

## Module - I

Session 1 Introduction

- Q Provide a brief overview of Java history, its creators, and its evolution.
- Java is an object-oriented, class-based general purpose, robust, secure, safe, programming language.
  - It was developed by Sun microSystem in 1993 by James Gosling. Its name was Oak. In 1995 it's name has been changed to Java because of some legal issues.
  - Java was designed for development of software for Electronic devices like TVs, VCRs, etc.
  - Java removes limitation like Portability and reliability.
  - Java is strongly typed.
  - In Java, compilation time contains translating programs into machine intermediate byte code which is called byte code.

⇒ Runtime activities like loading & linking of classes are needed to execute a program.

⇒ Most important feature of Java is platform independence. Java is not tied to any particular hardware or OS. Java programs follow achieve "Write once, Run anywhere". It can be executed anywhere on any system.

### Q2) Explain Java's features.

⇒ There are various features described by Sun Microsystems. Some of them are given below:

[1] Compiled & interpreted

[2] Platform independent

[3] Portable

[4] Object oriented

[5] Robust & Secure

[6] Distributed

[7] familiar, Simple & small.

- 18) Multithreaded & interactive
- 19) High Performance
- 20) Dynamic & Extensible

## 11) Compiled and interpreted :-

→ The program can be either compiled or interpreted, but in other languages but in java it combines both, so it is called a static system.

→ First java compiler translates source code into byte code.

→ Java interpreter generates machine code that can be directly executed by machine that is by running a program.

Source code

Byte code

Java

Machine code

Platform independent :-

→ Program written by user is called source code & executable program is called object code.

Value

→ Those objects can be executed in the particular CPU, so it can't be run on different platforms but Java removes this limitation.

→ In Java Compiler converts the code into byte code which can be executed in any system using JVM or interpreter. This Java is platform independent.

### [3] Portable :-

→ Java important feature is portability.

→ Java program can be easily removed from one computer system to another system anytime & anywhere. Changes in OS or System will not force any changes in Java program.

⇒ Java provides "Write once run anywhere" (WORA) which makes Java language portable with only one condition that the system must have interpreter & Java runtime environment.

## IV) object oriented :-

- ⇒ Every thing in Java is based on object.
- ⇒ Java comes with Extensive set of classes, which is arranged in Packages that we can use in our program by inheritance.
- ⇒ Java is a pure object-oriented language because object is the fundamental structure in Java.
- ⇒ Here even we can convert the data types into object by using wrapped class.

## IS) Robust & Secure :-

- ⇒ Java provides Protection to ensure reliable code.
- ⇒ Java has strict runtime & compile time checking for memory leaks.
- ⇒ Java also has concept of Exception handling which captures series of errors & eliminate the risk of system crashing so, Java is robust.

## 16] Distributed :-

- ⇒ Java is designed as a distributed language for creating applications on networks.
- ⇒ Java application can be open & across the remote objects location in internet. These multiple programmers at multiple remote location can collaborate & work together on a single project.

## 17] High Performance :-

- ⇒ High Performance is done by intermediate byte code & the multithreading enhances the overall execution speed of Java program.

## 18] Simple, Small & Familiar :-

- ⇒ Java is simple & small & simple.
- ⇒ It doesn't support pointers, Pre-processor header files, goto statement & others.
- ⇒ It eliminates class inheritance & overloading, multiple inheritance, thus Java is simplified version of C++.

Q3) Write down advantages of Java.

→ The following are the advantages of Java :-

1) Clear Syntax :-

→ Java inherit Syntax from C/C++ but Simplifies it by removing confusing Elements like Operator and overloading & Complex Pointer.

→ This make it beginner-friendly.

2) Easy Programming :-

→ Java Eliminates manual memory management & use automatic Garbage Collection.

→ This reduce the chances of Error & makes Java Programming Easier & more Robust.

3) Object-oriented :-

→ Java follow object oriented approach, allowing developers to think objects & design programming in terms of real-world objects.

→ Concept like inheritance, Polymorphism & Encapsulation make static code modular & reusable.

4) Clean Code :-

→ Java Promotes simplicity & readability in code, helping programmers create clean & smooth & error-free programs.

