DAY-9

21 June 2023 16:13

- ARITHMETIC OPERATOR
- RELATIONAL OPERATOR
- LOGICAL OPERATOR
- BITWISE OPERATOR
- SHIFT OPERATOR
- TERNARY OPERATOR
- UNARY OPERATOR

BITWISE OPERATOR

- JAVA DEFINES SEVERAL BITWISE OPERATORS , WHICH CAN APPLIED TO THE INTEGER TYPES , LONG, INT, SHIRT, CHAR, AND BYTE
- BITWISE OPERATORS ON BITS AND PERFORMS BIT-BY-BIT OPERATION
- TYPES OF BITWISE ARE :
    - BITWISE and [&]
    - BITWISE or [|]
    - BITWISE not [~]
    - BITWISE EXOR [^]

- ASSUME IF a = 60 AND b = 13.
- NOW IN BINARY FORMAT THEY WILL BE FOLLOWS -
    - a = 0011 1100
    - b = 0000 1101

BITWISE AND [&]

A	B	Y = A.B
0	0	0
0	1	0
1	0	0
1	1	1

A	0	0	1	1	1	1	0	0
B	0	0	0	0	1	1	0	1
Y	0	0	0	0	1	1	0	0

BITWISE AND OF 60 & 13 IS 12

BITWISE OR [|]

A	B	Y = A+B
0	0	0
0	1	1
1	0	1
1	1	1

A	0	0	1	1	1	1	0	0
B	0	0	0	0	1	1	0	1
Y	0	0	1	1	1	1	0	1

BITWISE OR OF 60 | 13 IS 61

BITWISE EXOR [^]

A	B	Y = A^B
0	0	0
0	1	1
1	0	1
1	1	0

A	0	0	1	1	1	1	0	0
B	0	0	0	0	1	1	0	1
Y	0	0	1	1	0	0	0	1

BITWISE EXOR OF 60 ^ 13 IS 49

BITWSIE NOT [~]

A	~A
0	1
1	0

A	0	0	1	1	1	1	0	0
~A	1	1	0	0	0	0	1	1

NOT OF A , i.e ~60 IS 195

LOGICAL OPERATORS

- THESE OPERATOR PERFORMS LOGICAL OPERATIONS ON THE OPERANDS
- THE OUTPUT OF THERE OPERATORS WILL ALWAYS BE IN BOOLEAN I.E TRUE OR FALSE
- TYPE OF LOGICAL OPERATORS ARE
  - LOGICAL AND [&&]
  - LOGICAL OR [||]
  - LOGICAL NOT [!]

LOGICAL AND [&&]

- IF BOTH THE OPERANDS ARE NON-ZERO OR TRUE , THEN THE CONDITION BECOMES TRUE

A	B	Y = A.B

X = 5, Y = 1, Z = 3

A	B	Y = A.B
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

X = 5, Y = 1, Z = 3

OPERAND 1 && OPERAND 2

(X>Y) && (Y >Z)

TRUE && FALSE = FALSE

LOGICAL OR [||]

- IF ANY ONE OF THE OPERANDS ARE NON-ZERO OR TRUE , THEN THE CONDITION BECOMES TRUE

A	B	Y = A+B
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

X = 5, Y = 1, Z = 3

OPERAND 1 || OPERAND 2

(X>Y) || (Y >Z)

TRUE || FALSE = TRUE

LOGICAL NOT [!]

- USE TO REVERSE THE LOGICAL STATE OF ITS OPERNAD
- IF A CONDITION IS TRUE THEN LOGICAL NOT OPERATOR WILL MAKE FALSE

A	!A
FALSE	TRUE
TRUE	FALSE

X = 5, Y = 1, Z = 3

!(OPERAND 1 || OPERAND 2)

!((X>Y) || (Y >Z))

!(TRUE || FALSE) =!(TRUE) = FALSE

SHIFT OPERATORS

- THESE OPERATORS PEFROM THE OPERATION BINARY DATA I.E 0'S AND 1'S
- THEY MOVE THE BITS DEPENDING ON WHICH TYPE OF OPERATOR IS BEING USED.
- THERE ARE TWO TYPES OF SIFT OPERATOR AND THEY ARE
    - LEFT SHIFT OPERATOR [<<]
    - RIGHT SHIFT OPERATOR [>>]

