




UNIT -1

INTRODUCTION TO JAVA

TOPICS TO BE COVERED

- 1.1 Basics of Java, Background/History of Java, Java and the Internet, Advantages of Java
 - 1.2 Java Virtual Machine & Byte Code
 - 1.3 Java Environment Setup
 - 1.4 Java Program Structure
 - 1.5 Procedure-Oriented Vs. Object oriented programming concept
 - 1.6 Basics of OOP: Abstraction, Inheritance, Encapsulation, Classes, subclasses and super classes, Polymorphism, message communication
 - 1.7 Compiling and running a simple "Hello World" program: Setting Up Your Computer, Writing a Program, Compiling, Interpreting and Running the program, Common Errors
- 

1.1 JAVA EVOLUTION

- Java is a general purpose, class based, and robust, secure, safe, object oriented programming language.
- It was developed by sun micro system in 1991 by James Gosling its name was **Oak**. In 1995 its name has been changed to **Java** because of some legal issues.
- Java was designed for development of software for consumer electronic devices like TVs, VCRs, toaster etc.
- Java removes limitation like portability and reliability of C and C++.
- Java program would run in fundamentally different execution environments.



- Java is strongly typed language.
- Compile time contains translating programs into a machine independent byte code representation.
- Runtime activities include loading and linking of classes needed to execute a program, optional machine code generation, dynamic optimization of program.
- Most striking feature of java is platform independence. Java is not tied to any particular hardware or OS. Java programs can be executed anywhere on any system.



JAVA FEATURES

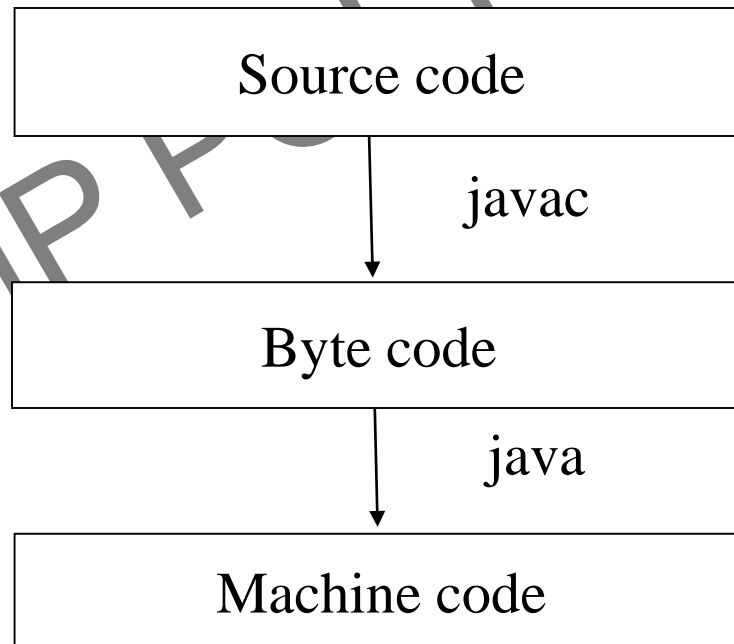
There are various features described by sun micro system.
Some of them are given here:

1. **Compiled and interpreted**
2. **Platform independent**
3. **Portable**
4. **Object oriented**
5. **Robust and secure**
6. **Distributed**
7. **Familiar , simple and small**
8. **Multithreaded and interactive**
9. **High performance**
10. **Dynamic and extensible**



1. Compiled and interpreted:

- Computer language is either compiled or interpreted. Java combines both so it is two stage system
- First **java compiler** translates **source code** into **byte code**.
- **Java interpreter** generates machine code that can be directly executing by machine that is running the program.



2. Platform independent:

- Programs written by user are called **source code** and executable program is called **object code**.
- Object can be executed by computer and object is specific to a particular CPU. So it cannot be executed on different platform. Java removes this limitation.
- When java program is compiled java compiler produces an object file which contains bytecode. These byte codes are not machine/CPU specific, it can be interpreted by JVM (Java Virtual Machine). So that code can be run on any JVM of computer system.
- The same byte codes can be executed by any JVM on any platform.
- Thus java is platform independent.



3. Portable :

- Java's important feature is portability.
- Java program can be easily moved from one computer system to another, anywhere and anytime. Changes /upgrade in OS .processors and system resources will not force any changes in java program.
- We can also download applet from a remote computer onto our local system via internet and execute it locally.
- Java provides Write Once Run Anywhere (WORA), which makes the java language portable provided that the system must have interpreter for the JVM .
- Java also has the standard data size irrespective of operating system or the processor .These features make the java as a portable language.



- Java provides portability in two ways:
- First, java compiler generates byte code instructions that can be implemented on any machine.
- Secondly, the size of primitive data types is machine-independent.

VPMP POLYTECHNIC



4. Object oriented

- Almost everything in java is an object.
- All program code and data reside within object and classes. Object model in java is simple and easy to extend.
- Java comes with extensive set of classes, arranged in packages that we can use in our program by inheritance.
- Java is true (pure) object oriented language because object is the outer most level of data structure in java.
- Everything in java is object even the primitive data types can also be converted into object by using the wrapper class.



5. Robust and secure

- It provides safe guards to ensure reliable code.
- Java has strict compile time and runtime checking for data types.
- Java also incorporates concept of exception handling, which captures series errors and eliminate any risk crashing system. So, java is robust.
- Java system verify not only all memory access but also ensures no viruses are communicated with applet.
- Absence of pointer in java ensures that program cannot gain access to memory location without proper authorization.



6. Distributed

- Java is designed as distributed language for creating applications on networks.
- Java applications can open and access remote objects on internet .These multiple programmers at multiple remote locations to collaborate and work together on a single project.

VPMP POLYTECHNIC



7. Simple ,small and familiar

- Java is small and simple.
- Many features of C and C++ those are redundant or sources of unreliable code are not part of java.
- It doesn't support pointer,pre-processor , header files,goto statement and other
- It eliminates operator overloads, multiple inheritance, thus java is simplified version of C++.
- It includes automatic storage management, typically using a garbage collector, to avoid safety problems of explicit de-allocation of memory references.



8. Multithreaded and interactive

- Java is multithreaded programming language.
- Multithreading means a single program having different threads executing independently at the same time. Multiple threads execute instructions according to the program code in process or a program.
- Java supports multithreaded programs. This means we need not wait for the application to finish one task before beginning other.
- **Java runtime** comes with tools that support microprocessor synchronization and construct smoothly running interactive system.



9. High performance

- High performance ,by intermediate byte code
- Multithreading enhances the overall execution speed of java program

10. Dynamic and extensible

- Java is dynamic language
- It is capable of dynamically linking in new class libraries, method and object.
- Java program support function written in other language such as C and C++.This function known as native methods.
- Native methods are linked dynamically at runtime.



