stdin stdout

cin cout

System.in System.out

Keyboard program monitor

for(. . .) {

. . .

for(. . . ) {

. . .

break;

} // end of inner loop

} // end of outer loop

abc: for(. . .) {

. . .

xyz: for(. . . ) {

. . .

break abc;

} // end of inner loop

} // end of outer loop

If(condition) { true

statements

}

Else {

Statements

}

While(condition) { true

Statements

}

Variable initialization

While(condition) {

Executable Statements

Variable increment/decrement

}

Variable initialization

do {

Executable Statements

Variable increment / decrement

} While(condition);

For(variable initialization; condition; incr/decr) {

Statements

}

Operators:

1. Arithmetic +, -, \*, / (div), % (modulo div)
2. Relational <, >, <=, >=, !=, ==
3. Logical && (AND), ||(OR), ! (NOT)
4. Assignment =
5. Increment ++

Decrement --

1. Conditional ?:
2. Bitwise & (AND), | (OR), ~ (NOT), ^ (Ex-OR), << (left shift), >> (right shift), >>>
3. Special , and instanceof

If x = 5 / 2 then X = 2

If x = 5 % 2 then x = 1

= & = =

A = 10; // assigns value 10 to A

A == 25 // compares the value of A i.e., 10 with 25

&&

C1 C2 C1 && C2 C1 || C2

T F F T

F T F T

F F F F

T T T T

!True - False

!False - True

++

Pre increment Post increment

++a a++

(a = a + 1) (a = a + 1)

A = 10; A = 10;

B = ++A; B = A++;

1. A = A + 1 => A = 11 B = A => B = 10
2. B = A => B = 11 A = A + 1 => A = 11

Conditional - ?:

(condition) ? true part : false part;

if condition is false

c = (a > b) ? a : b

if condition is true

Bitwise operators : operate on individual bits of the given operand

&

8 & 10 - 8

1 0 0 0

1 0 1 0

1 0 0 0

8 | 10 - 10

1 0 0 0

1 0 1 0

1 0 1 0

~1 - 0

~0 - 1

^ (Exclusive-OR) - ON when both bits are different & OFF when both bits are same

8 ^ 10 - 2

1 0 0 0

1 0 1 0

0 0 1 0

<< (left shift) : shifts specified no. of bits from left of the given operand and fills that many zeros to the right

If x = 11010111 then x << 3

1 1 0 1 0 1 1 1

out in

1 0 1 1 1 0 0 0

8 << 2 in a 32-bit machine

8 -> 0000 0000 0000 0000 0000 0000 0000 1 0 0 0

Then 8 << 2

0000 0000 0000 0000 0000 0000 0000 1 0 0 0

1 0 0 0 0 0

Therefore 8 << 2 will be 32

8 << 3

Given No. \* 2 power no. of bits

8 \* 2 power of 3 => 8 \* 8 = 64

8 >> 2

Given No. / 2 power no. of bits

8/ 2 power of 2 => 8 / 4 = 2

Vehicle v = new Car();

Vehicle

Car, Bike, Auto, Bus

Car c = getVehicle(); X

Vehicle vh = getVehicle( );

If(vh instanceof Car)

Car c = (Car)vh;

Else if(vh instanceof Bike)

Bike b = (Bike)vh;

0 or More times atleast once irrespective

based on condition of condition

**while do . . . while**



