INTRODUCTION TO PROGRAMS AND JAVA

1. **Special Operators:** There are few other operators supported by java language as explained below:
   a) **Instanceof operator:**
   - ❖ This operator is used only for object reference variable. The operator checks whether the object is of a particular type (Class type or interface type). Instanceof operator is written as:
     (Object reference variable) instance (class / interface type)
   - ❖ If the object on the left side of the operator is an object of right side, then the result will be true.

   Example:

```
Public class Test
{
     Public static void main(string args [] )
     {
          String name = "James";
          // following will return true since name is
type of string
          Boolean result = name instanceof string ;
          System.out.println(result);
     }
}
Output:  True
```

   b) **Dot operator:** Dot operator (.) is used to access class members i.e. methods and data.

2. **Bitwise Operators:**

- ❖ Java defines several bitwise operators, which can be applied to the integer types, long, int, short, char, and byte. Bitwise operator works on bits and performs bit-by-bit operation.
- ❖ Assume if a = 60; and b = 13; now in binary format they will be as down the table and its operation:

❖ The following table lists bitwise operators: assume integer variable A holds 60 and variable B holds 13.

Sr. No.	Operator	Description	Example
1.	& (Binary And Operator)	Copies a bit to the result if it exists in both operands.	(A & B) will give 12 which is 0000 1100
2.	|(Binary OR operator)	Copies a bit if it exists in either operand	(A | B) will give 61 which is 0011 0001
3.	^ (Binary XOR operator)	Copies the bit if it is set in one operand but not both	(A ^ B) will give 49 which is 0011 0001
4.	~(Binary Ones(1's) Complement Operator)	Unary and has the effect of flipping bits.	(~A) will give -61 which is 1100 0011 in 2's complement form due to a signed binary number.
5.	<<(Binary left Shift Operator)	The left operands value is moved left by the number of bits specified by the right operand	A << 2 will give 240 which is 1111 0000
6.	>>(Binary Right Shift Operator)	The left operands value is moved right by the number of bits specifies by the right operand	A>> 2 will give 15 which is 1111
7.	>>>(Shift right zero fill operator)	The left operands value is moved right by the number of bits specified by the right operand and shifted by the right operand and shifted values are filled up with zeroes.	