Note :

Java does not run on Windows 95 and MS-Dos because on DOS and Windows 95 file names have extension having 3(.htm or .txt) character while java file has extension having 4 characters (.java).

VPMP POLYTECHNIC



1.1 ADVANTAGES OF JAVA

- Java syntaxes are inherited from C/C++. In C++ many concepts are confused which are left out or more cleared in java.
- Java is easy to program because it eliminates the drawback of other languages like pointers, memory management etc. that affect code of robustness.
- Java is object oriented language, which helps the programmers to visualize the program in real life terms
- Java enables users to create smooth and clear code.
- Java is called portable language because it is single code can be run on any operating system.



- Java is platform independent, so it allows the same software to execute without change on heterogeneous set of devices.
- Java provides multi-threading, so user can run more than one thread activity at a time.
- Java codes are checked before loading and so they are secured from run time errors.
- Java loads classes dynamically, at the time when they are actually needed.
- Through inheritance, we can eliminate redundant code and extend the use of existing classes.



- The principle of data hiding helps the programmer to build secure programs that cannot be attacked by code in other parts of program.
- It is possible to map object in problem domain in the program.
- The data centred approach enables us to capture more details of a model in an implementable form.
- Object oriented systems can be easily upgraded from small to large systems.
- Java strongly associated with Internet.



1.1 JAVA AND INTERNET

- Java is strongly associated with internet because the first application program written in java was **Hot java**.
- Internet users can use java to create applet programs and run them locally using *java enabled web browser*
- internet users can also setup their web sites containing java applets that could be used by other remote users of internet.
- The ability of java applets made java a unique programming language for internet.



JAVA AND WWW (WORLD WIDE WEB)

- WWW is an open ended information retrieval system ,designed to be used in internets distributed environment.
- WWW contains *web pages which* provides *information and control.*
- We can navigate to a new document in any direction.
- HMTL tags that enables user to find ,retrieve, manipulate and display documents world wide.



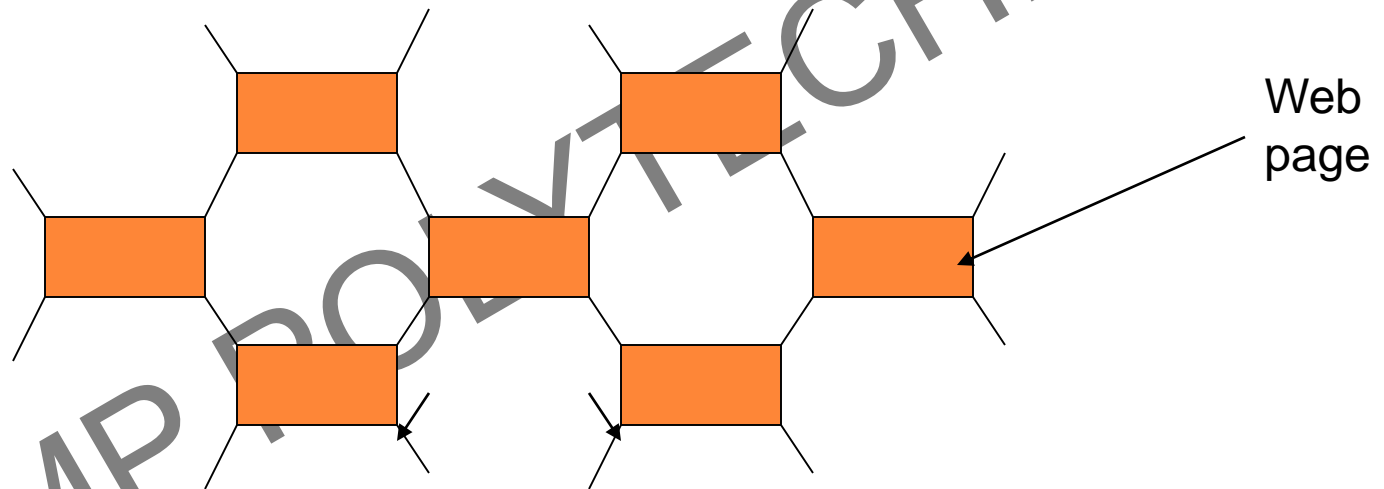


Figure :1



- Before java, WWW was limited to display of still (means motionless) images and texts.
- But incorporation of java into web pages ,web pages are capable of supporting animation, graphics, games and a wide range of special effects.
- Using java, the web has become more interactive and dynamic
- Using web, we can run java program on someone else's computer across the internet.
- Java communicates with a web page through special tag<APPLET> by following way:



