



Lecture 01



Contents

- What is Programming?
- Different approaches for Programming
- Different Programming Languages
- Difference between C,C++,Java
- What is Java?
- Features of Java
- OO concepts
- Sample Program
- What is JDK?



What is Programming?

- Programming is somewhat like working with building blocks.
- The five basic elements in programming are:
 - **input**: getting data and commands into the computer
 - **output**: getting your results out of the computer
 - **arithmetic**: performing mathematical calculations on your data
 - **conditional**: testing to see if a condition is true or false
 - **looping**: cycling through a set of instructions until some condition is met

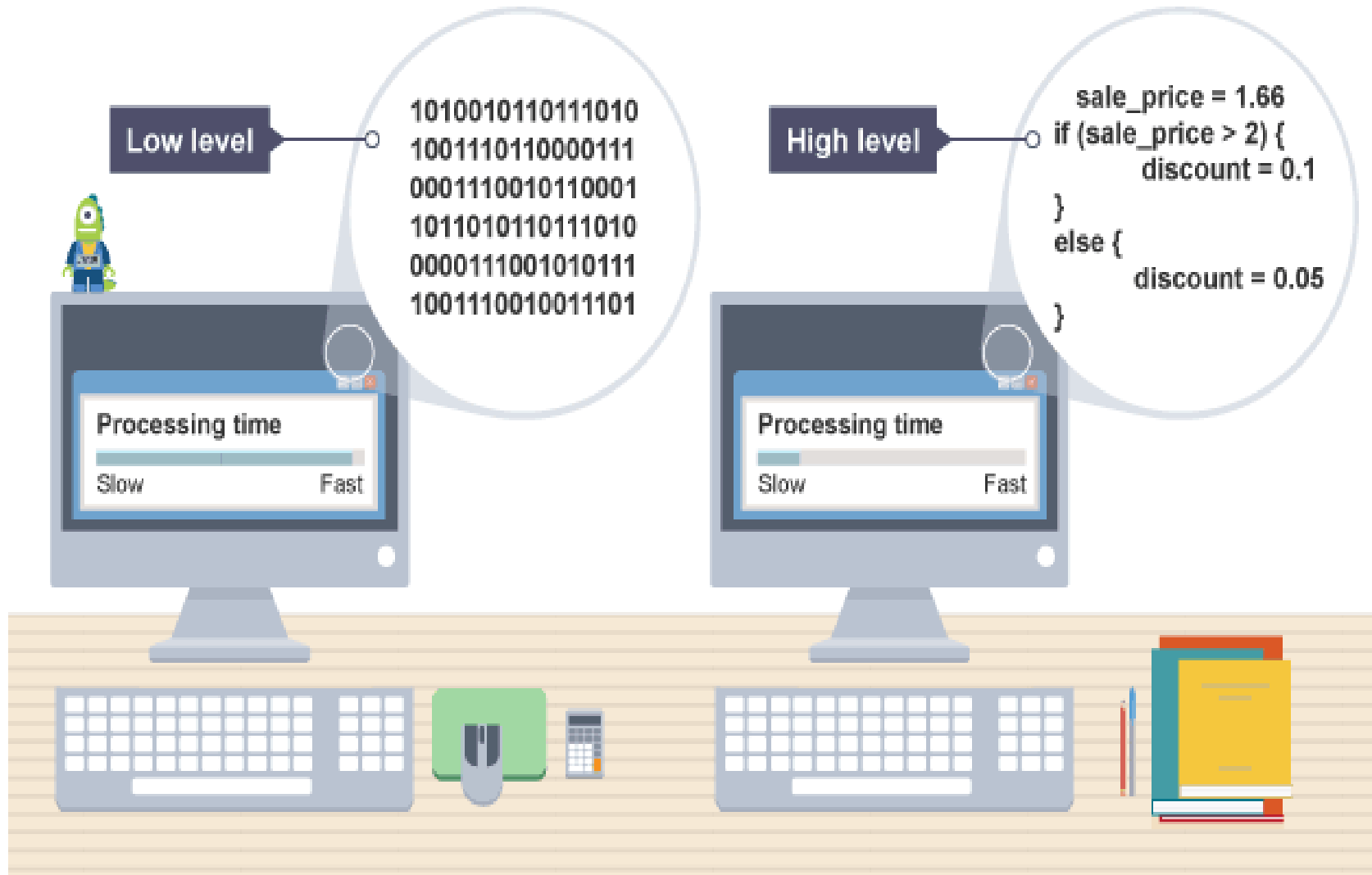
What is Computer Programming?

