

Assignment-3

G.Sathish

1.Bitwise Complement(~):

This unary operator returns the one's complement representation of the input value or operand, i.e, with all bits inverted, which means it makes every 0 to 1, and every 1 to 0. (or)

It performs a bitwise negation on an integer value by flipping all the bits. It operates at the binary level, flipping each bit from 0 to 1 and vice versa. The result is the one's complement of the given value.

Bitwise Complement

Ex:-

$a = 5$

Binary no of 5 is 0101 *

1's complement 1010 *

$\sim \text{Number} = -(\text{Number} + 1)$

$\sim(5) = -(5+1)$

$= -6$

Binary no of (-6) can be stated as \rightarrow 0110

1's comp 1001

+ 1 *

2's complement 1010

Hence 2's complement of (-6) is equivalent to 1's complement of 5

Hence we can say that complement of 5 is $-(5+1) = -6$.

Ans:

0101

8 4 2 1

0 1 0 1

1010

5 1

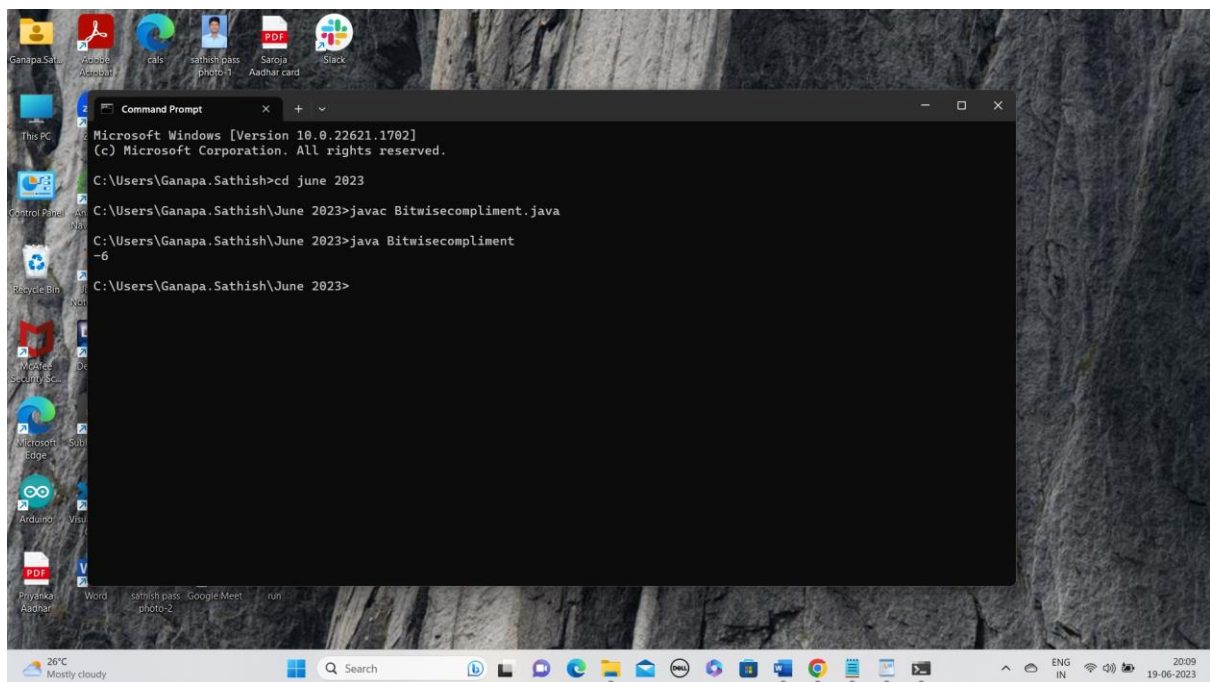
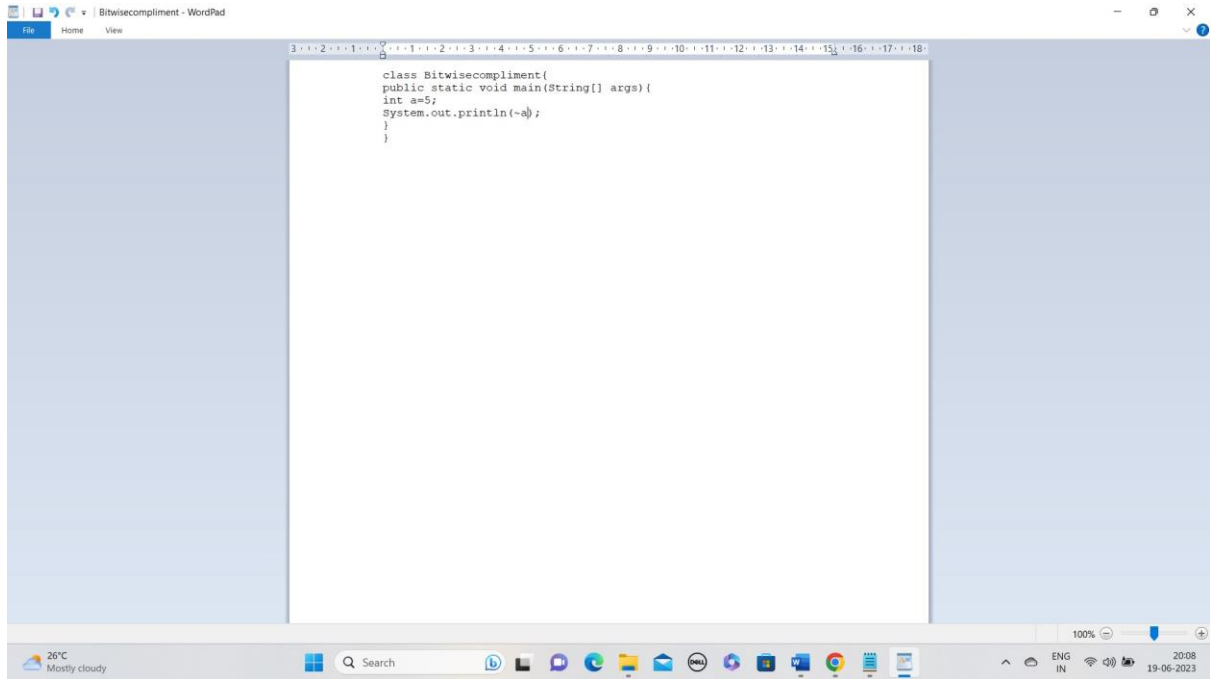
1011