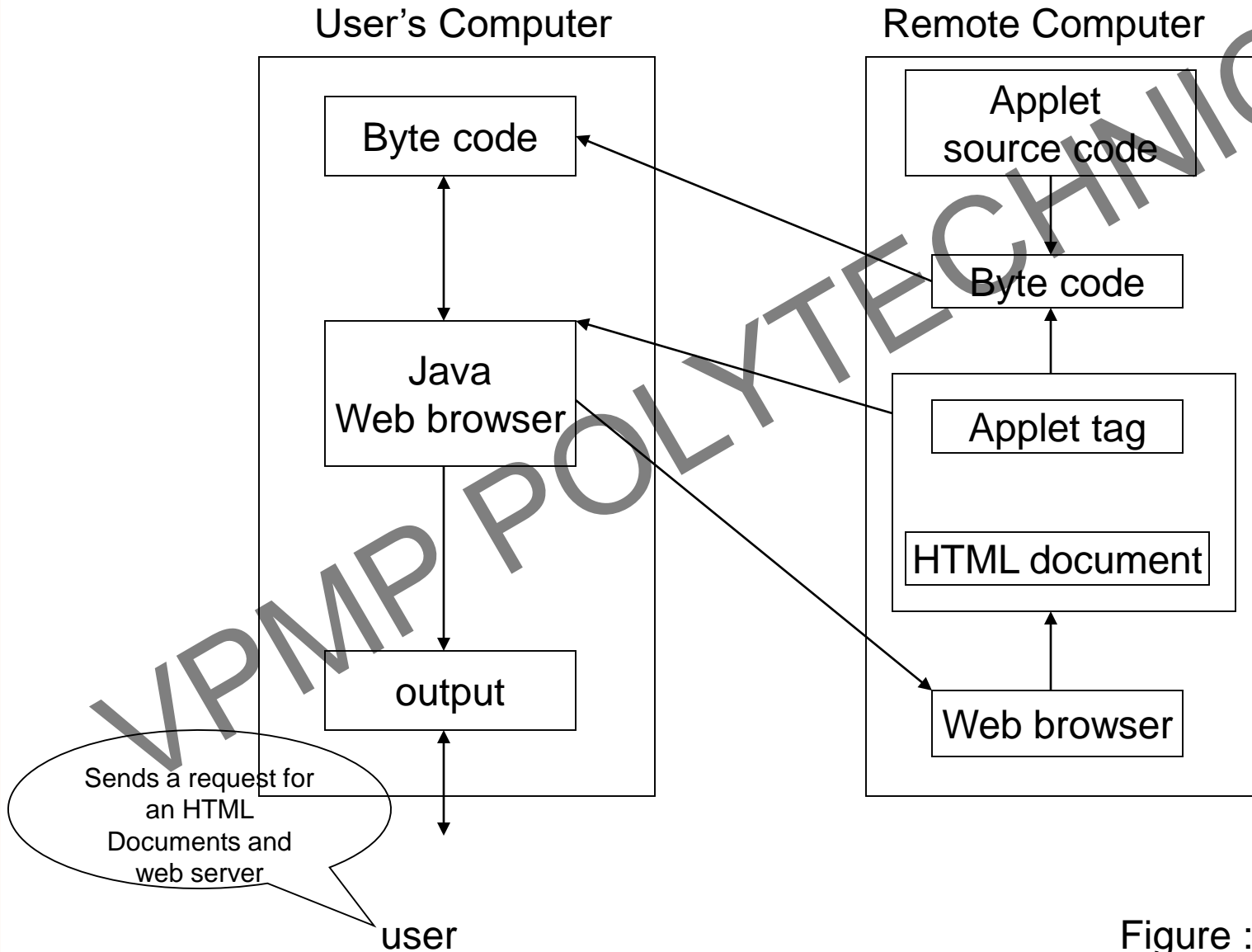


Figure :2

STEPS:

1. User sends a request for an HTML Document's web server. (Web server is a program that accepts a request , process the request and sends the required document)
2. HTML document is returned to the user's browser document contains APPLET tag which identifies applet.
3. Corresponding applet byte code is transferred to user's computer by java compiler
4. Java enabled web browser on user's computer interprets the byte code and provides output.
5. User may have further interaction with applet but with no further downloading from providers web server.



WEB BROWSERS

- Web browsers are used to navigate through the information found on the net.
- Browsers allows users to retrieve information spread across internet and display it using HTML.
- Web browse in java:
 - **Hot Java**
 - **Netscape Navigator**
 - **Internet Explorer**



1) HOT JAVA

- Browser from *sun micro system*.
- It enables the display of interactive content on the web using java language.
- **Hot java** written completely in java .
- It is currently available for the SPARC /Solaris platform as well as windows 95 and windowsNT.



NETSCAPE NAVIGATOR

- Browser from Netscape communication Co. is general purpose browser that can run java applets.
- It was most widely used browser at that time.
- It has visual display about downloading process and indication of the no. of bytes downloads.
- It supports java scripts ,a scripting language used in HTML documents.



INTERNET EXPLORER

- Browser for Micro windows 95 an NT.
- Navigator and explorer uses toolbars ,icons, menus and dialog boxes for easy navigation.
- Explorer uses JIT (just in time complier), which increases the speed of execution.



1.2 JAVA VIRTUAL MACHINE (JVM)

- All language compilers translates source code into machine code for a specific computer.
- Java is platform independent or machine neutral!!!
- Java compiler produces an **intermediate code (byte code or virtual machine code)** for a machine that does not exist. that machine is Java Virtual Machine(**JVM**).
- **JVM** exists only inside the computer memory.
- **JVM** is a simulated computer within the computer and does all major functions of real computer.



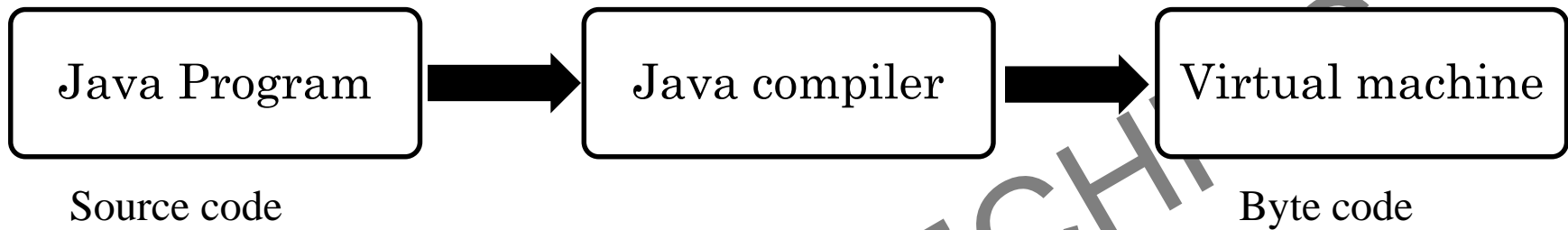


Figure :4.1 Process of compilation

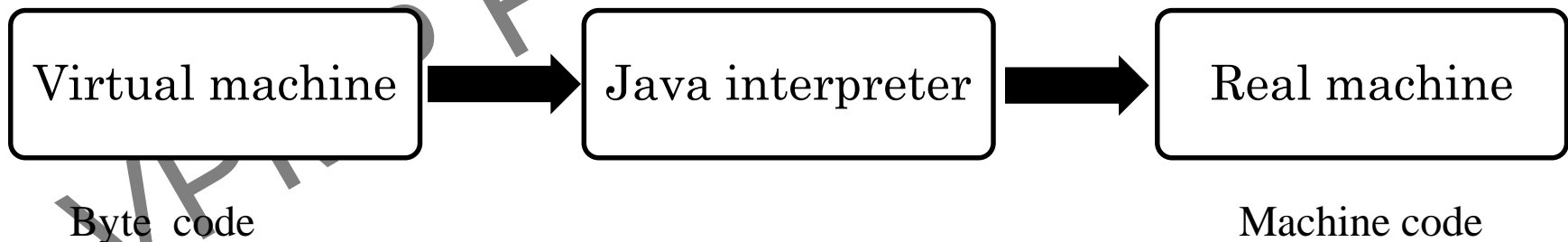


Figure :4.2 Process of converting byte code into machine code



- Byte code is not machine specific.
- Machine code is generated by the java interpreter by acting as intermediately between the virtual machine and real machine.