- Way of giving computers instructions about what they should do next.

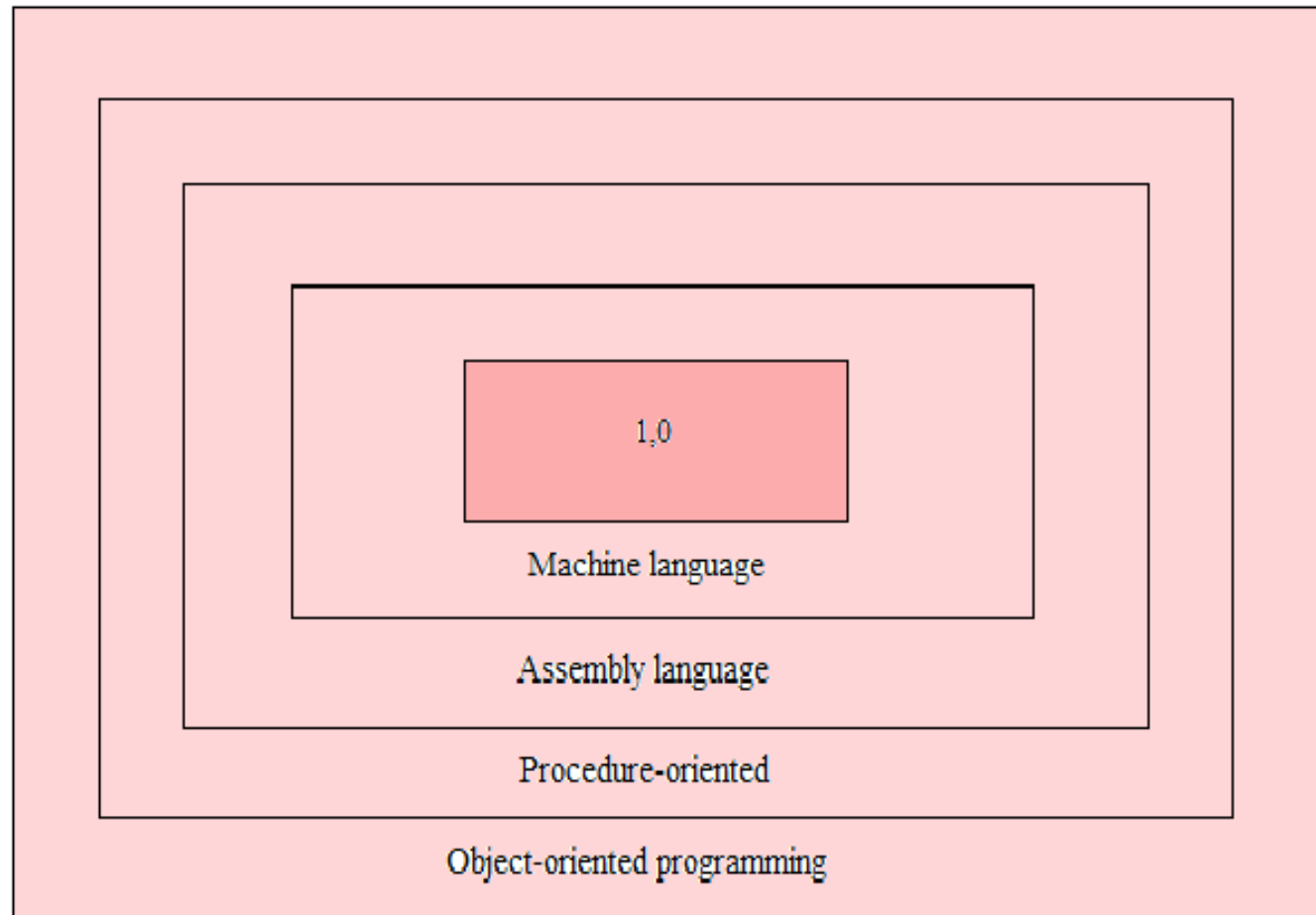
These *instructions are known as code*, and computer programmers write code to solve problems or perform a task.

