15-06-24.

METHOD OVERLOADING :- Reusing same name for different methods with different number of arguments [OR] with different type of arguments with [OR] without same datatype return type within a class.

→ It is a static polymorphism [OR] compile time polymorphism

EX:-

→
```
package sreeja;
public class Demo {
    void add (int a, int b)
    {
        System.out.println(a+b);
    }
    void add (float a, float b)
    {
        System.out.println(a+b);
    }
    public static void main (String[] args)
    {
        Demo p = new Demo();
        p.add(5,6);
        p.add(2.5f, 5.2f);
    }
}
```

O/P :- 11

7.7

this keyword -> Members & Methods [GLOBAL]

this ( ) -> Constructor [this method]

To call Member / Method Create a object

instance class variable

STATIC :- It is a keyword of JAVA. It can be placed

-> Before a member
-> Before a method
-> Before a class

1. BEFORE A MEMBER :-

STATIC MEMBER :- If we place static keyword before a member is called static member.

Ex :- static int a;

An objectless member is called Static member.

Only 1 copy of member is created for an entire class eventhough we have any number of objects.

If we effect in 1 object it reflects in another object.

It is created before main execution start.

It can be accessed :-

* without object name
* with object name
* with class name

Ex:-

```
-> package sreeja;

public class Demo {
    int a;                  // class
    static int b;           // object
    public static void main (String[] args)
    {
        Demo p = new Demo ();
        Demo q = new Demo ();
        Demo r = new Demo ();
        p.a = 1; q.a = 2; r.a = 3;
        System.out.println (" " + p.a + q.a + r.a);
        p.b = 10;
        System.out.println (" " + p.b + q.b + r.b + Demo.b);
    }
}
```

O/P:-

1 2 3

10 10 10

STATIC METHOD :- An objectless method is called static method.

SYNTAX :-

```
static returntype name (--)
{

}
```

It can be accessed with objet name, without objet name, with class name.

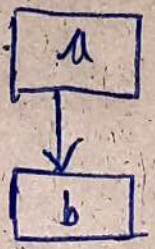
⇒
```
package sreeja;
public class Demo
{
    static void display (-)
    {
      System.out.println ("student are not participating");
    }

    public static void main (String [] args)
    {
       Demo.display ();
    }
}
```

O/P :- student are not participating

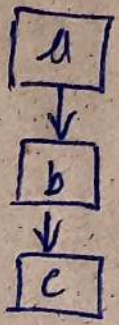
INHERITANCE :- It is a process by which 1 class objects acquires the properties of another class object.

TYPES OF INHERITANCE =>

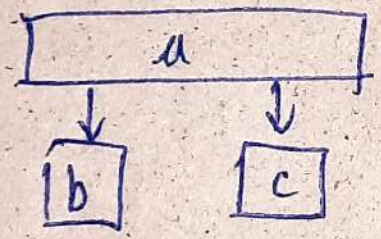
1. SINGLE :-



2. MULTILEVEL :-



3. HIRARCHIAL :-



4. MULTIPLE :- Not Possible X