**Methods**

**06-03-2025**

**================**

OOPs (Object Oriented Programming System)

---------------------------------------------------

concepts:

---------

 class, object, methods, constructor etc.

principles

----------

 Encapsulation, Inheritance, Polymorphism and Abstraction

class:

-----

 -> reference datatype

 -> collection of data (variables) and methods is called as "Class".

Ex:

 class WhileLoop{

 public static void main(String[] args)

 {

 }

 }

Object:

------

 -> reference to class.

 -> to access anything from the class, we need to define object.

Ex: Scanner class --> Input statement

 while taking the value for the variable in run-time (dynamic variable), we need "Scanner" class

Scanner class ==> pre-defined class

 in "util" package

import java.util.Scanner;

main(String[] args){

 Scanner scan = new Scanner(System.in);

}

class\_name object\_name = new class\_name();

Function Vs Method

----------------------

-> According to the syntax, function and method both are similar

but the difference is:

 the function can always define outside the class

and methods can always define inside the class.

Note:

----

Java does not support functions. Because to execute the single line of code, we must be required class. Function need not required classes. So, functions are not fundamental body for java program.

What is method?

------------------

-> method is a named block.

-> Block is the collection of statements must be enclosed with {}

-> method consisting of one or more than one statement to perform the specific functionality.

-> Methods can accept the data, process the data and also can return the data.

how to define the methods?

-------------------------------

class MyClass{

 public static void main(String[] args)

 {

 method()

 }

 public/private/protected static method(){

 }

}

class Class1{

 void m1(){

 }

}

Syntax:

 public/private/protected static return\_type method\_name(type data1, type data2,...)

 {

 // Method Body

 statement-1;

 statement-2;

 return value;

 }

here:

 -> according to application requirement, the method may return a value.

 Ex: if method want to display account balance (double type)

 that method might be defined with "double" as return type

 Syntax:

 double accountBalance(){

 return value;

 }

 to make return an age of the user from method, we can use as below:

 int returnAge(){

 return age;

 }

 -> If the method not to return anything, then we can use the return type as "void".

Note:

-----

method can able to execute for performing the task/functionality but not by its own.

-> We must be invoke/call that defined method to make it execute. That is called as "method calling".

Why we need method?

-------------------------

1) to improve the readability of the program

Ex: class Banking{

 main(){

 }

 check balance(){

 }

 transfer money(){

 }

 pay slips

 Credit

 loans

}

2) for the fast development with modular approach we need methods.

types of methods?

---------------------

-> Data --> parameter/argument

-> methods can be defined in 4-ways:

 1) Methods with no arguments and no return type

 2) Methods with no arguments and return type

 3) Methods with argument and no return type

 4) Method with argument and return type

Syntax for the method calling:

 method-name(val1, val2, val3,..);

-> To which method we are calling is called as "calling method"

from which method we are calling the calling method is called as "called method".

class SimpleCalculator{

 public static void main(String[] args)

 {

 addition();

 int result1 = subtraction();

 System.out.println("The subtraction of two numbers = "+result1);

 multiplication(12,13);

 int a = 10, b = 20;

 multiplication(a,b);

 System.out.println("The Quotient after the division is = "+division(10,3));

 }

 // method with no argumnets and no return type

 public static void addition()

 {

 int a = 100;

 int b = 200;

 int c = a + b;

 System.out.println("The sum of two numbers = "+c);

 }

 // method with no arguments and return type

 private static int subtraction(){

 int a = 1022;

 int b = 512;

 int d = a -b;

 return d;

 }

 // method with arguments and no return type

 protected static void multiplication(int a, int b){

 int c = a \* b;

 System.out.println("The multiplication of two numbers is = "+c);

 }

 // method with arguments and return type

 static int division(int a, int b){

 return a/b;

 }

}