finally

Finally is a keyword used in exception handling to define a block code that will always be executed whether an exception is thrown or not.

~~We also use the~~

We also use finally block to try-catch for handling the exception.

Syntax :

```
finally {
// code }
```

Syntax with try-catch

```
try { //code }
catch { //code }
finally { //code }
```

Only with try

```
try {
    //code }
finally {
    //code }
```

Note :
We use finally block in only error handling with try-catch block.

Page No.
Date

Example :-

```
class Addition {

    public int sum (int a, int b) {

        try {
            int c = a+b;
            return c; }
        finally {
            Sout(" program is end but
                finally will be run "); }

Class Main {

psvm (String [] args) {

    Addition a = new Addition ();

    int store = a.sum(3,4);

    Sout (c); } }
```

using try for use the finally

method return the value Method is end but finally will be run automatically

object of Addition class

call and return the sum

Output :-
program is end but finally will be run
9

STATIC KEYWORD

main method

```
class Main {
    public static void main (strings [] args) {
        // code
    }
}
```

Access modifier

not return

parameters

what ?

The 'static' keyword in the 'main' method allow the Java program to start running without needing to create an object of the class.

They are accessed using the class name and non-static accessed by the obj of the class

GYAN --

Aapne dekha hoga ki hum agar main class ke andar koi method bnate h to hame us method ko static bnana padta h, varna agar hum us method ko static nahi bnayenge to hame apni main class ka ek obj banana padega usse access karne ke liye.

Ex in code on the next page.

Page No.
Date

Ex →

Non-static Method

```
class Main {
    void display () {
        sout (" Display the world"); }
    public static void main (String [] args) {
        sout ("Hello world");
        Main a = new Main ();
        a.display ();        } }
```

create a object of the class

call the help of object

Static Method

```
class Main {
    static void display () {
        sout (" display the world"); }
    public static void main (String [] args) {
        sout ("Hello world");
        Main.display ();  → Use this
        OR
        display ();        }
```

call directly with the classname ()

This method work only main class

second method call directly only write the name of method.

GYAN

⇒ Hamara jo main method h vo ak static method h to static method ko static method se access kar sakte h bina koi object banaye par agar vo method non-static to hame us method ko access karne ke liye ak obj banana padega.

Ex :-

```
class library {
                    // Static Method
    static void book () {
        sout ("It is book"); }
                    // Non-Static Method
    void newspaper () {
        sout ("It is a newspaper") } }

class Main {
    psvm (string[]args) {
        library.book ();
        library obj = new library ();
        obj.newspaper ();
        obj.book () ;  } }
```

call non-static method directly by class name

create obj for call non-static meth. -04

try to call static method from obj
This is valid but are commended.

Static Block

Page No.
Date

~~only static~~

~~# [static] Method Method Class~~

syntax:

```
static {
    // code }
```

this static method is like a constructor its run automatically with any call.

agar hum isko tarah sirf static likh kar usko under code kar dete h to ye automatically chalega constructor ki tarah. bus us class ka ya to koi object bna ha ya to us class me se kisi auto simple static method ko call kari ho.

Ex:-

~~class other {~~
~~void show () {~~
~~sout ("show fn") ; }}~~

~~class Main {~~
~~psvm~~

~~Next page~~

Example on next page

Ex :-

```
class other {
    void display () {
        sout ("display the world"); }
    static {
        sout ("It is only static its run automatically"); }}

class Main {
    psvm (String [] args) {
        other obj = new other ();
        obj. display ( ); } }
```

This is a only static method its call automatically like a constructor

jaise hi hum ne is class ka obj banaya ye khud hi call ho gaya automatically

OUTPUT :-

It is only static its run automatically
display the world

First run static and second run the display method.

Page No.
Date

Static variable

We also create a static variable, ~~but we only access the variable through the static method~~

Note :- static method only static variable hi use kar sakte, but non-static static / Non-static dono use kar sakte h.

