FULL STACK DEVELOPER

RITU BALA RAI

DATE-06/01/2020

**Q1.** Why there is extra blank line after the end of output in java?

**A1:-** There is blank line at the end of output in java because we are using **System.out.println** instead of using this we should write **System.out.print** for not creating extra blank line in the end of java output.

**Q2.** Refer file Minmax.java

**Q3.** What is class Resolution in java?

**A3.** It is basically getting references to the symbolic representation of different symbols present in the class.

**Q4.** Difference between signed and unsigned int and how to represent in java?

**A4:** Signed integer:-

A signed integer is one with either a plus or minus sign in front.

Thus a 16 bit signed integer only has 15 bits for data.

Unsigned integer:-

Java does not supports unsigned integer.

A 16 bit unsigned integer has all 16 bits available.

This means unsigned integers can have a value twice as high as signed integers (but only positive values).

**Q5.** What will be the output of: a&b, a|b, a^b , ~a.

**A5:-** class Greatest {

public static void main(String args[])

{

int a= 12,b=13,max,ch,re;

max=a|b;

ch=a&b;

re=a^b;

System.out.println(max);

System.out.println(ch);

System.out.println(re);

}

}

**Output:-** 13 // a|b output

12 // a&b

1 // a^b

**Logic:-**

a=12, binary value of a= 1100

b=13, binary value of b=1101

**OR-**  a|b:- 1100

1101

OR operation result =1101 i.e. 13.

**AND-** a&b:- 1100

1101

AND operation result =1100 i.e. 12.

**^** will perform **XOR** and result will be 0001 i.e. 1 is result of a^b.

**~a** – will give **negated** value of a.

**Q6:-** What is shift operator? When it can be used?.

**A6:-** Shift operators are logical operators. Like we have signed shift operator (>>) and unsigned shift operator (>>>). We can use shift operators for speeding up the multiplication / division operation than traditional multiplication / division.

Each **operator shifts** the bits of the first operand over by the number of positions indicated by the second operand.

Example- op1<<op2 (bitwise left shift operator)

Here op1 is bit value of binary objects and op2 is number of bits needs to be shifted.

op1>>op2 (bitwise right shift operator).

Q7. Difference b/w switch case and else-if ladder?

Ans. i) Control in else-if ladder will go to each and every else-if statement till it finds a true statement to execute .

On the other hand, switch case will jump directly to the case specified in case statement.

ii) switch case allows comparison of single expression only against collection of several discrete values.

iii) Else if ladder statement works on the basis of true false (zero/non-zero) basis and switch case works on the basis of equality operator.

iv) In switch case use of break keyword is mandatory but not in if else if ladder.