**15] Portable :-**

- ⇒ Java Programs are Compiled into byte code, which can be Executed on any System with a JVM, making it highly Portable.

**16] Platform independent :-**

- ⇒ Once compiled, its code can run without modification on different Operating Systems, such as windows, macOS, or Linux.

**17] Multi-Threading :-**

- ⇒ Java allows multiple threads of execution to run simultaneously, enabling efficient multitasking & better utilization of system resources.

**18] Secure :-**

- ⇒ Java performs runtime checks & ensures that the code is safe from errors or malicious attacks before execution.
- ⇒ It also restricts access to critical resources, enhancing security.

**19] Scalable Systems :-**

- ⇒ object-oriented design ensures that Java applications can scale smoothly from small projects to large enterprise systems.

4 what is byte code how does the JVM execute byte code explain the importance of JVM in achieving platform independence.

#### \* Byte Code :-

- ⇒ Bytecode is the intermediate code generated by Java Compiler (javac) when a .java file is compiled.
- ⇒ It is stored in a .class file.
- ⇒ Bytecode is Platform-independent low-level representation of the Java Program.
- ⇒ It consists of instructions that the Java Virtual Machine (JVM) can interpret and execute.

#### \* How Does JVM Execute bytecode

##### 1) Loading byte code :-

- ⇒ The compiled bytecode is loaded by the JVM which is from the .class file into memory.

##### 2) Verification :-

- ⇒ JVM verifies bytecode to ensure that it is safe and secure. It follows Java Security rules and doesn't violate runtime constraints.

##### 3) Execution :-

### 13) Execution :-

⇒ The JVM interpreter used to translate bytecode into native machine code for line by line.

### 14) Runtime Services :-

⇒ JVM manage runtime task such as memory allocation, garbage collection, & thread management during execution.

Java Program → compiler → virtual machine

Source code → compiled into Byte Code

Figure :- Process of compilation

Virtual machine → Java interpreter → Real machine  
Byte code → converted into Machine code