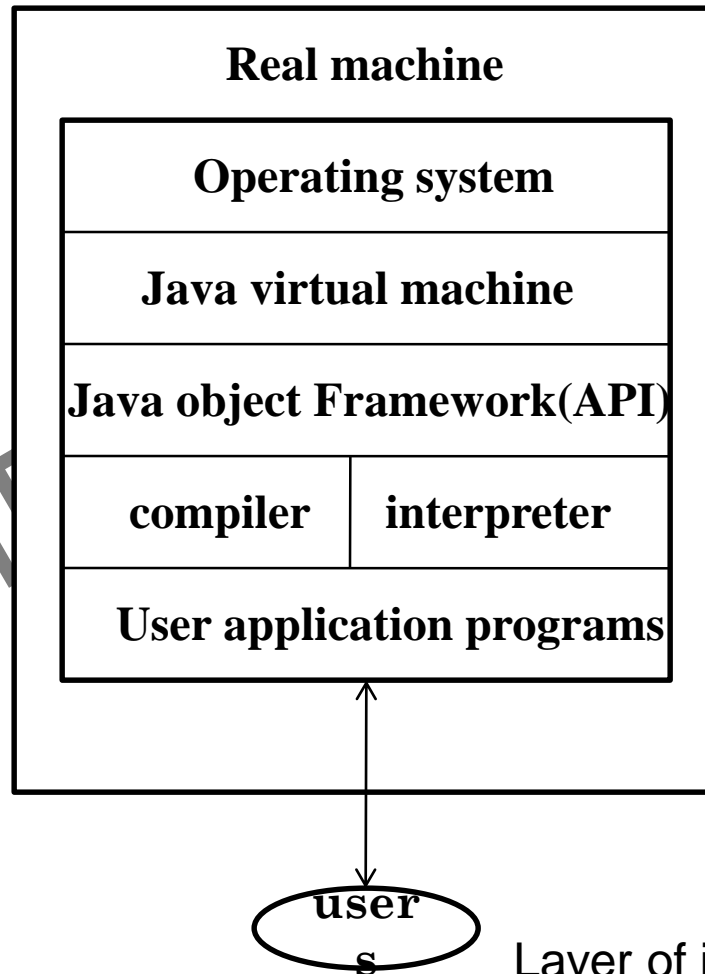


Figure :5

Layer of interactions for java program

JIT (JUST IN TIME COMPILER)

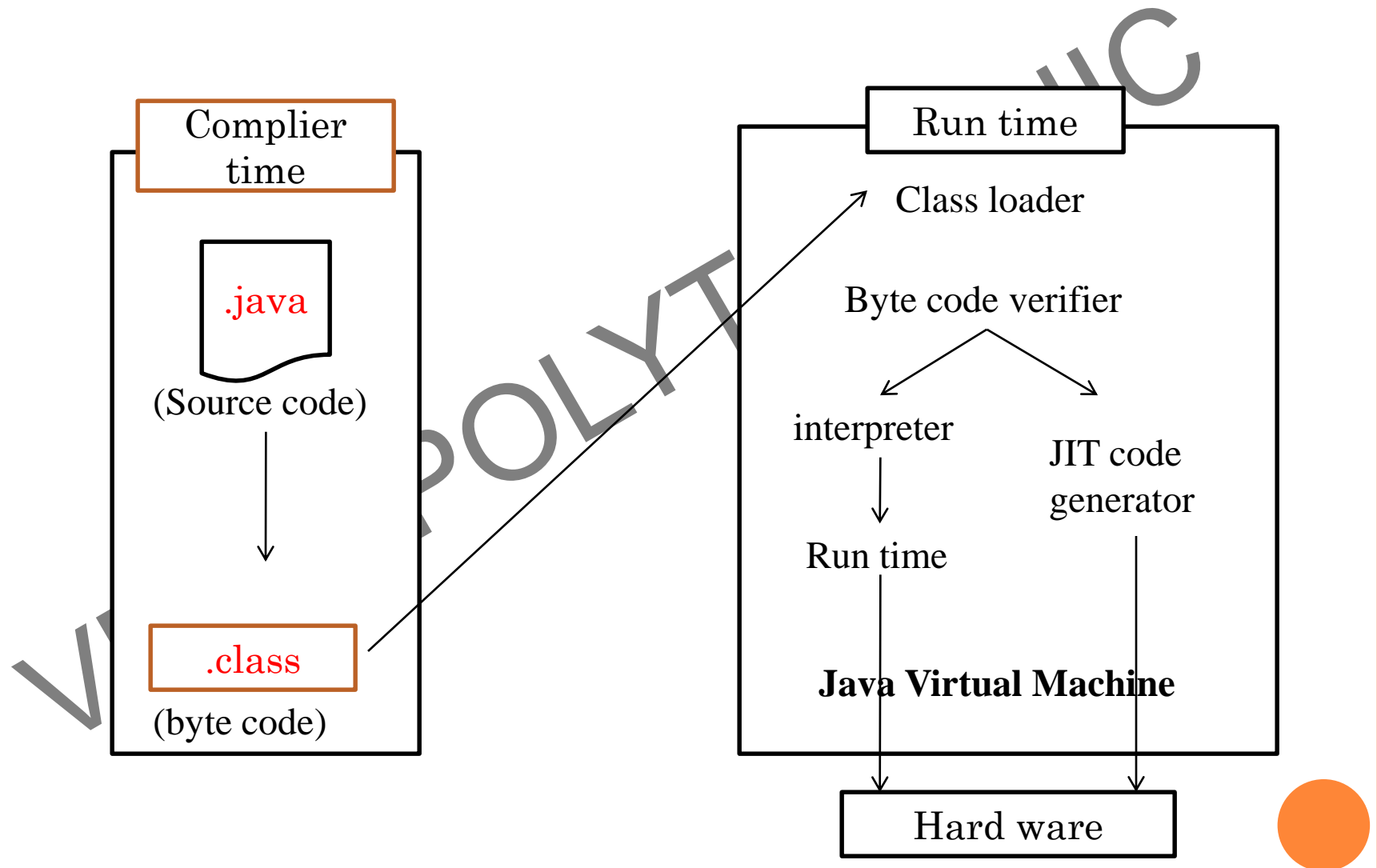
- JVM includes optional JIT.
- JIT dynamically compiles byte code into executable code.
- It takes byte code ,compiles them into machine code(native code) for machine.
- It compiles program on a method by method basis just before they are called.
- JIT is faster than JVM.



JRE (JAVA RUNTIME ENVIRONMENT)

- JRE consists of JVM and the program on the other .
- It runs code compiled for JVM by:
 - Loading the .class files
 - Verifying byte code
 - Executing the code





JAVA SUPPORT SYSTEM

- Internet connection
- Web server
- Web browser
- HTML
- Applet tag
- Java code
- Byte code

VPMP POLYTECHNIC



1.3 JAVA ENVIRONMENT SETUP

JE (JAVA ENVIRONMENT)

- JE includes large no. of development tools and hundred of classes and methods.
- Tools are part of JDK (java development toolkit) and classes and methods are part of JSL (API)
- $JE = JSL + JDK$

JDK : **J**ava **D**evelopment **T**oolkit

JSL : **J**ava **S**tandard **L**ibrary

API : **A**pplication **P**rogramming **I**nterface

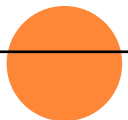


JDK (JAVA DEVELOPMENT TOOLKIT)

- It is a collection of tools that are used for developing and running java program.
- JDK includes
 - **applet viewer** (for viewing java applets)
 - **javac** (java compiler)
 - **java** (java interpreter)
 - **javap** (java disassembler)
 - **javah** (for C header files)
 - **javadoc** (for creating HTML documents)
 - **jdb** (java debugger)



No	Tools	Description
1	appletviewer	It enables user to run java applets.
2	javac (java compiler)	javac translates java source code to byte code files that the interpreter can understand.
3	java (java interpreter)	It runs application by reading and interpreting byte codes.
4	javap(java disassembler)	It enables users to convert byte code files into a program description.
5	javah(for C header files)	It Produces header files for use with native methods.
6	javadoc (for creating HTML Documents)	It creates HTML format documentation from java source code files.
7	jdb (java debugger)	It helps user to find error in our program.

