1. **Program to find the largest number using ternary operator ?**

class Ternary{

public static void main(String[] args){

int a=9, b=54, c=1, result, temp;

temp = a>b ? a:b;

result = c>temp ? c:temp;

System.out.println("Largest Number is:"+result);

}

}

Result =>

Largest Number is:54

## Check whether an alphabet is vowel or not?

## class Vowel {

## public static void main(String[] args) {

## char ch = 'A';

## switch (ch) {

## case 'a':

## case 'e':

## case 'i':

## case 'o':

## case 'u':

## case 'A':

## case 'E':

## case 'I':

## case 'O':

## case 'U':

## System.out.println(ch + " is vowel");

## break;

## default:

## System.out.println(ch + " is consonant");

## }

## }

## }

## Result =>

## A is vowel

## What is unsigned and signed integer ?

Unsigned: It consists of only non-negative values i.e

0 to 255.

Signed: It consist of both negative and positive values but in different formats like

* 1. 0 to +127
  2. -1 to -128

And this is about the 8-bit number system.

## Shift operator in java ?

## **<< (Left shift)** :-

## The operator that shifts the bits of number towards left by n number of bit positions is called left shift operator in Java. This operator is represented by a symbol <<, read as double less than. if we write x << n, it means that the bits of x will be shifted towards left by n positions.

## Let us take an example to understand the concept of the left shift operator. If int x = 20. Calculate x value if x << 3. The value of x is 20 = 0 0 0 1  0 1 0 0 (binary format). Now x << 3 will shift the bits of x towards left by 3 positions. Due to which leftmost 3 bits will be lost. Hence, after shifting, bits of x is 1 0 1 0  0 0 0 0 that is 160 in decimal form.

## 2. Right Shift Operator in Java

## If we write x >> n, it means that the bits of x will be shifted towards right by n positions. There are two types of right shift operators in java: a. Signed right shift operator (>>) b. Unsigned right shift operator (>>>).

## Signed Right Shift Operator in Java

The signed right shift operator >> shifts bits of the number towards the right and also reserve sign bit, which is leftmost bit. A sign bit represents the sign of a number. If the sign bit is 0 then it represents a positive number. If sign bit is 1, it represents a negative number.

## If the number is positive, the leftmost position is filled with 0. If the number is negative, the leftmost position is filled with 1. The signed shift operator uses the same sign as used in the number before shifting of bits.

## Unsigned Right Shift Operator in Java

The unsigned right shift operator in java performs nearly the same operation as the signed right shift operator in java. The unsigned right shift operator is represented by a symbol >>>, read as triple greater than.

The unsigned right shift operator always fills the leftmost position with 0s because the value is not signed. Since it always stores 0 in the sign bit, it is also called zero fill right shift operator in java.

DATE : 6-01-2020

**1. Class resolution.Whether it is done in compile time or runtime?**

**Compile-Time Resolution of Strings:**

When Strings are created with the help of ‘+’ operator, they get concatenated at compile time. This is referred to as Compile-Time Resolution of Strings. Compiler eliminates the concatenation operator and optimizes the string.

String str = "Geeks "

+ "for"

+ "Geeks";

the above code optimizes in the compile time:

String str = "GeeksforGeeks";

**RunTime Resolution of Strings**:

When Strings are created with the help of String literals along with variables and ‘+’ operator, they get concatenated at runtime only, as the value of the variables cannot be predicted beforehand. This is referred to as the RunTime Resolution of Strings.

String str = "Geeks " + var + "Geeks";

**2.Whether switch statement supports boolean or float or double?**