Figure : Process of Converting Bytecode into machine Code

\* Importance of JVM in achieving platform independence

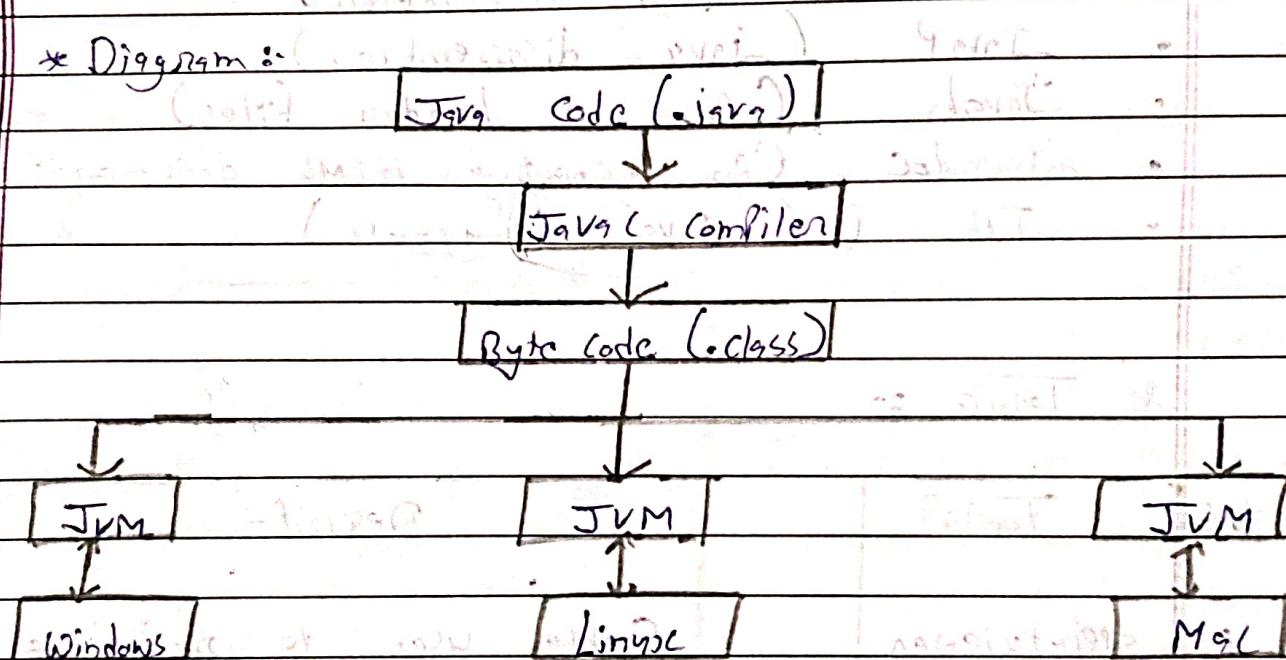
### 11) Bytecode as universal format :-

⇒ Java compile the program & generate .class file containing bytecode, which is

Platform-independent & not tied to any specific os

- 2) Platform-specific Executions :-
- JVM translate byte code as per sp into specific operating system machine code.

- 3) Write once, Run anywhere (WORA) :-
- Developers write & compile single program once, & the same byte code can run on any system with any JVM, eliminating the need for instantiations.



[5]

## Explain JDK with Diagram.

\* JDK (Java Development Toolkit) :-

→ It is a collection of tools that are used for developing & running Java Programs.

→ It includes following things :-

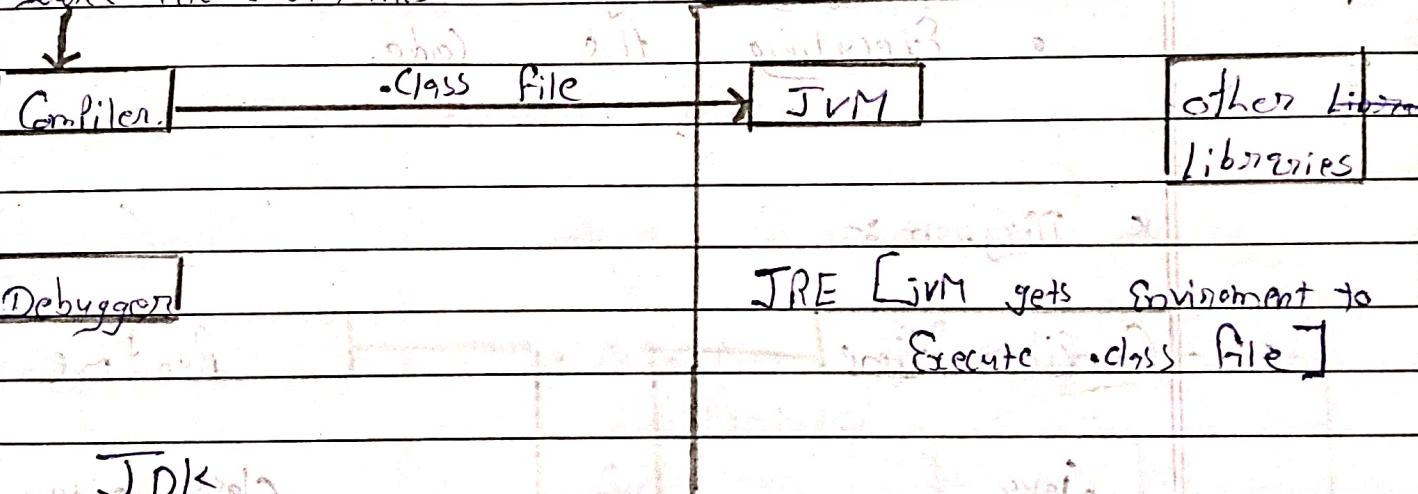
- applet viewer (for viewing Java applets)
- javac (Java Compiler)
- java (Java interpreter)
- Javap (Java disassembler)
- Javah (For header files)
- Javadoc (For creating HTML documents)
- Jdb (Java debugger)

\* Table :-

No	Tools	Description
1	appletviewer	Enable user to run Java applets
2	Javac	Translate Java Source code into bytecode that interpreter understand
3	Java	It runs application by reading & interpreting byte code.

4.	javadoc	Enable the user to convert the bytecode files into a program description.
5.	Javadoc	Create HTML Format documentation from given source code file.
6.	Javash	Produce header files to use it with native methods.
7.	Jdb	Analyze. It helps users to find errors in our program.