- End goal is to create something:
Anything from a web page or a piece of software or even just a pretty picture.
- A computer program is a list of instructions that enable a computer to perform a specific task.
- Computer programs can be written in high and low level language, depending on the task and the hardware being used.

Difference between low level & high level



Difference between low level & high level



Layers of computer software

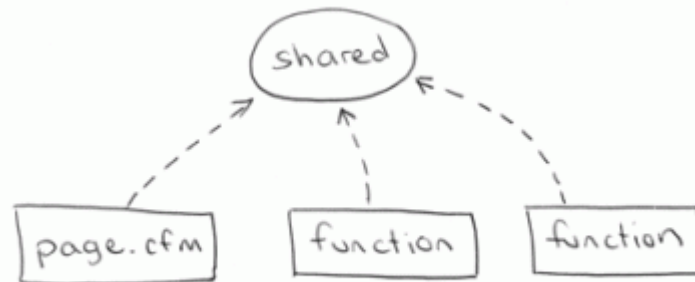


Approaches of Programming

- Procedural Programming Approach
- Structured Programming Approach
- Modular Programming Approach
- Object Oriented Programming Approach

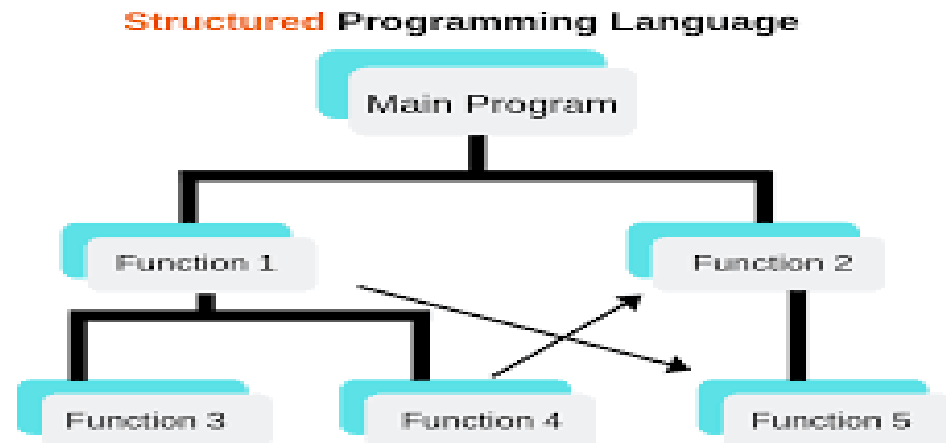
Procedural Programming Approach

- Procedural Programming is a subset of structured programming, **derived from structured programming**, based on the concept of the *procedure call*.
- Procedures, also known as routines, subroutines, or functions, simply contain a series of computational steps to be carried out.
- Software maintenance can be difficult and time consuming. Importance is given to data. Hard to reuse.
- Example: ALGOL, COBOL, BASIC, Pascal, Fortran



Structured Programming Approach

- Its main purpose to enforce a logical structure on the program being written to make it more efficient and easier to understand and modify.
- The **structured programming language** allows a programmer to code a program by dividing the whole program into smaller units or modules.
- Structured programming is not suitable for the development of large programs and does not allow reusability of any set of codes.
- E.g C





Procedural programming Vs Structured Programming

- A **procedural programming** language consists of a set of **procedure** calls and a set of code for each **procedure**. A **structural programming** language emphasizes on separating a **program's** data from its functionality.

