-: JAVA Programming :-

- Java is Programming language which help us write codes to make the computer do a particular task.

* Java → is Mediator between human and computer

⇒ Java is high level OOP language, devloped by Sun Microsystem.

features

- Simple
- Object-oriente → Java based on objects concepted like, Inheritance, Polymorphism, abstract & encap are supported.
- Secure
- Platform Indepondent
- Multithreaded
- High Performance
- Dynamic

write code → (convert) byte code → read by special software calle ⇒ Java interpreter

Langage English

(not readble by humans)

Translat

Machine Friendly langage

Java used ⇒ Mobile Apps
- desktop Applicatin
- Robotics

* you need to store something in your computer shelf
  ⇓
  This shelf called Memory

Variable ⇒ storage container

; used for end a command.

int score = 100
  ↓ variable
→ only contain whole Numbers.

⇒ when we store the name

String name = "James" ;

(2) Boolean :- Used for Decision Making
~~The values are~~
Sometimes we require special kind of value to be stored, for decision Making.
The values are True & false
Boolean Value

→ key word use ⇒ Boolean

boolean decision = true ;

Conditional Statement ⇒ checks the condition

(1) if :- do task when condition is true
(2) else :- " " " " is fals

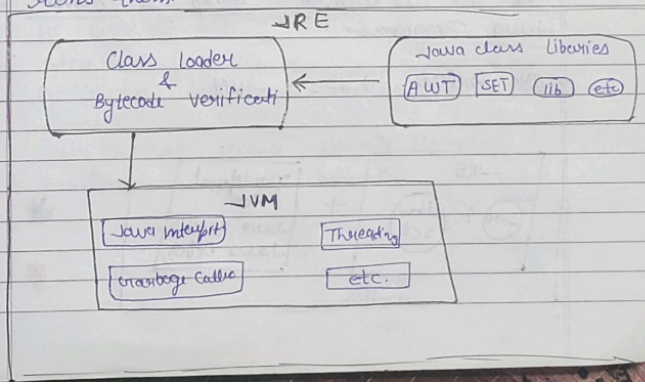
~~(<)~~ less than
(<)
(>) greater

JAVA Runtime Environ & Java Virtual Mach
JRE

→ Java Runtime Environment is the part of the Java Development kit.
JDK
→ It is the most common environment available on devices to run java Programs.

JRE = JVM + Java Libraries + Supporting file

* JRE cannot compile Java Programs ; it only runs them.



JVM ⇒ Java Virtual Machine

JVM is a virtual machine that runs Java bytecode.

⇒ JVM does

→ Converts bytecode ⟶ machine Code
→ Execute the Program
→ Handles Memory
→ Performs garbage collection

* JVM is the reason Java is Platform independent.

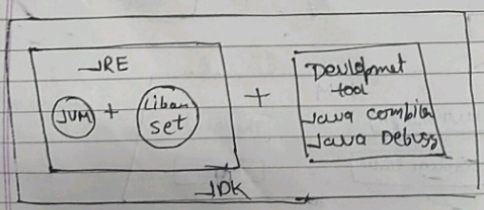
~~JDK (~~

JDK (Java Development kit)

JDK = JRE + Development tools

The JDK is used to develop & run Java Programs.

JDK include → JRE + JVM



Java and its Features. (1991 Sun Microsystem)

Java is the high level object-oriented Programming language. Its follow the WORA (Write once Run Anywhere) Principal.

The Major features are:

1- Platform independent

Java Code is compiled into bytecode, which can run on any os using JVM.

2- Object-Oriented

Java uses the concept of class & Object, interit & Polymorphism.

It makes Programs to understand & secure.

3- simple

Java removes complex features of C & C++ like
- Pointer
- No Multiple Inheritance.

4- Secure

Provide a secure execution environment.

5- Robust

Java Provides strong Memory Maygent & exeption handling.

JAVA APIS

Java API is a Collection of Predefind Classes and Packages Provided by Java for developers.

Example of Packages.

- java.lang → basic classes (String, Math)
- java.util → Scanner, Array list
- java.io → Input/output classes.

→ APIS saves time because Programmers do not need to write code from scratch.

:- Lexical Tokens :-

lexical tokens are the smallest Meaningful units in a Java Program.