\* Diagram to explain the method Source file (.java files) converted to binary



JDK

Plain old stuff

language standard

platform 9 bit and

JDBC

16

Explain Java runtime Environment  
also Java Standard Libraries.

### \* JRE (Java runtime Environment)

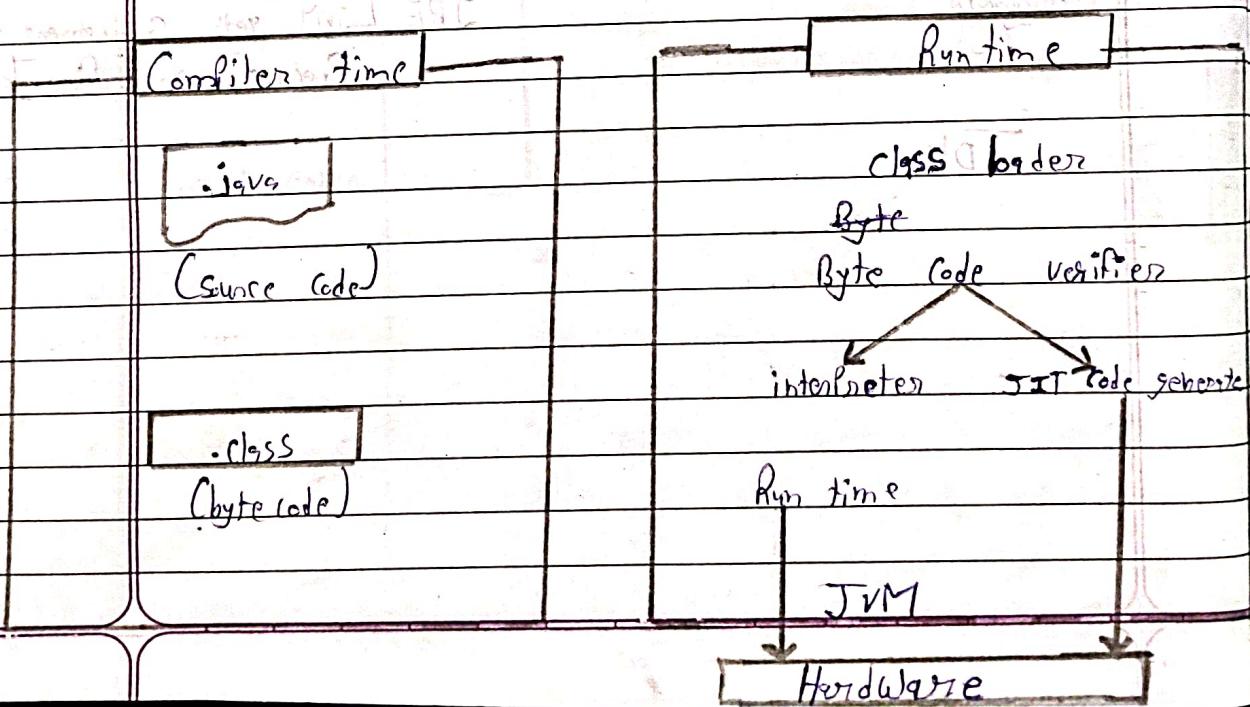
⇒ JRE consists of ~~JVM~~ JVM and the Program on the other.

⇒ JRE is Part of Java Platform that Provides the necessary tools & runtime Environment to Execute Java application.

⇒ It includes :-

- Loading the class files
- Verifying byte code
- Executing the code.

### \* Diagram :-



## \* JSL (Java Standard Library)

→ It includes hundreds of classes & methods which is grouped into Several functional Packages.