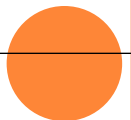


JSL :**JAVA STANDARD LIBRARY** OR **API** :**APPLICATION PROGRAMMING** **INTERFACE**

- It includes hundreds of classes and methods 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	Language support package (lang)	It includes classes and methods required for <u>implementing basic feature of Java.</u>
2	Utility package (util)	Collection of classes to provide <u>utility functions</u> such as date and time functions.
3	Input output package(io)	A collection of classes required for <u>input/output manipulation.</u>
4	Networking package(net)	Collection of classes for <u>communicating with other computers via internet.</u>
5	Abstract Window toolkit package(awt)	Collection of classes that <u>implements Platform independent Graphical User Interface</u>
6	Applet package(applet)	Collection of classes that <u>allows us to create java applets.</u>



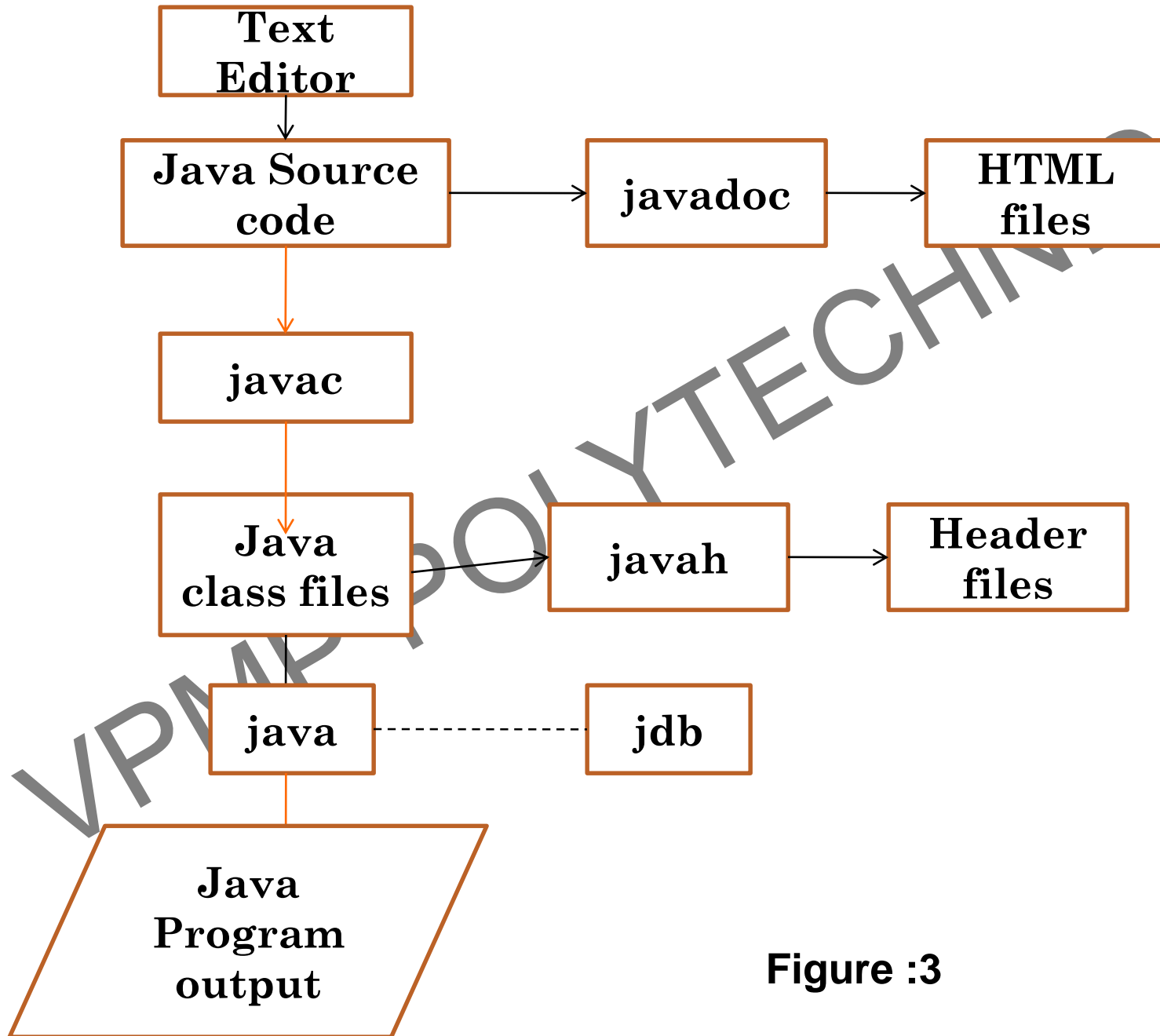


Figure :3



IMPLEMENTING JAVA PROGRAM

- Java application program implementation involves a three steps:
 - Creating the program (.java file)
 - Compiling the program(using `javac`)
 - Running the program (using `java`)



1.4 JAVA PROGRAM STRUCTURE

BASIC STRUCTURE OF JAVA PROGRAM

- Java program may contain many classes of which only one class defines a main method.
- Class contain data members and methods that operate on data members of class.
- Method may contain data type declaration and executable statements.java program define classes and put them together.



Documentation section	← Suggested
Package statement	← optional
Import statement	← optional
Interface statement	← optional
Class definitions	← optional
Main method class { Main method definitions }	← Essential

Basic structure of java program



DOCUMENTATION SECTION:

- This section comprises a set of comment lines giving the name of the program, author and other details, which the programme would like to refer to at a large stage.
- Comments must explain why and what of classes and how to algorithms.
- Java supports three types of comment line:
 - **Single comment line** : //
 - **Multiline** : /* */
 - **Documentation line** : /** */



PACKAGE STATEMENT:

- The statement declares a package name and informs the compiler that the classes defined here belong to this package.
- `package student ;` //student is package

package statement is optional.



IMPORT STATEMENTS:

- After package statement (but before class definitions) be a number of import statements. This is similar to the #include statement in C.

import packagename.classname;

- This statement instructs the interpreter to load the **test** class contained in the package **Student**.
- Using import statement ,we can access to classes that are part of the other named packages.