Types of Tokens :-

1. keyword :- Reserved words
   Example- class, int, Public

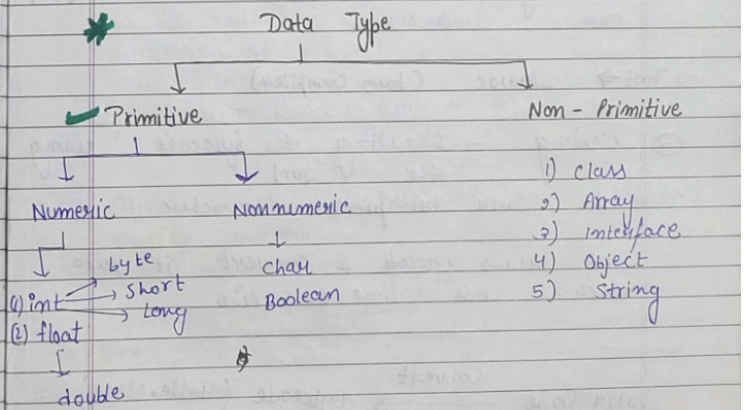
2. identifiers :- Name of Variables, classes, Methods
   Student Name, Main.

3- Literals :- Constan Values.
   Example: 10, 3.14, 'A', "Hello"

5- Operators :- Symbols used to Perform operations
   Example - +, -, *, &&

6- Separators :- used to Separate Code Components
   Example (), {}, ,, ;

Data type :→ data types defines the kind of data a Variable can hold



Compile & Run in Java :-

Java is a compiled + interpreted language. This means Java code goes through two main steps

① Compilation
② Running

Compilation :-

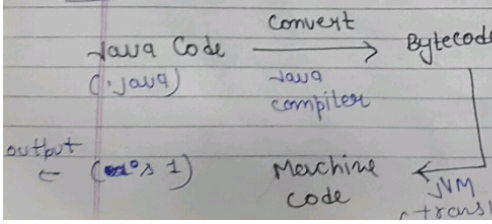
Compilation is the process of converting human readble java code (.java file) into bytecode (.class file) which the JVM can understand

→ Tool → Javac (Java Compiler)

② Running → Exculting the bytecode using the JVM

Tool → Java interpreter / Launcher

JVM reads bytecode & convert it into machine code line by line



Class & Object :-

In Java, classes & object are fundamental concept of OOP.

Class :-

⇒ A class in Java is a blueprint, template or design from which object are created. It defines

① Variables
② Functions

```
class Student {
    int rollNO;
    String name;
    void display() {
        System.out.Println (roll NO + "," + Name);
    }
}
```

Object :- An object is a real world entity created from class.

it represent

- State → Value of Variable
- Behaviours → Methods or actions

Ex: `Student S1 = new Student ();`

Variable Declaration :-

→ Declare three int a, b, c

→ a=10, b=10 // Example of intilization

Variable is a name given to a memory location where data is stored.

Variable declaration means telling the compiler what type of data the variable will store.

Intilization :-
Means giving a value to the variable when it is declared

data Type Variable Name = Value;

int age = 20;

Types of Variable

local
1) Declared inside a Method.
2) Must be intilize before use
3) No default value
int num = 10;

instance
inside a class
→ Each Object gets its own copy
default Value
int rollNo;

Static
- Declared Static keyword
- shared by all Objects the class
- Only one copy
Static String Cmp

Operatores in Java :-

Java Operatores are special symbol that perform Operations on variables & Values.

Type of Operation

(1) Arthmetic :- Used for Mathmatical Calculation
(+) Addition
(-) subtract
(*) Multi
(++) inc (--) dec (%) Modulo

(2) Assignment :-
used to assign values to variable
(=) , (+=) (-=) (%=)

(3) Relational
used to combine or modify boolean operation
(&&) (||) (!) (NOT)

(4) Bitwise :- used to Perform Operation on individual bits or integer types
& , | , ^ , ~ , <<, >> , >>>

(5) Ternary :- A shorthand for an-if-then-else Statment
condition ? expression 1 : Expression 2

Type Casting :-

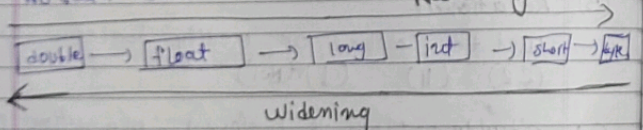
Type casting in Java means converting one data type into another.

There are two type of Casting.

① Widening

When a smaller data type is automatically converted into a larger data type.

No data loss occurs




```
Public class WidenExample
Public static void main (String [] args)
{
    int num = 10;
    double d = num
System.out.Println(d);
```

② Narrowing :- (Explicit)

When a larger data type converted into a smaller data type.

Data loss is possible

Command line Argument in Java

Command line arguments are values passed to a Java Program when the Program start.

These argument are recived in the main() Method through the array.