→ Most commonly used Packages are -

- lang (language support Package)
- util (utility Package)
- io (input/output Package)
- net (Networking Package)
- AWT (Abstract window toolkit Package)
- applet (applet Package)

No	Package name	Description
1	Lang	<ul style="list-style-type: none"> <li>• It includes classes &amp; methods which required for implementing basic features of Java.</li> </ul>
2	util	<ul style="list-style-type: none"> <li>• collection of classes &amp; methods to provides utility functions such as date &amp; time function</li> </ul>
3	io	<ul style="list-style-type: none"> <li>• A collection of classes needed for input/output manipulation.</li> </ul>
4	AWT	<ul style="list-style-type: none"> <li>• Collection of classes that implements platform independent graphics user interface</li> </ul>

PS net

• Collection of classes for communicating with other computers via internet

• applet

• Allows us to create applets

17

## Difference between Procedure-oriented Programming (PoP) and object-oriented Programming (oop)

No.	Point	PoP	oop
1	Meaning	Program is divided into small parts called functions.	Program is divided into parts called object.
2	Flow	PoP follows Top Down approach.	oop follows Bottom up approach.
3	Access	PoP does not have any access specifier	oop has access specifier named Public, Private, Protected
4	New data	Adding new data and function is not easy	Adding new data & function is easy.
5	Data hiding	PoP does not have proper way of data hiding	oop has data hiding features
6	Used	It is used to design medium-sized program	It is used to design large & complex program
7	Inheritance	There is no concept of inheritance & data hiding.	There is a concept of inheritance & data hiding.
8	overloading	overloading is not possible	overloading is possible
9	Example	C, VB & Pascal	C++, Java, Python

[8]

Explain structure of simple Java

Program & discuss the purpose  
of main method and Explain  
the meaning of Public, static  
void main (String args)

### \* Basic Structure of Java Program

→ A Java Program is divided into  
different Sections, some of which  
are optional, while others are  
essential.

→ Below B

→ Below

→ Below is an explanation

### \* Diagram:-

Documentation Section	Suggested
Package Statement	optional
Import Statement	optional
Interface Statement	optional
Class definitions	optional
main method class { ... }	Essential

Explanation & Examples :-

### 1) Documentation Section :-

- Provide meta data about Program, such as name, purpose, & program details.
- Types :-

#### II Single-line Comment :-

Eg :- // This is single-line comment

#### III Multi-line Comment :-

Eg:- /\* This is multi-line comment  
use to describe the code

#### IV Document Comment:-

Eg:- /\* This is document comment

\*/

### 2) Package statement :-

- Various classes are grouped into Single package for better organization.

• Syntax of Package Statement :-

Eg :- `com.PackageName.Student;`

### 13] Import statement :-

- Allows to use the classes & methods from other packages.
- Syntax :- ~~import~~ import Packagename. Classname;
- Eg :- import java.util.Scanner;

### 14] Interface statement :-

- Used to declare interfaces, which ~~contains~~ contain methods declaration but no implementations.
- Syntax :- interface InterfaceName {
   
    void method();
   
    void method();
 }
- Eg:- interface Car {
   
    void start();
   
    void stop();
 }

### 15] Class definitions:-

- Define blue print of objects, which contains members & methods.
- Syntax :- class Class Name {
   
    int variable;
   
    void method();
 }

Eg :- class Student {

    string name;

    int id;

    double per;

}

### Q) Main method - class :-

- This main class is act as Entry Point for the any Java Program.

- It is essential

Syntax :-

Public ~~class~~ class mainclass {

    Public static void main (String [] args) {

        System.out.println ("Hello Java");

}

Eg :- Public class student {

    Public static void main (String [] args) {

        System.out.println ("Hello Java");

}

3

### \* Purpose of the main method :-

⇒ The main method is the entry point of any Java application.

⇒ The JVM start executing the program from main method.

→ It creates objects, calls methods, & manages program flow flow.

\* Explain meaning of Public static void main (String[] args) :-

- Public : makes the main method accessible to JVM from anywhere.
- Static : Belongs to the class, not an object,
- Static & allowing JVM to invoke it without creating an instance of object.
- Void : The main method does not return any value.
- main : The method name which is recognized by the JVM as the Entry Point.
- String[] args : Holds command-line arguments which is passed to the program when executed.

[9]

Explaining what command line arguments are to add and write to program to do addition of 2 numbers.

- Command line arguments in Java allow a program to behave specifically based on input provided during execution.
- Definition :- Command line arguments are the parameters that one passes to the main class at the time method class is at the time of execution of program.
- Syntax :- use args[]
- We use String args[] as parameter in main method class & it holds the arguments which is passed at the time of execution of program.

