19 January Saturday 019-346 2019

Methods

specific piece of code which is use to call multiple time when needed instead of writing the whole code everytime.

Syntax-

```
<accessspecifier> <modifier> <datatype> methodName(
        list_of_parameter)
    {
        // body
    }
```

- modifier - It defines the access type of the method & it is optional.
- datatype - method may return a value.
- methodName - It is name of method. The method signature consist of method name & the parameter list.
- method body - body defines what the method does.

Method Overloading - When a class has two or more methods by the same name but diff. parameters, it is known as method overloading.

Ex- int max(int a, int b)
{
}

int ~~float~~ max(float a, float b)
{
}

int ~~String~~ max(String a, String b)
{
}

Pass by Value - The method parameter values are copied to another variable & then the copied object is passed.

Pass by Reference - An reference to the actual parameter is passed to the method.

Constructors

- Similar to Method
- No return type
- Obje Method name same as class name.
- Constructor invoked during object call.

Example-

```
class Prac {
    Prac()
    { // body
    }
}
```

Default Constructor - In this the default value of the used data type is assigned to the variable.

Parameterized Constructor - When value is assigned in constructor during function call.

```
class Drive {
    String v; //string Vehicle
    Drive(String vehicle)
    { v=vehicle; // this.vehide=vehicle
    }
```

Default cons. can not be called if parameter is made.


```
public static void main(String args[])
{
    Drive obj = new Drive("Car");
    System.out.println(obj.v);
}
}
```

Constructor Overloading- Two or more constructor different in parameter.

```
class Drive {
    String vehicle;
    int speed;
    Drive()
    {
        vehicle = "";
        speed = 0;
    }
    Drive(vehicle)
    {
        this.vehicle = vehicle);
    }
    Drive(vehicle, speed)
    { }
}
```

2019

January
Saturday
026-339

26

Static Keyword -

The Keyword static indicates that the particular member belongs to a type itself, rather than to an instance of that type. It is mainly used to help memory management. The Keyword can be applied to variables, methods, blocks and nested class.

Static class -

We can not declare a top-level class with static modifier, but we can declare a ~~method~~ nested class as static.

In Java, class can be define within a class, known as nested class, are of two types - static & non-static.

```
class <OuterClass>{
---
static class <StaticNestedClass>{
---
}
class <Inner class>{
---
}
}
```

* Object creation of non-static inner class -

```
A objA = new A();
B objB = objA.new B();
```

object of outer class is necessary for accessing inner-class.

* Object creation of static inner class -

```
C objC = new A.C();
```

Static Block -

These are the piece / block of code in a program which executes before the main functions starts execution.
(सबसे पहले ये चलेगा तब कुछ और होगा)

```
static {
    - - -
    - - -
}
```

These are written outside the main().