**OPERATORS**

* An operator is a special symbol or keyword that is used to designate mathematical or some other type of operations.
* These operations can be performed on one or more than one values called as operands.

**ARITHEMATIC OPERATIONS**

+ \*

Multiplication

Additive

* /

%

int i;

char a;

res='a'+i;

System.out.println("Char and int Addition= "+res);

The resultant will always be of greater datatype i,e of int in above case.(int-4bytes,char 1 byte)

**COMPOUND ASSIGNMENT OPERATORS:**

*+= ,-=, \*=, /= ,%=*

i=10;

j=20;

i+=j; //i=10+20

**INCREMENT AND DECREMENT OPERATOR**

* Increases and decreases variable value by one unit
* Can be used after or before the variable.
* When used after variable – post increment and post decrement

int i=10;

*1. Substitute*

*2. Utilize*

*3.Incement*

int j= i++;

System.out.println(j); //10

* When used before variable – pre increment and post increment

j=++i;

*1.Incement*

*2.Substitute*

*3.Utilize*

System.out.println(j); //12

* They are also called as unary operators

**RELATIONAL OPERATOR**

* *== --- returns true when both sides of equation are equal else returns false*
* *!= --- returns true if both sides of equation is true*
* *< --- returns true if left side of the equation is greater than right else returns false*
* *> --- returns true if right side of the equation greater than left else returns false*
* *>= , <=*
* Return value of relational operator is Boolean(true or false).

**LOGICAL OPERATOR**

* *NOT --- can be used in only one operator. Unique operator which returns false if the right hand operator condition is true* ***(!)***
* *AND --- true only when both the conditions are true* ***(&&)***
* *OR --- true if any one of the condition is true* ***(||)***

**BITWISE OPEATOR**

* *Opeartions will be performed bit by bit*

*&,| ,^*

* *Cannot declare result variables as byte because result value will be automatically converted to int.(lossy conversion error)*