**VARIABLE:**

Declaration:

Datatype var;

Initialization:

Var= value;

Utilization:

s.o.p.(ln);

Operator :

* Is a special symbol or keyword i.e., used to designate mathematical operation or some other type of operation .
* These operations can be performed on one or more than one values called as “operands”.
* Arithmetic Operators:
* (+,-)🡪**additive**
* **(\*,/,%) 🡪multiplicative arithmetic**
* Assignment Operator: “also known as compound assignment operators”
* i+=j;
* i-=j;
* i\*=j;
* i%=j;
* i/=j;
* first arithmetic operator is working later the result is assigned to the variable which is on left-hand-side.
* Increment ‘++’/ Decrement Operator ‘--’:

|  |  |
| --- | --- |
| Pre-Increment | Post-Incerement |
| Int i=10; | Int j=10; |
| I++ | J++ |
| 1. Increment | Substitute |
| 1. Substitute | Utilize |
| 1. Utilize | 1. increment |

Relational/comparison Operators:

We`ll see the relation b/w two values using Rel.OP.

* == returns true when both sides of the equation are equal else returns false
* != returns true when both sides of the equation are not equal else returns false
* > returns true if the left side of the equation is greater than the r.h.s else returns false
* < returns true if the left side of the equation is less than the r.h.s else returns false
* >= returns true if the left side of the equation is greater than or equals the r.h.s else returns false
* <= returns true if the left side of the equation is less than or equals the r.h.s else returns false
  + - The return of the relational expression is always the bolean value.

Logical Operators:

1. NOT -! –unary-(we can use it with a single operand)-return alse if the operand right to the operator is true and viceversa
2. AND-&&-if both operands are true then it is true.
3. OR- ||-if any of the operands results true then it is true.
4. XOR- if both values(0,1) are similar then it is false

If the values are different then it gives true

***Control Flow Statements:***

* If:

CONDITION

CODE /STMNTSS OUTSIDE IF BLOCK

CODE /STMNTSS INSIDE IF BLOCK

Else-if:-

CONDITION

true false

CODE /STMNTSS OUTSIDE IF BLOCK

Code inside elseif block

CODE /STMNTSS INSIDE IF BLOCK

Code inside elseif block

LOOPING STATEMENTS:-

FOR:- FOR(INITIALIZATION;TEST CONDITION;INCREMENT/DECREMENT)

CODE

INTIALIZATON

CONDITION

CODE /STMNTSS inSIDE FOR

INC/DEC

CODE /STMNTSS OUTSIDE IF BLOCK

***While:-***

**Write a program using control flow statements of and looping stmnts wr in the user can print his name once if the days of the week are weekdays**

**Print 5 times if it is weekend**

***Methods:***

***Dt:23-aug-19***

Arrays:-

***Def :- group of homogeneous data that has ,some index and fixed data.***

Index of an array will always starts from zero

**Ways of declaring arrays:**

* + - * int []array;
      * int[] array1;
      * int array2[];
* Array IndexOutOfBoundsException