Ex :-

```
public class Main {
    static int a;
    int b;

    psvm (String[] args) {

        a = 5;
        // b = 5
        Main obj = new Main();
        obj.b = 5;
    }
}
```

static → static int a;

int b; ← Non-static

a = 5; ← then variable is static & use easily

// b = 5 Not possible Because b = non static variable

obj.b = 5; → but variable is not static we create object for access the variable

Variable

Local and Instance Variable

- **Instance Variable**

* An instance variable in java is a variable that declared within a class but outside on any method.

* Initialization : Instance variable are already initialized with default values, therefore not compulsory to initialization.

* Instance variables are accessible to all methods, constructor and blocks in the class.

- **Local variable**

A variable declared within a method is known as "Local variable".

* Initialization : Local variable must be initialized before they used. They not have default values like instance variable

* local variable are only accessible within the method which they are declared.

Page No.
Date

Example :- local and Instance variable

```
class Main {

    int a;  // Instance variable
                not be need to initialized

public static void main (String[] args) {

    int b = 0;  } }
```

Local variable, It's compulsory to initialized.

Object & Instance

GYAN

⇒ Suppose that hai koi, hai chij ak object Tum object ho me object hu, hum humans class ke instance member h. In dono words ka matlab similar but sye dono alag h jab bhi class ki bat hogi to instance ka use hoga.

Extra example — Audi car ak object Aur wo car class ki ak instance member h.

Overloading & Overriding

* Overloading

→ Define multiple methods in the same class with the same name but different parameters.
Allows methods to have the same name but perform differen task based on parameter.

* Overriding

- Providing a specific implementation of a method in a subclass that is already defined in its superclass.
- In overriding we use keyword @Override

Page No.
Date

Example of Overloading

```
class Calculator {

    int add (int a, int b) {
        return a+b ; }

    double add (double a, double b) {
        return a+b ; }

    int add ( int a , int b , int c) {
        return a+b+c ; }

}
```

Same name with different parameters in known as overloading

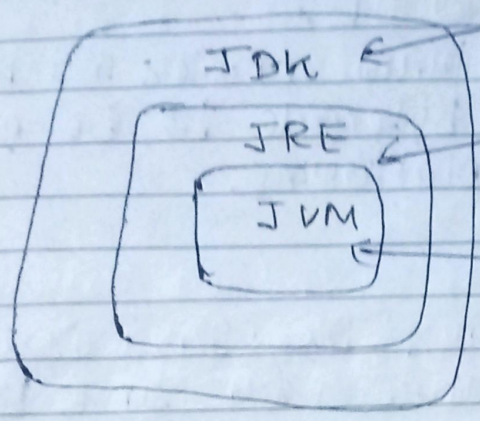
Example of Overriding

```
class Animal {
    void makeSound () {
        Sout ("Some generic sound") ; }}

class Dog extends Animal {

    @Override
    void makeSound () {
        Sout ("Bark") ; }}
```

Override the method

JAVA

JDK → JRE → JVM



Java development kit

Java runtime environment

Java virtual machine

* JDK (Java-Development kit)
→ JDK is a software development kit used to develop java application.
→ It includes java compile (Javac), Java runtime environment (JRE) and Java virtual machine.

* JRE (Java Runtime Enviroment)
→ JRE is a part of The JDK and its environment required to run java applications
→ In includes JDK, libraries and other necessary for executing Java byte byte code.

Page No.
Date

* JVM (Java Virtual Machine)
- The JVM is a important component of the JRE and JDK.
- It's a virtual machine that enables Java bytecode to be executed on various platforms.

~~Note :- JDK (Java Development Kit) iske andar hame JRE~~

Note : Java ka use karne ke liye hum JDK install karke jiski wajhe se java ke program run kar pate h.

JDK ke andar hi java compiler hota tha hum "javac" iska use karke hum program ko compile karke h aber hame ak bytecode file mil jati h.
Ab hum us file ko kisi bhi OS par execute kare to execute ho jayegi kyoki java platform independent hota h.
Sabhi OS me java compiler (javac) same hota h islive hame usse mili byte code file ko kabhi bhi execute kar sakte h.