↳ Public static void main (String args[]) {  
    }

→ Here args is declared as array of string.

→ Command line arguments are passed when we run the program :-  
To Compile : javac test.java  
To Run : java test Basic C++ C Java

→ These arguments are assigned to the array of args as follows:-

Basic → args[0]

C++ → args[1]

C → args[2]

JAVA → args[3]

\* Program :-

Public class add {

    Public static void main (String args[]) {

        int a = Integer.ParseInt (args[0]);

        int b = Integer.ParseInt (args[1]);

        System.out.Println ("The sum of a & b :"  
                              + (a + b));

}

Output :-

run : javac add.java

java add 5 5

output : The sum of a & b : 10

Q1] Explain read statement & Explain the meaning of `console().readline()`, write a program to print your name using `console().readline()`

→ The read statement is used to read input from the user, using Keyboard often from the console using Keyboard.

→ In Java, we typically use classes like `Scanner` or `BufferedReader` to read input.

→ Java Provides Several methods to read data from the console:

I] `Console()`: This method, allows the program to interact with the Commandline.

⇒ It is System class method.

⇒ When you call `System.console()`, it give you a way to read input from user through command-line or from where you run program.

II] `readline()`: This method Provided by the `Console` class.

→ It allow you to read a line of text that user type into command line.

→ After user write Enter the text & Press Enter, `readline()` return that text as String.

## \* Program :-

Public class name {

    Public static void main (String args[]) {

        String n = System.console().readLine();

        System.out.println(n);

Run : - javaC name.java

java name

Input : - Dhruv

Output : - Dhruv

III What is Scanner class, why we use Scanner class, find maximum among three numbers using Scanner class also find minimum number, write down application of this.

- The Scanner class in java is part of the `java.util` package & it is used to get input from the user.
- ⇒ It provides various methods to read different types of data such as integers, strings & floating-point numbers from user.

\* why use the Scanner class?

⇒ If

III user input :-

- The prime purpose of the Scanner class is to handle user input & accept the data from user during the program execution.

[2] Easy to use :-

⇒ It provides methods like `nextInt()`, `nextLine()` and `nextDouble()` that make it easier to read various types of input.

[3] Flexible:- It can read the data from various sources like keyboard, file, etc.

## \* Program 8

```
import java.util.Scanner;
```

```
Public class MaxMinFinder {
```

```
    Public static void main (String args[]) {
```

```
        Scanner in = new Scanner (System.in);
```

```
        System.out.print ("Enter the first number: ");
```

```
        int num1 = in.nextInt();
```

```
        System.out.print ("Enter the second number: ");
```

```
        int num2 = in.nextInt();
```

```
        System.out.print ("Enter the third number: ");
```

```
        int num3 = in.nextInt();
```

```
// Find max
```

```
        int max = num1;
```

```
        if (num2 > max) {
```

```
            max = num2;
```

```
}
```

```
        if (num3 > max) {
```

```
            max = num3;
```

```
}
```

```
// Find min
```

```
        int min = num1;
```

```
        if (num2 < min) {
```

```
            min = num2;
```

```
}
```

```
if (num3 < min) {
```

```
    min = num3;
```

3) If condition

+ max ab + min ab

+ max ab + min ab

System.out.println("Maximum number : " + max);

System.out.println("Minimum number : " + min);

Scanner file Dr. closed:

2)

Output :-

Enter the First number : 10

Enter the Second number : 20

Enter the third number : 30

Maximum number : 30

Minimum number : 10

\* Application of JSP :-

II Web application :-

⇒ JSP is widely used in building web application through Frameworks like Spring, Hibernate & JavaScript JSP Server Pages (JSP).

IP] Mobile application :-

⇒ Android development is primarily based on JSP, as it uses Android SDK.

### 13) Enterprise Software :-

→ Java is used for making large - scale Enterprise Systems, esp. especially for backend development.

### 14) Big Data Technologies :-

→ Java is used in processing big data through framework like Hadoop.

### 15) Gaming :-

→ Java is used for developing video games, especially for mobile & online games.

### 16) Scientific Applications :-

→ Java is used for making scientific & mathematical applications due to its high performance & rich libraries

X → K