 1111 1111 1111 0111 0101 1011

2. Observe the results of the positive and negative integers with the same magnitude. Try to make an assumption on how Java represents a negative integer.
   Hints:
   - Observe the left-most bit and see whether it needs to be a specific value for all positive integers. How about all negative integers.
   - What will happen if you try to add the binary numbers corresponding to x and –x?
   - If you minus 1 from the binary representation of a negative integer, what can say about each bit of the result compared to its corresponding position in the positive integer with similar magnitude?

State your assumption below.

Switch all the 1's and 0's of the positive version of that number and that increase it by one.

For example:

5 in binary representation is 0000 0000 0000 0000 0000 0000 0000 0101

To make it -5, first you swap 1 with 0 and 0 with 1. So, it becomes 1111 1111 1111 1111 1111 1111 1111 1010

Then you plus one to that number -> 1111 1111 1111 1111 1111 1111 1111 1011

3. Write another program to show the binary representation of
   a. The maximum value that an `int` value can hold (the most positive)
   b. The minimum value that an `int` value can hold (the most negative)

   Hints: The some static data in the *Integer* class will be useful.

Value	32-bit Binary Representation
MAX_VALUE	0111 1111 1111 1111 1111 1111 1111 1111
MIN_VALUE	1000 0000 0000 0000 0000 0000 0000 0000

List the source code of you program below.

```
import java.lang.Integer;

public class test{

	public static void main(String[] args) {

		System.out.println(Integer.toBinaryString(Integer.MAX_VALUE));;

		System.out.println(Integer.toBinaryString(Integer.MIN_VALUE));;

	}

}
```

Submit this worksheet as a group via http://www.myCourseVille.com (Assignments > Hands-on Experiment # 4) before noon of the day after your lecture.