Why use :

→ To provide input without using scanner or Option Pane

→ To Make the program more flexible

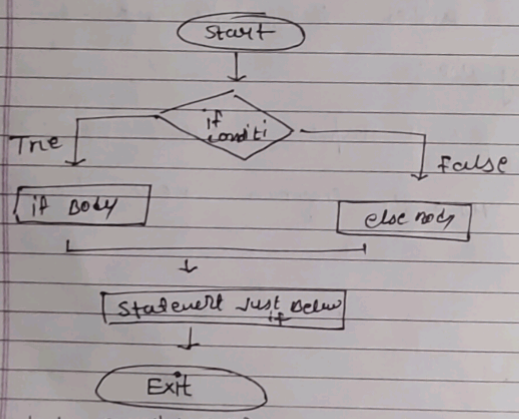
2nd UNIT

1. Declarations Control Structure Arrays :-

if - else Statement :-

The if-else statement is a decision making statement that checks a condition.
if the condition is true, the if block execute, otherwise the else block execute.

Flow chart




```
int marks = 60
if ( marks >= 40) {
System.out.println ("Pass");
} else {
    System.out.println ("fail");
}
```

Switch Statement :-

The switch statement is used when we have multiple choices. It compares a variable with different cases.

```
int day = 2;
switch (day) {
  case 1:
System.out.println ("Monday") : break;
  case 2;
System.out.println ("Tuesday") ; break;
  default ;
System.out.println ("Invalid");
}
```

LOOPS

A loop is a control structure used to repeat a block of code multiple times until a specific condition become false.

Types of loop

1) while
2) do while
3) for

① for :- The for loop is used when the number of iterations is known.

Example for (initialization ; condition ; increment) ;
{
}

```
for (int i = 1; i <= 5; i++){
  System.out.println(i);
}
```

② while :- The while loop checks the condition first & if it is true the block of code runs

```
int i = 1;
while( i<=5)
System.out.println(i);
  i++;
}
```

③ Do while :- The do while loop runs the code at least once.

```
do {
  // code
} while (condition);
```

```
int i = 1;
do {
System.out.println(i);
i++
} while (i<=5);
```

Explain Break & Continue :-

1) Break :- It is used to stop or exit a loop or switch statement immediately.

- To stop a loop when a certain condition is met.
- To exit switch cases

```
for (int i = 1; i <= 5; i++){
  if (i == 3){
    break;
  } System.out.println(i);
}
```

2) Continue :- used to skip the current situation iteration of the loop and move to the next iteration.

- To skip unwanted values
- To continue the loop without executing remaining code of the iteration

```
For (int i = 1; i <= 5; i++) {
  if (i == 3) {
    continue
  } System.out.println(i);
```

Enums :-

Enum (Enumeration) is a special data type that is used to define a fixed set of constants.

Example → days of week, directions, Months are value that never change

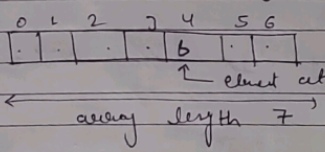
Use ?

1. Improve readability
2. Ensure Type - Safety
3. Avoid errors.

Example:
```
Enum Day {
    Mon, Tue, Wed, Thu, Fri, Sat, Sun
}
Public Class TestEnum {
Public Static Void main (String [] args)
    Day d = Day.Monday;
    System.out.println(d);
}
}
```

-: Arrays :-

An Array is a collection of similar data type element stored in contiguous memory location



1. One Dimensional
This is the simplest form it stores data in a single row.
```
int [] num = {10,20,30,40};
```

2. Two Dimensional
Also called Matrix or Table array. Data is arranged in rows & columns
```
int [][] matrix = {
{10,20,30}
{40,50,60}
```

3. Multi Dimensional → An Array containing more than two dimension.
Used for complex data
```
int [] [] [] arr = {
    {{1,2}
    {3,4}}
    {{5,6}
    {7,8}}
};
```

Enums :-

Enum (Enumeration) is a Special data type that is used to define a fixed set of constants.

Example → days of week, directions, Months are value that never change

Use ?

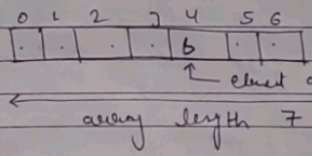
1. Improve readability
2. Ensure Type - Safety
3. Avoid errors.

Example: Enum Day {
Mon, Tue, Wed, Thu, Fri, Sat, Sun
}

```
Public class TestEnum {
Public static void main (String [] args)
    Day d = Day.Monday;
    System.out.println(d);
}
}
```

-: Arrays :-

An Array is a collection of similar data type element stored in contiguous memory location



1. one Dimensional
This is the simplest form it stores data in a single row.
int [] num = {10,20,30,40};

2. Two Dimensional
Also called Matrix or Table array. Data is arranged in rows & colums
int [][] matrix = {
{10,20,30}
{40,50,60}

3. Multi Dimensional → An Array containg more than two dimension.
Used for complex data
int [] [] [] arr = {
{{1,2}
{3,4}} ,
{{5,6}
{7,8}} };

Anonymous Array :-

is an array that has no name. It is created for one time use, mostly to pass values directly to a Method.