0101

0110

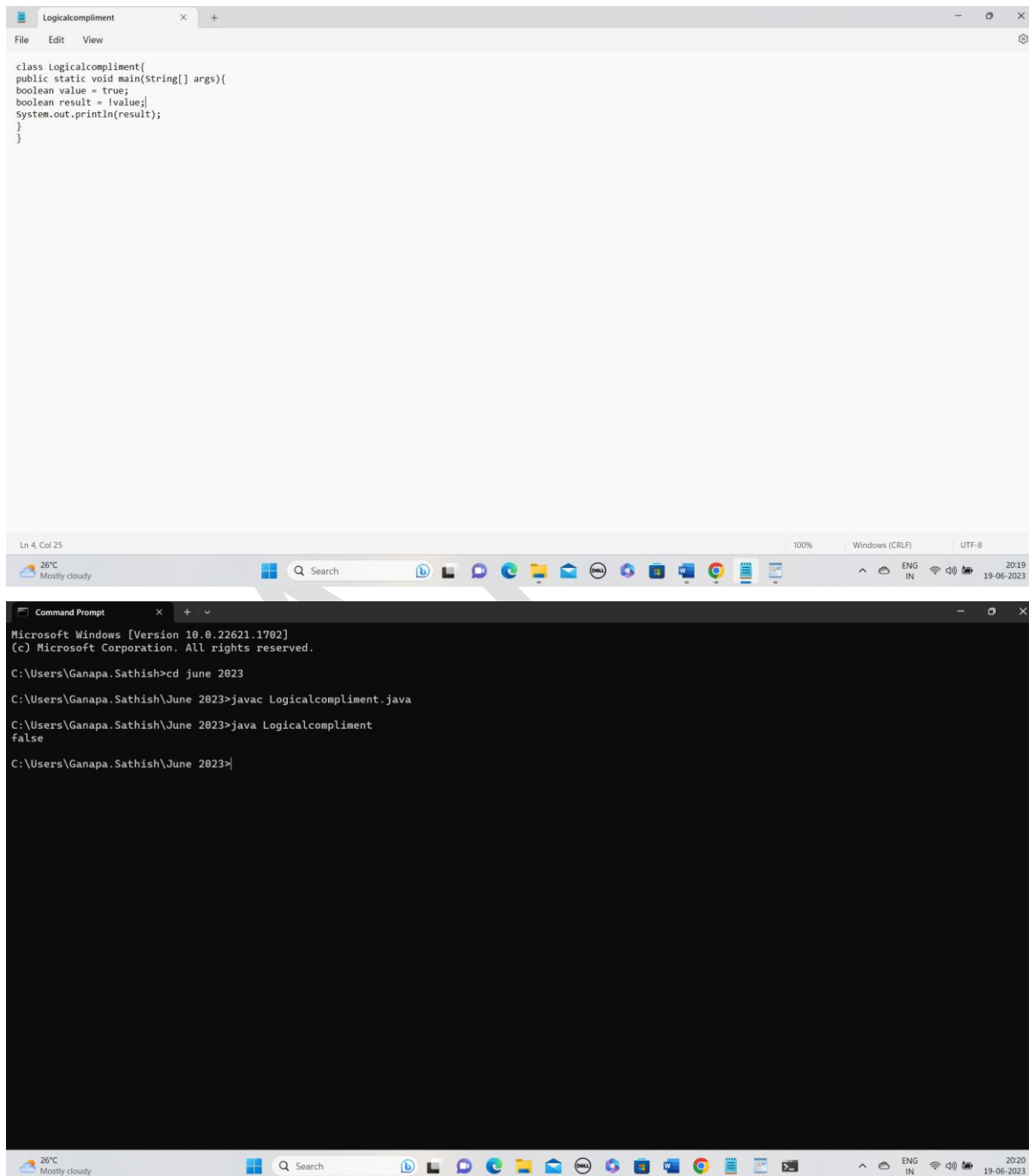
1

0111



2.Logical Complement (!):

The logical complement operator (!) is used to negate a boolean value. It reverses the logical state of the operand. If the operand is true, it will return false, and if the operand is false, it will return true.



The image shows a Windows desktop environment. At the top, a code editor window titled 'Logicalcompliment' contains the following Java code:

```
class Logicalcompliment{
public static void main(String[] args){
boolean value = true;
boolean result = !value;
System.out.println(result);
}
}
```

Below the code editor, a Windows taskbar is visible with various application icons. At the bottom, a Command Prompt window is open, displaying the following commands and output:

```
Microsoft Windows [Version 10.0.22621.1782]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ganapa.Sathish>cd june 2023
C:\Users\Ganapa.Sathish\June 2023>javac Logicalcompliment.java
C:\Users\Ganapa.Sathish\June 2023>java Logicalcompliment
false
C:\Users\Ganapa.Sathish\June 2023>
```

The Command Prompt output shows that the program compiled successfully and executed, printing 'false' as the result of the logical complement operation on the initial 'true' value.

In the above example, the initial value of the `value` variable is `true`. The logical complement operator negates this value, resulting in `false`. The `result` variable is then assigned the value `false`,

2)

****After long datatype the value can be stored in BigInteger

BigInteger:

If you want to store a value like 9223372036854775808, which exceeds the range of the long data type in Java, you can use the BigInteger class to handle arbitrary-precision integers. The BigInteger class can store and perform operations on integers of any size.

3)

Range for float and double:

Floating-Point Data Type	Values	Size (bits)*	Storage Requirement (bytes)	Default Value	Precision	Decimal Digits	Range
float	IEEE 754 Floating-Point	32	4	0.0f	Single	6 decimal digits	3.4e-038 to 3.4e+038
double	IEEE 754 Floating-Point	64	8	0.0d	Double	15 decimal digits	1.7e-308 to 1.7e+308