LEFT SHIFT OPERATOR

- THE JAVA LEFT SHIFT OPERATOR [<<] IS USED TO SHIFT ALL OF THE BITS IN A VALUE TO THE LEFT SIDE OF SPICIFIED NUMBER

OF TIMES

- SYNTAX :

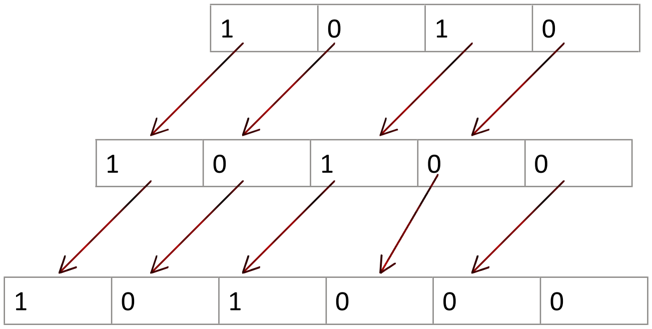
  OPERAND 1 << OPERAND 2

- WHERE, OPERAND 1 IS ON WHICH THE OPERATION PERFROMED AND OPERAND 2 SPECIFIES THAT HOW MANY TIMES THE OPERATION SHOULD BE PERFORMED.

EXAMPLE :

10 << 2

- HERE FIRST 10 IS CONVERTED INTO BINARY I.E , 1010
- THEN THE BITS ARE MOVED TO LEFT POSITION



NOTE : IF WE CONDUCT LEFT SHIFT ON A NUMBER THEN THE VALUE WILL BE MULTIPLIED BY 2

RIGHT SHIFT OPERATOR

- THE JAVA LEFT SHIFT OPERATOR [>>] IS USED TO SHIFT ALL OF THE BITS IN A VALUE TO THE RIGHT SIDE OF SPICIFIED NUMBER OF TIMES
- SYNTAX :

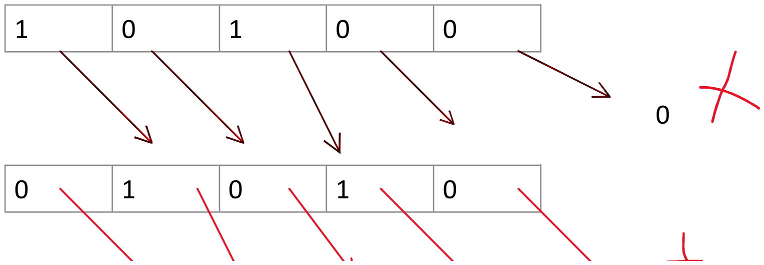
  OPERAND 1 >> OPERAND 2

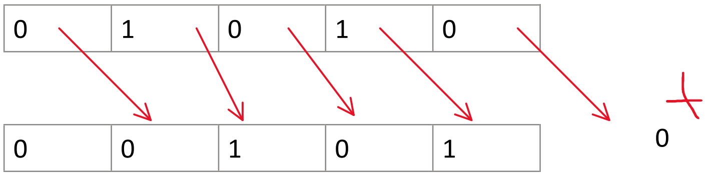
- WHERE, OPERAND 1 IS ON WHICH THE OPERATION PERFROMED AND OPERAND 2 SPECIFIES THAT HOW MANY TIMES THE OPERATION SHOULD BE PERFORMED.

EXAMPLE :

20 >> 2

- HERE FIRST 20 IS CONVERTED INTO BINARY I.E , 10100
- THEN THE BITS ARE MOVED TO LEFT POSITION





0	1	0	1	0	
0	0	1	0	1	0

NOTE : IF WE CONDUCT RIGHT SHIFT ON A NUMBER THEN THE VALUE WILL BE DIVIDED BY 2

X = 10

OPERATIONS	ANSWERS
X >> 1	5
X >> 2	2
X >> 4	0
X << 1	20
X << 2	40
X >> 3 << 3	8
(X>>3 <<3) + (X<<2)	

10/2 = 5

5 /2 = 2.5

2/2 = 1

1*2 = 2

2*2 = 4

4*2 = 8

48