* After use it gets destroy automatically

new int [] {10,20,30};

features

- No Name
- used only once
- Save Memory
- Automatically removed by JVM

Difference b/w String & String Buffer

String	String Buffer
① String is immutable	① Mutable
② consume more memory	consume less memory
③ Thread - Safe	Thread Safe (Synchronized Not Method)
④ with the String class you cannot create modifiable string	The string Buffer class is used to create modifiable string
⑤ represent fixed length	doesnt represent fixed length
⑥ It is slow	It is faster
⑦ fixed unchanging strings	Single - threaded string Manipulation.

String :- sequence of character.

Method

length()
toUppercase()
equals()
compareTo()
concat()
trim()
replace()
split()

UNIT-3

Class, Interface & Access Control

Static Variable :-

A Static Variable is a Variable that belongs to the class not the object.

It is created only once in memory and shared by all object.

Static Method :-

A Static Method belong to the class not object
you can call it without creating an object.

Constructor

Constructor is the special Method which is automatically called when an object is created.

It has the same name as the class & does not have any return type

features :-

- Same name as class
- No retur type
- automatically called
- Cannot be static, final or abstract
- used to intilize Variable.

Type of Constructor :-

1) Default :-
A constructor created automatically by Java if we don't define any constructor.

```
Class Student {
    Student () {   // default constructor
System . out . Println (" Default Constru Called
Public Static Void Main (String [] args) {
        Student s = new Student ();
// Calls Constructor custom
```

2) Parameterized :- constructor that has parameters (values).

```
class Student {
    int id;
    String name;
    Student ( int i, String n)  // Parameterized
        id = i;
        name = n;
    public static void main( String [] args){
        Student S= new Student (101, "Rahul");
}
```

—: Constructor overloading :-

When a class has multiple constructor with different parameters. It is called constructor overloading

```
class Demo {
    Demo (){
// Constructor 1
    System.out.println ("Default"); }
    Demo (int x){
// Constructor 2
    System.out.println("value : "+x);
}
```

Difference b/w Constructor & Method

Constructor	Method
1) constructor name must be same as the name of the class.	Method's name can be anything.
2) doesn't have any return type.	Must have a return type.
3) invoked by the system implicity	invoked by the Program They are invoked explicity
4) Can be used to intilize an object	4) Method consist of Java code to be execute.
5) They are not inherited by subclass	They are inherit by subclass.

Method :- A method is a block of code that performing a specific task.

⇒ Single function or Method can perform ~~different behavi~~ depending of the object & parameters.

Difference b/w Method overloading & Method Overriding

Method Overloading	Method Overriding
Compile time Polymorphism	Run-time Polymorphism
① It occurs with in the same class.	① It occurs between two classes Superclass & Subclass
② inheritance is not involved.	② inheritance is involved
③ Parameters must be different	③ Parameter must be same.
④ Return type may or may not be same	④ return type must be same
⑤ Static binding	⑤ dynamic binding
⑥ Static Polymorphism	⑥ Dynamic Polymorphism
⑦ No special keyword use	⑦ virtual & override keyword use

-: This keyword :-

This refers to the current object of the class.

Use :-

① To refer current Object's instance variable.

② To call current class Method.
→ this.method name()

③ To call current class constructor
→ this()

④ To return current object
→ return this;

```
class student {
    int id;
  Student (int id) {
    this.id = id;
  }
}
```

Super

super refers to the parent class Object (base class)

Use ? ① To access Parent class instance variable

② To call parent class Method
→ super.methodname

③ To call Parent class Constructor

```
class A {
  int x = 10;
}
class B extends A
  int x = 20
  void show() {
    System.out.println
    (super.x);
```

-: Inheritance :-

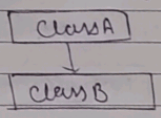
* Inheritance is a fundamental concept of OOP. ~~that allow to aquire~~ &

* Inheritance is a mechanism in Java in which one class (subclass) aquire the properties & methods of another class (superclass).

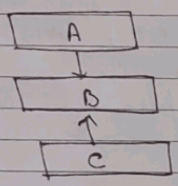
Example:
```
class Animal {
    void eat () {
    System.out.println ("Eating ... ");
}
Class Dog extends Animal
    void bark () {
    System.out.println ("Barking")

Class Test {
Public static void main (String [] args)
    Dog d = new dog ();
    d.eat ();
    d.bark ();
```

Type of Inheritance

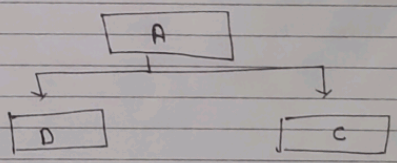
- single :-
one subclass (Parent) inherit the properties & Method of one superclass (child)



- Multilevel :-
A class is derived from another class derived class, forming a chain of inheritance



③ Hierarchical :- one superclass is inherit by multiple subclasses.



