BITWISE OPERATORS:- it is used to perform bits to manipulate the integer value.
SYNTAX:- &,|,^,~,<<,>>.

1) (&)AND OPERATOR:- it accepts only when both the conditions are true value (in binary form).
CONDITIONS:-{0-flase
1-true}
EXAMPLE:- binary code for 4 is 0100
binary code for 3 is 0011
the resultant of 4&3 is 0000 (in binary code), 0 (in integer value).

2) (|)OR OPERATOR:- it doesn't accepts only the false condition.
EXAMPLE:- binary code of 4 is 0100
binary code of 3 is 0011
the resultant of (4|3) is 0111 (in binary code), 7 (in integer value).

3) (^)XOR OPERATOR:- it accepts if one operand is false and another operand is true.
EXAMPLE:- binary code of 4 is 0100
binary code of 3 is 0011
the resultant of (4^3) is 0111 (in binary code), 7 (in integer value).

4) (~)NOT OPERATOR:- takes the opposite value of single operand in binary code.
EXAMPLE:- binary code of 4 is 0100
the resultant of ~(4) is 1011 (in binary code), ~(11) (in integer value).
or (~N)=-(n+1) i.e., n=4
-(4+1) = -5

5) (>>)RIGHT SHIFT:- removes the binary code value for right.
EXAMPLE:- binary code of 4 is 00000100 (binary code in 8 bytes)
the resultant of (4)= 00000100
the resultant of (4>>2)= 00000001

6) (<<)LEFT SHIFT:- adds the values in the binary code from right.
EXAMPLE:- binary code of 4 is 00000100 (binary code in 8 bytes)
the resultant of (4)= 00000100
the resultant of (4<<2)= 0000010000

TERNARY OPERATORS:- * the first os the camparison aurgument.
* the second is the result upon the true comparison.
* the third is the result upon the false comparison.
SYNTAX:- (? :)
EXAMPLE:- (a?b:c) evaluates to b if the value of a is true, or otherwise to c.