INTERFACE STATEMENT:

- Interface is like a class but includes a group of method declarations. This is also an optional section.
- It is used only when we want to implement the multiple inheritance feature in the program.

VPMP POLYTECHNIC



CLASS DEFINITIONS:

- A java program may contain multiple class definitions. Classes are the primary and essential elements of a Java program. The number of classes used depends on the complexity of the problem.

VPMP POLYTECHNIC



MAIN METHOD CLASS:

- Every java stand alone program requires a main method as its starting point, this class is essential part of a java program.
- Simple java program may contain only this part.
- The main method creates objects of various classes and creates communications between them.
- On reaching the end of main ,the program terminates and control passes back to the operating system.



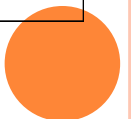
1.5 PROCEDURE-ORIENTED Vs. OBJECT ORIENTED PROGRAMMING CONCEPT



PROCEDURE-ORIENTED Vs. OBJECT ORIENTED PROGRAMMING

No	Procedure oriented programming	Object oriented programming
1	Program is divided into small parts called functions.	Program is divided into parts called objects.
2	Importance is not given to data but to functions as well as sequence of actions to be done.	Importance is given to the data rather than procedures or functions because it works as a real world.
3	POP follows Top Down approach.	OOP follows Bottom Up approach.
4	POP does not have any access specifier.	OOP has access specifier named Public, Private, Protected
5	In POP, Data can move freely from function to function in the system.	In OOP, objects can move and communicate with each other through member functions.

6	To add new data and function in POP is not so easy.	OOP provides an easy way to add new data and function.
7	In POP, Most function uses Global data for sharing that can be accessed freely from function to function in the system.	In OOP, data cannot move easily from function to function, it can be kept public or private so we can control the access of data.
8	POP does not have any proper way for hiding data so it is less secure.	OOP provides Data Hiding so provides more security.
9	In POP, Overloading is not possible.	Overloading is possible in the form of Function Overloading and Operator Overloading.
10	Examples of POP are: C, VB, FORTRAN, and Pascal.	Example of OOP are : C++, JAVA, VB.NET, C#.NET.



JAVA ,C AND C++

Java and C

- Java is an object oriented language has mechanism to define class and objects.
- To build up a simple and safe language, java team did not include some of C features in java.

Java does not include

- C unique statement goto ,sizeof ,typedef
- Data types like struct ,union and enum
- Modifiers keywords like auto ,extern, register ,signed and unsigned
- Explicit pointer type



- No pre-processor so it does not use #define, #include and #ifdef
- java does not support any mechanism for defining variable arguments to function

Java adds,

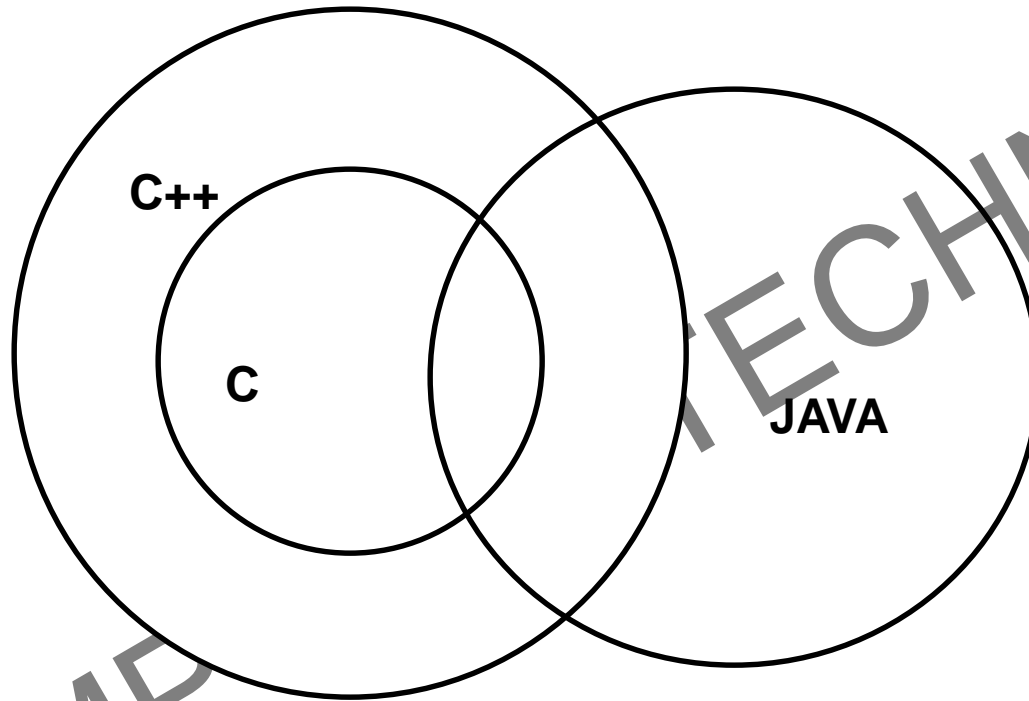
- Java functions with no arguments must be declared with empty parameters and not with the **void** keyword
- It adds new operator , instanceof and >>>
- It uses break and continue



JAVA AND C++

- Java is true object oriented language while C++ is basically C with object oriented extension.
- ***Java does not support***
 - Operator overloading
 - Multiple inheritance
 - Global variable
 - Explicit pointer
 - Header files
 - Garbage collector ,but java uses finalize () method
 - Not have template class as C++
- C++ is superset of C. Java may be considered as first cousin of C++ and a second cousin of C.





1.6 BASICS OF OOP: ABSTRACTION, INHERITANCE, ENCAPSULATION, CLASSES, SUBCLASSES AND SUPER CLASSES, POLYMORPHISM, MESSAGE COMMUNICATION

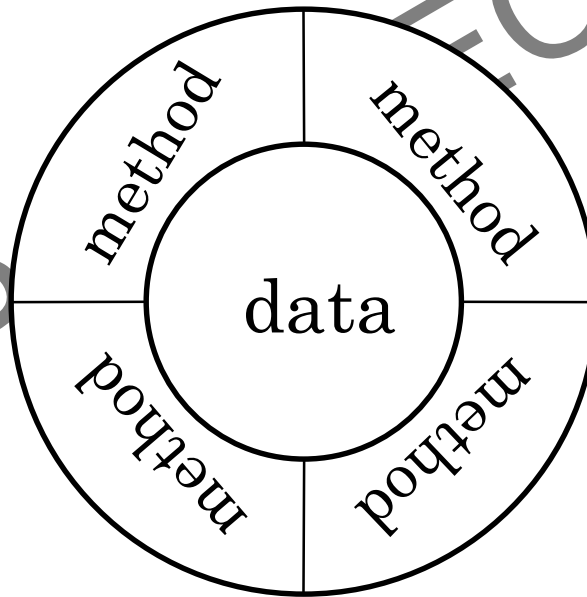


BASICS OF OOP'S

- OOP is an organization and development, which attempts to eliminate some of the pitfall of converting programming method.
- OOP treats data at a critical element in the program development and doesn't allow it to flow freely around the system .
- It ties data more closely to the function that operate on it and protect it from unintentional modification by other function.
- OOP allows us to decompose a problem into a number of entities called object. And build data and functions around these entities.