- One Parent many children.

Abstraction

Abstraction is the process of hiding unnecessary details & showing only essential features of an Object.

```
Abstract class Animal {
    abstract void sound(); // hidden implementation
}
    class Dog extends Animal {
        void sound() {
        system.out.println("Barks");
    }
```

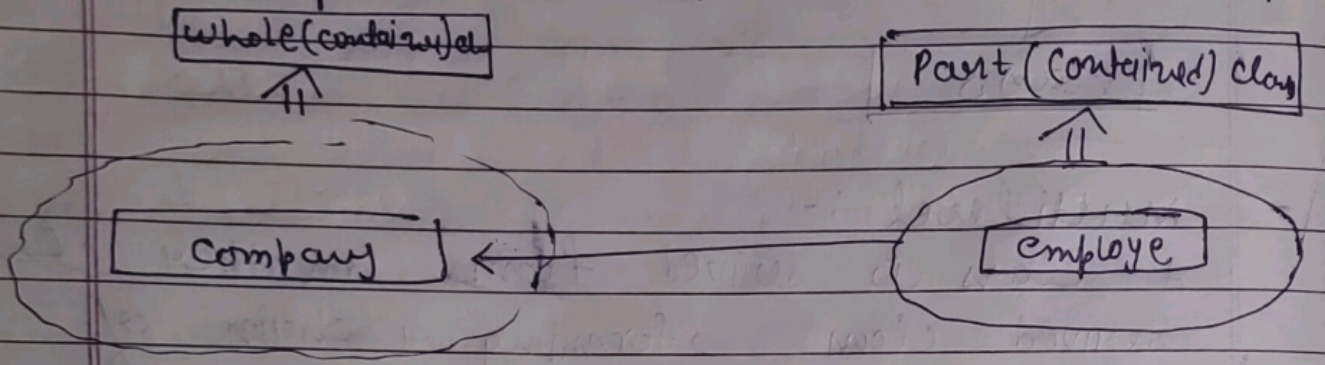
Difference b/w Abtract class & interface :-

Abstract	Interface
1) Abstract class can have ~~use~~ abstract & ~~Public Modifiers~~ non abstract Method	1) Interface can have only abstract Method.
2) Doesn't support Multiple Inheritance	Multiple 2) Support ^ Inheritance
3) Contains Constructor	3) Doesn't contain constructor
4) Fast	Slow
5) We can extend only one abstract class	We can implement multiple interface
6) abstract keyword used to declare abstract class	interface keyword

Aggregation

is a Type of relationship where one class HAS another class. but both can exist independently.

It represents a weak HAS-A realtionship



- One Object Contains another Object
- But the Contained Object does NOT depend on the container Object
- If the container Object is destroyed other can still alive

-: Access Modifiers :-

are keyword in OOP that control the visibility & accessibility of classes, methods, variable & constructor.

• 4 Type of access Modifiers

① Public :- A Public member can be accessed from anywhere same class, same Package or a diffen Package

```
Public class A {
Public void show () {
System.out.println ("Public Method");
```

② Private :- A Private Member can be accessed only within the same class.
No other class can use it.

```
class A {
private int data = 10;
Private void display () {
System.out.println ("Private Method")
```

③ Protected :-
A Protected Member can be accessed in,
- the same class
- the same Package
- & Subclass in a different Package

```
Class A {
  protected int num = 20;
}
    Class B extends A {
      void show () {
    System.out.println (num);
```

④ Default :-
If no modifier is written, it is called default access.
Default Package Members can be accessed only within the same package.

```
class A {
void show () {
System.out.println ("Default Meth
}
}
```

Package

A package is a folder that is used to organize classes, interfaces & sub-packages.

It helps keep the code clean, grouped & easy to Manage.

Type

1- Built-in ⇒ already provided by java
   - java.util
   - java.io
   - java.lang

2- User defind ⇒ Created by the Programmer)

Example :-

```
Package mypack;
Public Class A {
   Public Void show () {
System.out.Println ("Hell from packge A")
}
}
```

Difference b/w Call by Value & Reference

Call by Value	Call by Reference
1⇒ Calling function send copies of data.	1) calling function send adress of data
2⇒ original data remains unchanged	2) original data can be Modified
3⇒ Usage for immutable data or small data	3) for Mutable data or large data
4⇒ A New memory location is created	4) No new Memory location is created
5⇒ Safe (original Data is Protected)	5) Risky (original Data may be change
6⇒ used for Primitive data types in Java	used when passing Object in Java

UNIT - 4

⇐ Applet & AWT Control ⇒

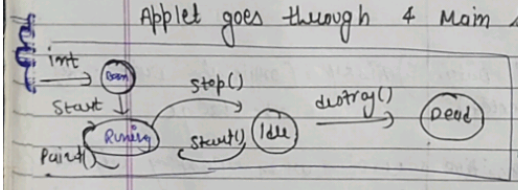
What is Applet ?

