OPERATOR AND EXPRESSIONS

Arithmetic operator and expressions:

- +, -, *, /, %.
- These operators can be applied on any type of data except Boolean.
- % operator works on float and double data types without taking the decimal point in quotient but the divisor and dividend contain the decimal point.
- To write down the expression one should be familiar with the precedence of operators.
- *, /, % have higher precedence than +, -.
- To change the precedence of operators () are used.
- The mechanism of converting data type internally by the compiler is known as “COERCION”.
- Data types of expression:
    - The addition, subtraction, multiplication or division between the byte, short and int results as int as they belong integer type of data.
    - Byte and short are provided by java for backward compatibility.
    - If float is used in the expression then the result is float.
    - If Double is used in the expression then the result is double.
    - When char is used the result is int type as char is part of int(codes).

Increment / decrement operators and expressions:

- Post++, post- -, ++pre, - -pre.
- Pre and Post operators both work in same way.

- In pre increment/decrement first the value is incremented/ decremented and then utilized.
- In post increment/decrement first the value is utilized and then incremented/decremented.
- The data types remain the same on application of these operators.
- These operators cannot be applied to Boolean type of data.

Bitwise operators:

- These perform operations on bits of data.
- These operations are faster compared to other operators.
- Bitwise operators available in java are:

→ AND &

- result is true if both are true.

→ OR |

- result is true if any one of the values is true.

→ NOT ~

- result is inversed.

→ XOR ^

- result is true if any one of it is true.

→ Right shift >>

- numbers are shifted to right by one unit and the empty space is filled by '0'.

→ Unsigned right shift >>>

- this operator extends the right shift operator.

→ Left shift <<

- numbers are shifted to left by one unit and the empty space is filled by '0'.

➢ These operators can be performed on only integer type of data.

Bit Merging and Masking:

➢ One of the aplication of bitwise operators.

➢ They are used to check whether the bits are marked as zero or one.

➢ The process of setting one of the bit as '1' is known as merging (using or operator).

➢ To check if the bit is on or not by using and operator is known as masking.

➢ 4 bits is also known as nibble.

➢ XOR operation is used for Swapping two numbers without inserting any third number.

Widening and Narrowing:

➢ It is related to conversion of data type of given data item.

➢ Widening: the process of storing the data of smaller size data type into a larger size data type can be done automatically by the compiler.

- The compiler directly converts the data type without showing any errors i.e from source data type to the destination data type.
- Since it is done internally it is said as automatically.
- The source and destination data type should be compatible.
- it is also known as upcasting.

➢ Narrowing: the process of storing the data of larger size data type into a smaller size data type.

- due to difference in size it may lead to loss of data.

- it is not compatible.
- typecasting is required for narrowing.

➢ Boolean is compatible only with Boolean.