

Working with Logical and Boolean Operator :

|| : Logical AND (All the conditions must be true)
| : Logical OR (At least one condition must be true)
! : Inverter, Making true as a false and false as a true

Boolean Opeartor :

& : Boolean AND
| : Boolean OR
* Boolean operator will evaluate all right side expression regardless of first condition is true/false.
public class Test
{
 public static void main(String[] args)
 {
 int z = 5;
 if(++z > 5 || ++z > 6) //Logical OR
 {
 z++;
 }
 System.out.println(z); //7

 System.out.println(".....");

 z = 5;
 if(++z > 5 | ++z > 6) //Boolean OR
 {
 z++;
 }
 System.out.println(z); //8
 }
}

public class Test
{
 public static void main(String[] args)
 {
 int z = 5;

 if(++z > 6 & ++z > 6)
 {
 System.out.println("Inside If");
 z++;
 }
 System.out.println(z); //7
 }
}

Working with Bitwise AND(&), Bitwise OR(|) and Bitwise X-OR (^) :

Truth table of AND gate (A.B) :

A B Output
0 0 0
0 1 0
1 0 0
1 1 1

Truth table of OR gate (A + B) :

A B Output
0 0 0
0 1 1
1 0 1
1 1 1

Truth table of X-OR Gate $A \oplus B$

A B Output
0 0 0
0 1 1
1 0 1
1 1 0

System.out.println(5 & 6);
System.out.println(5 | 6);
System.out.println(5 ^ 6);

4 2 1
Binary of 5 1 0 1
Binary of 6 1 1 0

How to perform calculation :

1	0	1
1	1	0
1	0	0
1	1	1
0	1	1

5 & 6 The result will come from truth table
5 | 6
5 ^ 6

//Program :

public class Test
{
 public static void main(String[] args)
 {
 System.out.println(5 & 6); //4
 System.out.println(5 | 6); //7
 System.out.println(5 ^ 6); //3
 }
}

public class Test
{
 public static void main(String[] args)
 {
 System.out.println(false ^ true); //true [Same input output false]
 }
}

Bitwise Complement Operator (~)

* It will not work with boolean literal, It will only work with number.
public class Test
{
 public static void main(String[] args)
 {
 System.out.println(~true); //error
 }
}

public class Test
{
 public static void main(String[] args)
 {
 System.out.println(~ 5); //-6
 System.out.println(~ -4); //3
 }
}

Member access operator :

* It is also known as Dot operator.
* It is mainly used to access the member of the class.
class Welcome
{
 public static final int MAX_VALUE = 999;

 public static void batchInfo()
 {
 System.out.println("Batch 46");
 }
}

public class Test
{
 public static void main(String[] args)
 {
 Welcome.batchInfo();
 System.out.println(Welcome.MAX_VALUE);
 }
}

new Operator :

* It is also kyword in java.
* By using new keyword we can create an object with the help of class name and constructor.
* new keyword is responsible to provide **default initial value** for non static variable with the help of java compiler.
class Welcome
{
 int val = 999; //non static variable

 public void batchInfo() //non static method
 {
 System.out.println("Batch 46");
 }
}

public class Test
{
 public static void main(String[] args)
 {
 Welcome w = new Welcome();
 w.batchInfo();
 System.out.println(w.val);
 }
}

Limitation of if else :

The major drawback with if condition is, it checks the condition again and again so It increases the burdon over CPU so we introduced switch-case statement to reduce the overhead of the CPU.

Switch case statement in java :

It is a selective statement so, we can select one statement among the available statements.
break is optional but if we use break then the control will move from out of the switch body.
We can write default so if any statement is not matching then default will be executed.

Allowed type in switch expression : byte, short, char, int, Byte, Short, Character,Integer, enum(1.5), String(java 7v)
Not Allowed in switch expression : long, float, double, boolean
Note : long is allowed from JDK 14V with new style of switch case.

* In the case label which is a constant we can take variable but the valriable must be declared as final
* While working with byte and short the range of the case must be within the range of byte and short.

import java.util.*;
public class SwitchDemo
{
 public static void main(String[] args)
 {
 Scanner sc = new Scanner(System.in);
 System.out.print("Please choose any color name :");
 char colour = sc.next().toLowerCase().charAt(0);

 switch(colour)
 {
 case 'r' : System.out.println("Red") ; break;
 case 'g' : System.out.println("Green");break;
 case 'b' : System.out.println("Blue"); break;
 case 'w' : System.out.println("White"); break;
 default : System.out.println("No colour");
 }
 System.out.println("Completed") ;
 }
}

import java.util.Scanner;
public class SwitchDemo1
{
 public static void main(String args[])
 {
 System.out.println("\t\t**Main Menu**\n");
 System.out.println("\t\t**100 Police**\n");
 System.out.println("\t\t**101 Fire**\n");
 System.out.println("\t\t**102 Ambulance**\n");
 System.out.println("\t\t**139 Railway**\n");
 System.out.println("\t\t**181 Women's Helpline**\n");

 System.out.print("Enter your choice :");
 Scanner sc = new Scanner(System.in);
 int choice = sc.nextInt();

 switch(choice)
 {
 case 100:
 System.out.println("Police Services");
 break;
 case 101:
 System.out.println("Fire Services");
 break;
 case 102:
 System.out.println("Ambulance Services");
 break;
 case 139:
 System.out.println("Railway Enquiry");
 break;
 case 181:
 System.out.println("Women's Helpline");
 break;
 default:
 System.out.println("Your choice is wrong");
 }
 }
}

import java.util.*;
public class SwitchDemo2
{
 public static void main(String[] args)
 {
 Scanner sc = new Scanner(System.in);
 System.out.print("Enter the name of the season :");
 String season = sc.next().toUpperCase();

 switch(season) //String allowed from 1.7 [It is constant]
 {
 case "SUMMER" :
 System.out.println("It is summer Season!!");
 break;

 case "RAINY" :
 System.out.println("It is Rainy Season!!");
 break;
 }
 }
}

public class Test2
{
 public static void main(String[] args)
 {
 long val = 1;
 switch(val) //Error, can't pass long, float and double
 {
 case 1:
 System.out.println("Hello");
 break;
 }
 }
}