An Applet is a small Java Program that runs inside a web browser or applet viewer.

It cannot run on its like normal Java Application.

- Applet is the Part of Java GUI

Applet goes through 4 main Stages



Applet life cycle

(1) init () ⇒ Called once when the applet is created

(2) Start() ⇒ Called every time the Applet becomes acti

(3) Paint () ⇒ used to draw shapes, images on the Applet screen

(4) Stop () ⇒ called the user leves the Applet Page

(5) destroy() ⇒ called when the Applet is removed from memory.

Difference b/w Java Application & Applet

	Application	Applet
(1)	Contains Main Method	1) does not contain main Method.
2)	Can be run without a browser	2) Requires a browser
3)	Entry Point is Main Method.	3) Entry Point is init Method
4)	Generally used for Console Programs	Generally used for GUI interface
5)	More Secure	Less Secure
6)	Can run independently	6) cannot run independently

Applet life Cycle :-

The Applet life cycle represent the different stages through which an Applet passes from Start to end.

Purpose

(1) init ⇒ • initilize variable, load image, font, Allocate resou

(2) Start ⇒ Start animations, Start threads, Resume activi

(3) Paint ⇒ Display output on screen

(4) Stop ⇒ Pause animations, Stop running threads

(5) destroy ⇒ cleanup work, Free memory & resources

Explain the applet tag ?

The <applet> tag is an HTML tag used to display & run a Java Applet inside a web page

* This tag acts as a bridge between HTML and Java Applet

Syntax

<applet Code = " Applet File. Class "
width = " width " height = " height ">
</applet>

<Param tag>

The <param> tag is an HTML tag used inside the <applet> tag to send values

⇒ used to pass parameters (data / value) to a Java Applet from an HTML Page

Syntax

<param name = " parameter Name "
Value = " parameter Value >

AWT (Abstract Window Toolkit)

Awt is a set of Java classes used to create GUI (Graphical User interface) Application.

⇒ Using AWT we can create windows, buttons, textfiels & Visual component

Used :⇒

1) To create GUI Screen
2) To intract with the user.
3) To build window base Progra
4) To handle events

components :-

1) Label
2) Button
3) Text field
4) Text Area
5) checkbox
6) Scroll bar
7) Pannel
8) Menu

UNIT - 5

Exception handling, Multi threads I/O

Exception

* Exception is an unexpected or abnormal event that occurs during program execution & stop the normal flow of the program.

Example
```
int a = 10;
int b = 0;
System.out.println(a/b);
```

Type
1. Checked → at compile time
2. unchecked → occur during runtime
3. error → Cannot handle by program

-: Exception handling :-

Exception handling is a mechanism to detect and handle runtime errors so that the program does not crash.

* When an unexpected event occurs, Java transfers control to the exception-handling code.

Advantage:-
- → Prevent program crash
- → Seprate errors handling code
- → Shows meaningful errors msg to the user
- → Improves program reliability & stability.

Eh keyword

try → Block of code where exception may occur

catch → Handles the exception

finally → Execute always whether exception occurs or not.

throw → used to throw an exception manually.

throws → Declares exception in method signature

~~try - catch - finally~~

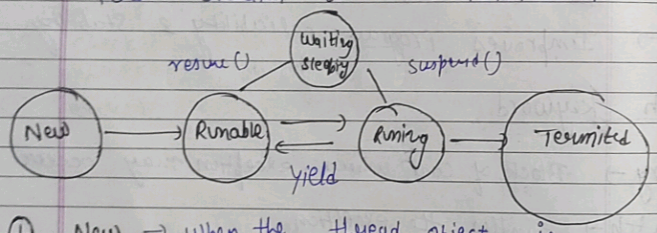
Thread

A Thread in Java is the smallest unit of a Process that can run independently.

It is used to Perform multiple task at the same time. (multithreading)

Life Cycle of The Thread

Thread LC represents the different states a thread goes through from creation to termination.



① New → when the thread object is created using thread class.

② Runable → when we call start() method the thread moves to runable state.

③ Runing → when the thread gets CPU time, it enters the runing state.

④ Blocked / waiting

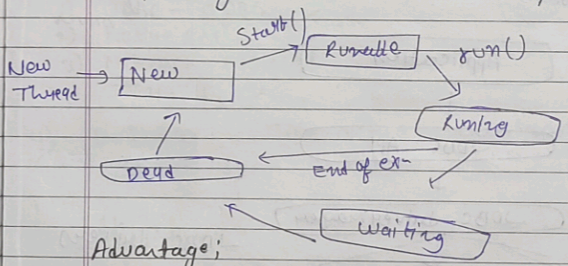
A running thread can temporarily stop & move to waiting states.

