**Core Java - Assignment**

**Module - 1**

1. **Data types , Variables and Operators :**

**Theory : -**

1. Primitive data-types in Java :

->Primitive data types are the most basic data types in Java. They are used to store simple values like numbers and characters. Unlike non-primitive data types (objects), primitive data types are not objects and do not have methods associated with them.

-> There are eight primitive data types in Java:

1. Byte : 8-bit signed integer. It ranges from -128 to 127.
2. Short : it take 16 bits to store value and it’s range belongs to -32,768 to 32,767.
3. Int : it will occupies 32 bits to store value and it can store from -2,147,483,648 to 2,147,483,647.
4. Long : it takes 64 bits to store big Int values in -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
5. Float : it will occupies same memory as Int but difference is you can store decimal value in float where in Int it is not possible. Like 3.14 pi it is decimal value.
6. Double : it is same as long but it can store more precision floating value.
7. Char : it takes 16 bits to store any character but the thing you can store only single character.
8. Boolean: it only stores true or false value.
9. Variable declaration and Initialization :

-> Declaring a variable means defining data-type with its name which allocates memory to your variable.

-> Initialization means to assign some value when you declare a variable and store it.

For example, int a = 10 ;

Where int is a data-type and a is memory location where 10 is stored.

1. Operators : Arithmetic, Relational, Logical, Assignment, Unary & Bitwise.

-> Operators are used when you want to do some operation on your variable which will give you some result in the end.

-> Arithmetic operator : it is basic mathematical operator which you do in daily life when go to purchase something from shop.

1. Addition ( + ) : generally, it will take 2 operand and give result from it.
2. Subtraction ( - ) : It will do same as addition but the difference is it will minus 1 value from second value.
3. Multiplication ( \* ) : It will take 2 operand and do some task with it and give you a result.
4. Division ( / ) : It will divide one value from another value.
5. Modulo ( % ) : it will give you reminder when you use it one variable.
6. Exponentiation ( \*\* ) : it basically means like power operator In maths we use 24 means 16.

Where 2 is operand and 4 is power that it means to multiply 2 to 4 times it will gives 16 as output.

-> Relational Operator : it is used to compare two values .

1. Equal to ( == ) : it will check both value are equal or not.
2. Not Equal to ( != ) : it is used when both values are not equal.
3. Greater Than ( > ) : It is used for check one value is greater than another value.
4. Less than ( < ) : It is used when one value is smaller than another one.
5. Greater than Equal to ( >= ) : it is used when you want compare value which greater than or equal to that value.
6. Less than Equal to ( >= ) : it is used when you want compare value which less than or equal to that value.

-> Logical Operator : It is used when you want to combine two or more condition at one time.

1. AND operator (&&) : when you want that not only one but when both condition match on that time you should use this operator it will execute.
2. OR operator ( || ) : From multiple condition you at least one condition get true on that time you want to execute block of code you should use OR operator.
3. NOT operator ( ! ) : It will return opposite result of your condition like when you want when condition get wrong and you want to turn it in true then you should use ! operator to change the result from false to true or true to false.

-> Assignment Operator :

1. Plus equal to ( += ) : it is used to when you want to add value and assign value at the same time you can use it.
2. Multiply equal to ( \*= ) : when you want to multiply then assign to another variable.
3. Minus equal to ( -= ) : it is used to subtract some value and assign to other variable.
4. Division equal to ( /= ) : It is used to assign after dividing some value.
5. Modulo equal to ( %= ) : if you want assign the reminder value to another variable.

-> Unary Operator : it only need single operand.

1. Increment : when you want increase value by 1 and it only needs one operand.
2. Decrement : when you want decrease value by 1.

-> Bitwise Operator : when you want to perform operations on bits level.

1. Bitwise AND (&) : it is used to perform bitwise and operation.
2. Bitwise OR ( | ) : it performs bitwise or operation
3. Bitwise XOR ( ^ ) : it performs bitwise XOR operation.
4. Left Shift ( << ) : it is used to when you want to shift bits on left side.
5. Right Shift ( >> ) : it is used to shift bits on the right side.

4. Type Conversion and Type Casting : when you want to convert one datatype to another datatype during that period we use type conversion or you can say type casting.

-> Type conversion : Most probably compiler compiles the code and there is not problem in code like converting one smaller datatype to bigger datatype. It will be done during compilation . But sometime you have to explicitly change datatype of your code.

-> In simple language when compiler change datatype of variable then it is called type conversion. But when programmer explicitly let code know that it should change the datatype of particular variable. Then it is called type casting.