- Object= Data + Method
- Data of object can be accessed only by the method associated with that project.

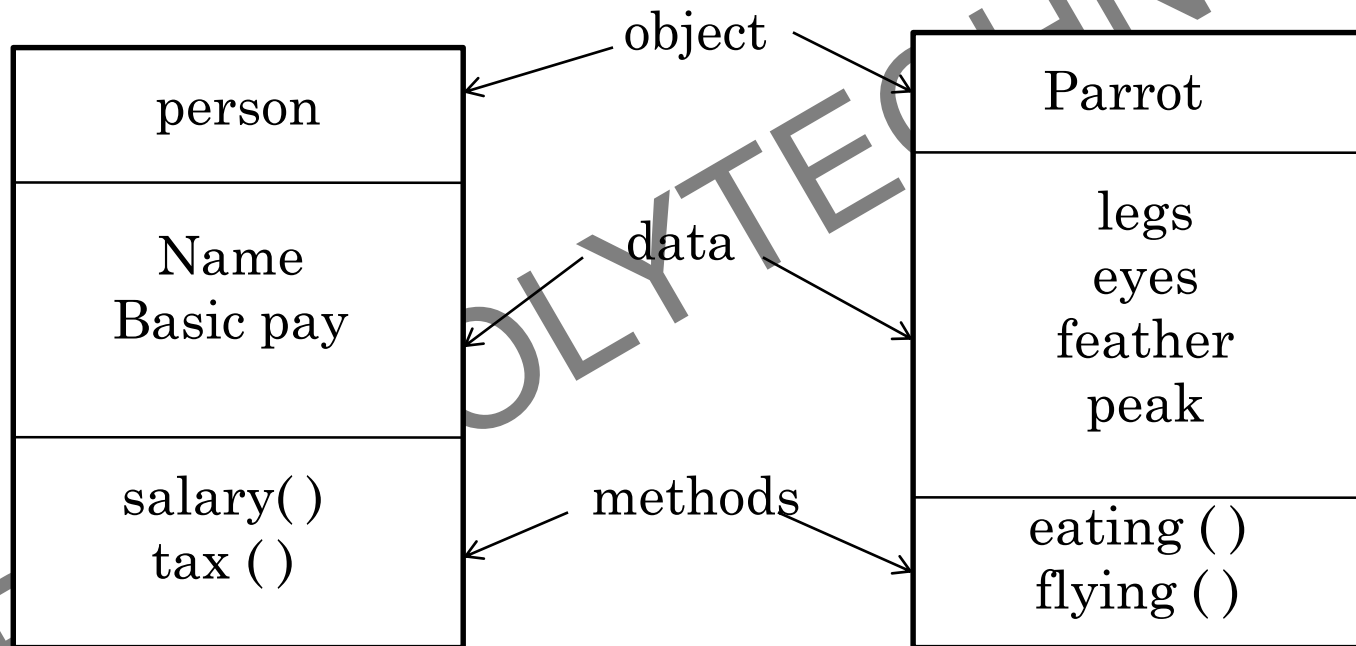


BASIC CONCEPT OF OBJECT ORIENTED PROGRAMMING

1) Object and classes

- Objects are basic runtime entities in object oriented system.
- Object may represent a person, place, chair, or any item.
- Object can interact without having to know the details of each other's data or code.
- Object is a basically a set of data and code to perform operation on data

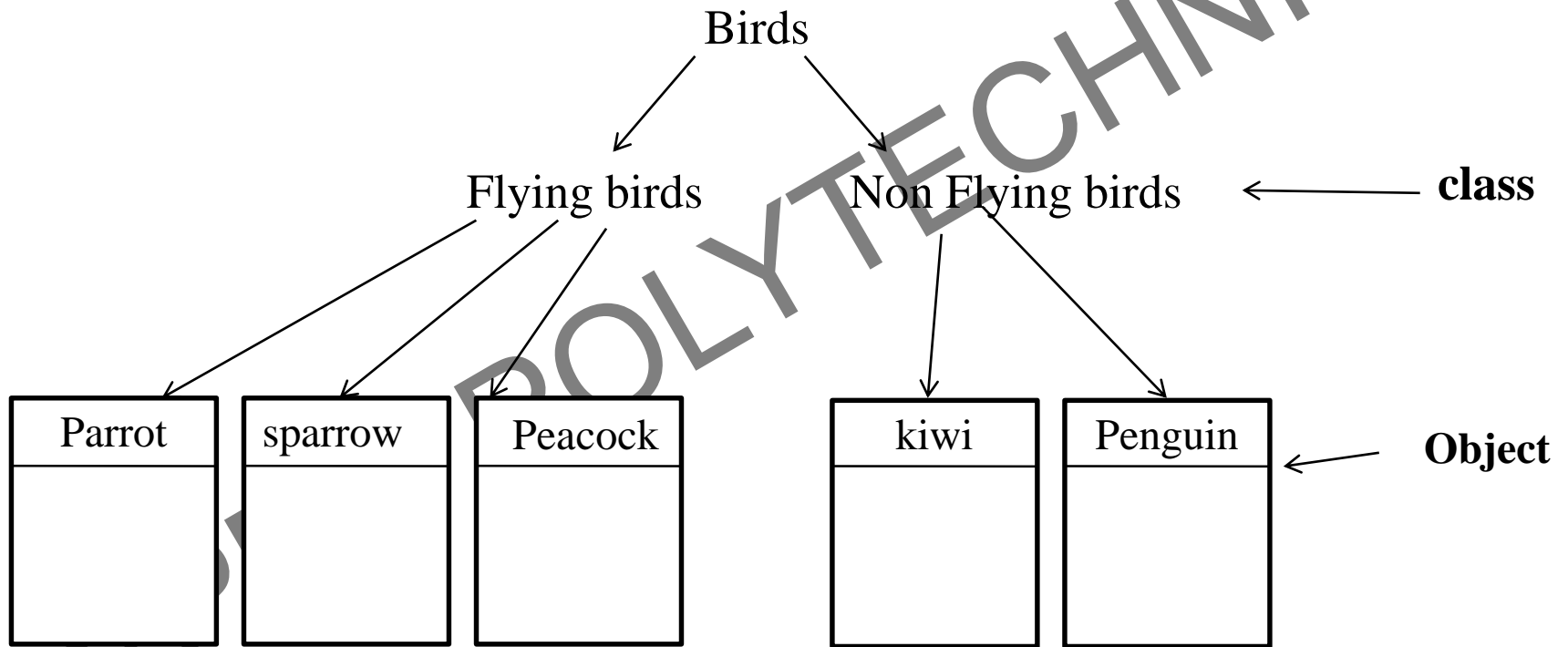




2) CLASS

- Class is prototype that defines the variables and the methods common to all object of similar type.
- Classes are user defined data types and behave like built in type of programming language.
- Simply, Class is a collection of logically related data items which includes data (variable) and function (methods) for data.