- sleep()
- wait()
- blocked()

⑤ Dead ⇒ After the run() method finishes the thread enters the Dead state.

Multithreading

Multithreading is the process of executing multiple threads simultaneously within single program. Each thread runs independently & Perform a seprate task.



Advantage;

1) fast execution
2) Maximum CPU usage
3) Multiple task run simultaneously
4) Better Performance in GUI

JDBC

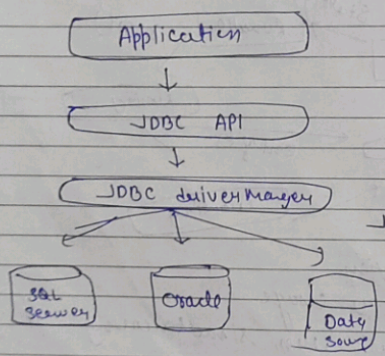
Java Database Connectivity

It is an API (Application Programming Interface) in Java that allows Java Program to connect to a database send SQL queries & retrieve results.

JDBC architecture is based on two tier architecture involving

1) Java Application (Client)
2) JDBC Driver Manager
3) JDBC drivers
4) Database

1) Client ↓ directly
2) Database
~~& connect through JDBC driver~~



1) JDBC API :-

- It provides classes & interface (Connection, Statement, Result Set)
- Help Java Program intract with the database.

2) JDBC Driver Manager

a) Control all ~~database~~ drivers
- Loads driver using

3) JDBC Drivers

Drivers act as a bridge b/w Java Progm & database 4type

1) JDBC - ODBC
2) Native API
3) Network protocol
4) Thin - Driver

4) Database - The actual database (mySQL) that stores data.

Advantage

1) Platform independent
2) Works with all relational database
3) Easy to use API
4) Support SQL command
5) Provid secure & fast connectivity.

Garbage Collection in Java

Garbage collection is the memory management of Java.

Garbage collection is the process by which the JVM automatically removes unused object from memory to free up ~~memory~~ space.

works :-

1) JVM continously monitors heap memory
2) It find object that are not being used
3) These object mark as garbage
4) JVM Garbage collection remove them.
5) Memory is freed for new object

⇒ Before destroying the object garbage collector may call the finalize() method.

```
class Test {
  Public Static Void Main (String [] args) {
// Object created
   Test Obj1 = new Test();
   Obj1 = null
   System.gc();
System.out.println("End of the Prgm")

Protected Void finalize() {
  Sn.o.p ("Object destroy by garbage Collector")
```

1) write a Program to Prime No ro not?

```
import java.util.Scanner;
Public class PrimeCheck{
 Public Static Void Meim (String [] args){
    int n = 17;
    Scanner sc = new
   Scanner (System.in)

System.out.Println ("Enter a number");

      int n = sc.nextInt();
      boolean Prime = true;
       if (n <=1)  Prime = false
    for(int i=2;   i<=n/2; i++)

      if (n%i ==0)
  break Prime = false;
   Syste.ou
```

---

*

```
Public class Prime {
Public Static Void Mai (String [] args);

      int n= 17;

      boolen Prime = true;
    if (n<=1) Prime=  false
for (int i=2; i<=n/2 ; i++)
  if ( n%i ==0) {
   Prime = false;
break
Syste out ( Prime? "Prime"; Not Prime)
```

② Even or Odd

```
import Java. Util. scanner
  Public class Evenodd {
Public Static Void main( String [] args) {
    scanner sc = new
    scanner (System. in) ;
System out. Println ( "Enter the number);
      int n =   sc. nextInt ();
    if (n%2 == 0) {
System. out. Println (" n + " is Even");
else {
System. out. Println ( " n + " is odd ");
}
}
```

③ Fibonacci series :-

```
Public class Fibonacci {
Public Static Void Main (String [] args) {
  int n = 10;    // how many num to Print
    int a=0 , b=1 ;
System. out. Print ( a + " " + b + " ");
for (int i= 2; i<n ; i++) {
    int c = a +b
System. out. Print (c+ " ");
    a=b;
    b=c;
}
```

I/O Stream

File input Stream

File input stream is used to read data from a file in the form of bytes

- Read data byte by byte
- used for reading binary file

File Output

used to write data into a file in the form of bytes

- write data byte by byte
- used for write binary & text data
- Can create file it it does not exit

-: Paper Solve :-

SECTION "C"

20) Answer No 20

Difference Btw Method overloading & Method Overriding

Method Overloading	Method Overriding
1) It is also known as compile time Polymorphism	1) Method overriding also known as Run time Polymorphism
2) ~~Using~~ It is faster	It is slower
3)	