Modular Programming Approach

- When creating a modular system, instead of creating a monolithic application (where the smallest component is the whole), several smaller modules are written separately so when they are composed together, they construct the executable application program. Typically these are also compiled separately, via separate compilation, and then linked by a linker. A just-in-time compiler may perform some of this construction "on-the-fly" at run time.
- In modular programming , each module perform logically discrete functions, interacting through well-defined interfaces.

Modular Programming Approach

- Modular programming is the process of subdividing a computer program into separate sub-programs. A module is a separate software component.
- It can often be used in a variety of applications and functions with other components of the system.
- Some programs might have thousands or millions of lines and to manage such programs it becomes quite difficult as there might be too many of syntax errors or logical errors present in the program, so to manage such type of programs concept of **modular programming** approached.

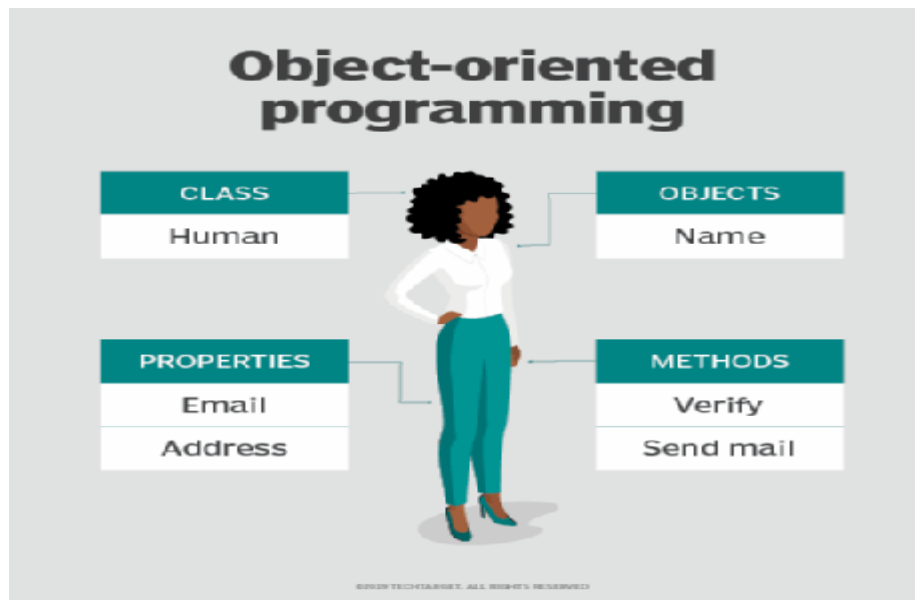
Modular Programming Approach

- Each sub-module contains something necessary to execute only one aspect of the desired functionality.
- Modular programming emphasis on breaking of large programs into small problems to increase the maintainability, readability of the code and to make the program handy to make any changes in future or to correct the errors.

e.g. header files in C

OO Programming Approach

- Object-oriented programming (OOP) is a programming language model in which programs are organized around data, or objects, rather than functions.
- An object can be defined as a data field that has unique attributes and behaviour.
- Example: Java, Python, JavaScript, C++ etc.



Difference Between C,C++,Java

Feature	c	C++	Java
Paradigms	Procedural	Procedural , OOP	Purely Object oriented
Pointers	Yes,Very commonly used	Yes, very commonly used, but some form of references available too.	No Pointers; references are used instead
Pre-processor directives	Supported (#include, #define)	Supported (#include, #define)	Not Supported
Header files	Supported	Supported	Use Packages (import)