3) Data abstraction and Encapsulation:

- **Data abstraction** refers to act of representing essential features without including background details or explanation.
- **Data encapsulation** means wrapping up of data and methods into a single unit.
- The data is not accessible to the outside and only these methods, which are wrapped in class access it this insulation of the data from direct access by program is called **data hiding**.



4) Inheritance

- **Inheritance** is a process by which object of one class use the properties of object of another class.
- It supports hierarchical classification.
- It provides Reusability of code. We can add an additional feature to an existing class without modifying it .This is possible by deriving a new class from the existing one.
- New class with additional features can be created, that new class with have combined features of both the new classes.



5) Polymorphism

- **Polymorphism** means the ability to take more than one form. Simply, it means that are the same operation have may behave different on different classes.

6) Dynamic Binding

- Binding refers to the linking of procedure call to the code to be executed in response to call.
- Dynamic binding means code associated with a given procedure call is not know until the time or call at runtime.
- It is associated with polymorphism and inheritance.

7) Message Communication

- Object oriented program consist of a set Object communicates with one another by sending and receiving information.

1.7 COMPILING AND RUNNING A SIMPLE "HELLO WORLD" PROGRAM: SETTING UP YOUR COMPUTER, WRITING A PROGRAM, COMPILING, INTERPRETING AND RUNNING THE PROGRAM, COMMON ERRORS



SIMPLE JAVA PROGRAM

```
class Test
{
    public static void main(String args[ ])
    {
        System.out.println( "welcome to Java");
    }
}
```




1) First line : **class Test**

- declares a class which is an object oriented construct.
- java is true object oriented language and therefore everything must be placed inside a class.
- **class** is keyword and it declares that a new class definition follows. **Test** java identifier(name) specifies name of the class to be defined.

2) Second line : **Opening brace : {**

3) Third Line : **public static void main (String args[])**

Every java program must include main() method. It is the starting point for interpreter to begin the execution of program.



MAIN () METHOD OF JAVA

- **public:** The keyword public is an access modifier. It declares main method as unprotected. It makes method accessible to all other classes.
- **static :** It is a keyword, which declares method is for entire class and not a part of any object of class. The main must always be declared as static since the interpreter uses this method before any objects are created.
- **void** does not return any value.
- **String args** declares parameter names **args** which contain an array of object of class type string.



4) Output line:

System.out.println (“Hello”);

class

object

method



- This line is similar to `printf ()` statement of C or `cout<<` construct of C++.
- Since java is true object oriented, every method must be part of an object.
- **println** method is a member of the **out** object, which is a data member of a **System** class. **println** method always append a new line character to the end of the string. This means subsequent output will start on a new line.
- Semicolon at the end of each statement is used. **Every java statement must end with semicolon.**



5) Fifth line :**closing brace : }**

It is the ending of program.

- Note: java program must name same as in class which includes **main() method** otherwise **interpreter cannot run that class.**

VPMP POLYTECHNIC



DIFFERENCE BETWEEN PRINT() AND PRINTLN() METHOD

print () method: print and wait

- This method sends information into a buffer. This buffer is not flushed until a newline (or end of line) character sent. As a result **print() method** prints out on one line until a new character is encountered.

`System.out.print (“hello”);`

- We can also use `\n` to force the display to be brought to the next line by printing a newline character as follows:
- `System.out.print (“ hello \n”);` `//this would work same as println () method`



println () method : prints a line and move to next line

- This method takes the information provided and displays it on a line followed by a line feed.

```
System.out.println( "java !!!");
```



COMMAND LINE ARGUMENT

- If we want, our program to act in a particular way depending on the input provided at the time of execution.
- This is done in java programs by using command line arguments.
- Definition: **command line arguments are parameters that are supplied to the application program at the time of on invoking it for execution.**
- We can write java program that can receive and use the arguments provided in the command line.

public static void main (String args[])



- **args** is declared as an array of strings(known as string object) .
- Any arguments provided in the command line are passed to the array **args** as its elements.
- We can simply access the array elements and use them in the program as we want.
- Command line arguments are passed when we run the program

To compile: `javac Test.java`

To run : `java Test Basic C++ C Java`

- This command line contains four arguments .These arguments are assigned to the array **args** as follows:

Basic	→	args[0]
C++	→	args[1]
C	→	args[2]
Java	→	args[3]



```
class Cmd1
{
    public static void main (String args[ ])
    {
        int count=0;
        String str;
        count=args.length; //to find how many arguments
        are passed to command line
        System.out.println ( "total no. of arguments passed = "
        + count );
        while (          i < count )
        {
            str= args[ i ];
            i=i+1;
            System.out.println ( i + " : " + "Java is " +
            str + " !!!" );}}}

```


E: \java program \ javac Cmd1.java

E: \java program \ java Cmd1 **simple robust secure
portable dynamic object_oriented**

Output :

total no. Of arguments passed= 6

1 : Java is simple !!!

2 : Java is robust !!!

3 : Java is secure !!!

4 : Java is portable !!!

5 : Java is dynamic !!!

6: Java is object_oriented !!!



READ STATEMENT

- We may also give values to variables interactively through the keyboard using **readLine()** method.
- **readline()** is invoked using an object of the class **DataInputStream**.
- **readLine()** reads the input from the keyboard as a string which is then converted to the corresponding data type using the data wrapper classes.
- We have used the keywords **try** and **catch** to handle any errors that might occur during the reading process.



Example :

```
import java.io .DataInputStream;
class Reading
{   public static void main (String args [ ])
{
DataInputStream obj=new DataInputStream(System.in);
    int i=0;
    float f=0.0 F;
    try
    {
        System.out.println (“enter an integer no “);
        i= Integer.parseInt ( obj. readLine ( ) );
        System.out.println (“enter float no “);
        f = Float.valueOf (obj.readLine ( )).floatValue(());
    }
}
```

```
catch(Exception e)
```

```
{
```

```
    System.out.println ( "enter proper value" );
```

```
}
```

```
    System.out.println( "int no i= " + i );
```

```
    System.out.println( "float no f= " + f);
```

```
}
```

```
}
```

VPMP POLYTECHNIC



OUTPUT:

```
E :\ java program\ javac Reading.java
```

```
E :\ java program\ java Reading
```

```
enter an integer no
```

```
56
```

```
enter float no
```

```
23.455
```

```
int no i= 56
```

```
float no f=23.455
```



VPMP POLYTECHNIC

Thank you