21 Answer No 21

Package

Package is the folder that can store is used to organize the classes, ~~Methods~~ & Subpackages. interface

These are Mainly 2 types
Pre Built in
1) ~~User~~ defind → Those Packages are already provided by java developers.
called ~~java~~ userdefind Package like
→ and come with JDK
1) Java.net → Networking classes
2) Java. ~~Scanner~~ Util → Contain utility class
3) Java. I/O → I/O classes
4) java.sql → Database Programing classes created
2) user defin ⇒ Those Packages are ~~develop~~ by the user called user define Packages
↳ to orgnize their own classes
1) ~~file save~~ Create a folder
2) Compile ~~java~~ Code the file
3) ~~Class creation~~ create a java file

24) Answer No - 24

—; Exception handling :-

Exception is ~~the process when the~~ an Unexpected & abnormal event occur during the execution. It is change the flow of Program. To save error and Program crashing uses Exception handling.

Type 1) Checked
2) Unchecked
3) Error

Try - catch finall

Try - catch - finally is the keyword to used Exception handling.

Try :- Try → contains risky code using for runnable the program

catch :- catch block used for perovide clear output
↳ hondl the exception

finally :- ~~Finally~~ is used for the end of the Program without any changes.
↳ always runs used for clenup

Example :-

```
Public Class Exception {
Public static void main (String [] args);
        int n = 10;
        if (int) (10/0)
        int n=
        try { int a = 10/0
        System.out.println (" n " , Print the)
        Catch (Arthmetic Exception) {
System.out.println (" n ", "Error" cant divided);
}
System.out.println (" progm continuous);
```

25) Answer No- 25

Number is Prime or not.

```
import java.util.Scanner
Public Class Primecheck {
    Public Static Void Main( String [] args);
        Scanner sc = new
        Scanner (System.in)
        System.Println("Enter the number");
        int n = sc.nextInt();
        Boolean Prime = True
        if (n<1) : Prime = false
        for (int i=2 , i<= n%2 , i++)
            if ( n%i == 0);
            Prime = False
            break;
        System.out.println( " Prime? "
                "Prime": "Not Prime")
```

26) Answer 26

Constructor

Constructor are the special method it is called automatically when object is created.

Types of constructor

1) Default
2) Parametized

A Constructor created automatically if we don't define any constructor.

1) Default :- When the Constructor have not any Parameter java provide automatically a Constructor called Default Constructor

2) Parametized :- when the Constructor have any value called Parametized constructor. That accepts value when object is created

Event handling

Event handling means when the occur any event called event handling like

1) Performing a task
2) key board recalled operation Perform
3)

SECTION (A)

1) Java Virtual Machine
2) OOP
3) Java development kit
4) Extends
5) ~~Finalize ()~~
6) Similar
7) Application Programming Interface
8) start ()
9) Java Database Conectivity
10)
11)

SECTION 'B'

12) Answer No 13

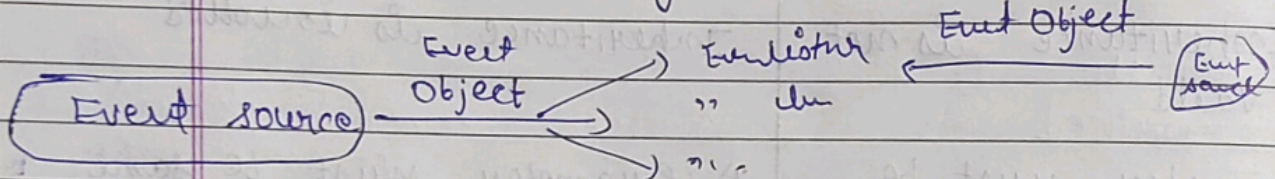
Difference blw Method overloading & Method overriding

Method overloading	Overriding
1) Represent compile time Polymorphism	1) Represent Run time Polymorphism
2) It occurs with the same class	2) It occurs between two class subclass & Superclass
3) Inheritance is not valid	Inheritance is involved
4) Parameter must be different	4) Parameter must be same
5) Static binding	5) Dynamic binding
6) Perform similar task in different ways	Task redefining a parent class method in the child class
7) No special keywords use	7) virtual & override key

Event Handling

Event handling is the mechanism that allows a program to detect user action such as clicking a button, typing on the keyboard & respond to those actions.

These user actions are called Event. Java handles these events using event handling. Delegation Model.



1) Event Source
   - The object that generate the event

2) Event
   The Actual user Action
   click
   press

3) Event listener ⇒ Method to handle event
   - Action " → handle button click
   - Mouse " → " Mouse Action
   - Key " → " keyboard

Methods contain handle events
A Java interface that contains