Page No.
Date

suppose Ab hamne java compile ke me compile karke humne ek byte code file dedi.

Ab us byte code file ko execute karne ka kam (JVM) karta h. jo JRE ke andar hota h, ab apne program bnate time kuch files, library ko import kara hoga. To vo chij JVM ko to nahi pta To vo help lega JRE kyoki sari files JRE me hogi to JVM waha se file import karke pure program ko execute kare lega.

Extra
Ale OS me "javac" same hota hai par OS me bytecode file same generate hoti par JVM sabka alag hota h. har OS ka apni language me likha hota h.

Page No.
Date

Stack & Heap

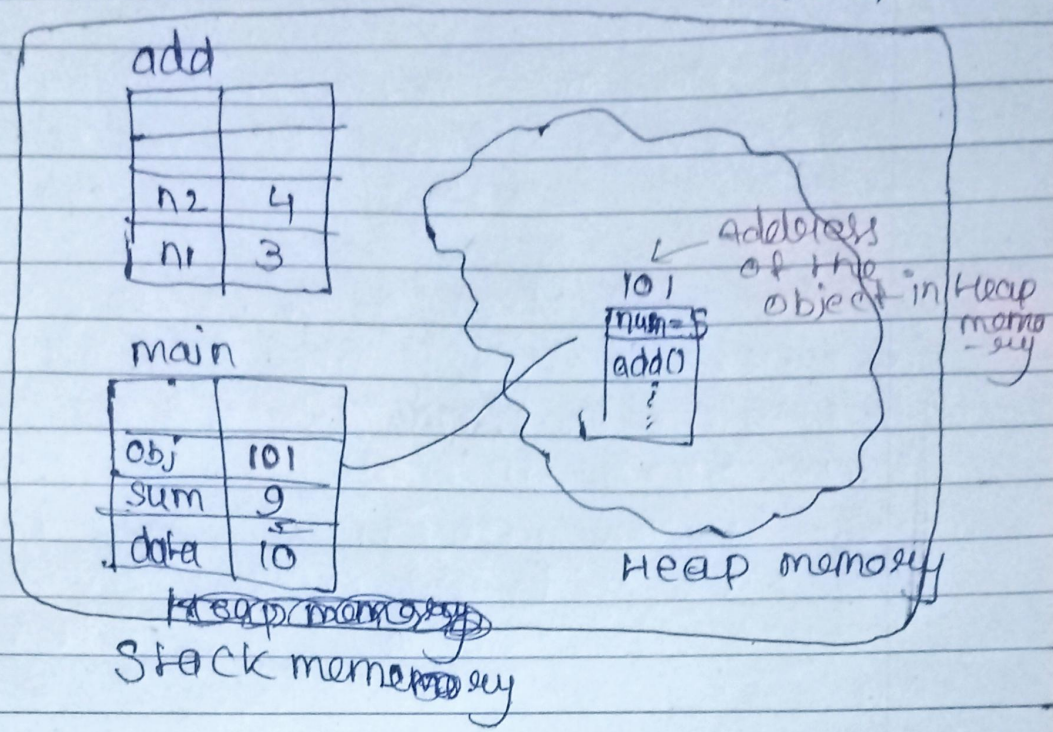
In this session I want teach you how to Java program allocate space.

```
class Calculator {
    int num=5; // Instance Variable
    public Int add (int n1, int n2) {
        return n1+n2; } }

public class demo {
    public static void main (String[] args) {
        int data =10;
        Calculator obj = new Calculator();

        int Sum = obj.add(5,4);

        sout(sum); } }
```

/* In this program we give to argument 5 and 4 to add method and add method return and return data store in sum variable. */

Page No.
Date

JVM



jo methods hote h vo unka data stack memory me save hota h par jo object hota h vo to pura class ki copy hoti h uska data to hum ak stack data address me save nahi sakte.

to jo object hote h unko hum heap memory me save karte h. 'phir heap memory me jo unko address hota h us reference address ko hum stack me save kar dete h.