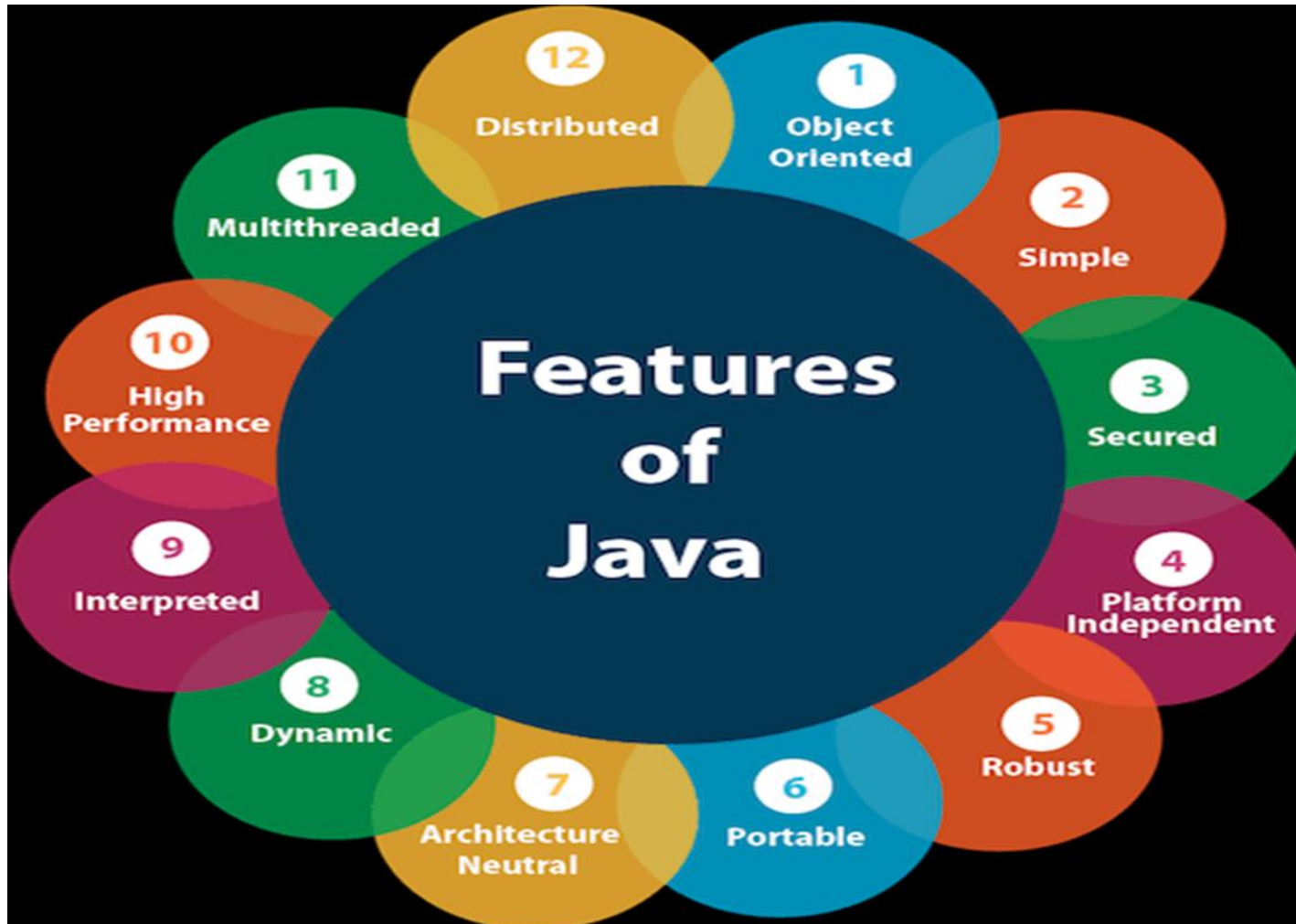
Storage Allocation	Uses malloc, calloc	Uses new, delete	uses garbage collector
Database Connectivity	Not Supported	Not Supported	Supported
Code Translation	Compiled	Compiled	Interpreted
Complex Data Types	Structures, Unions	Structures, Unions, Classes	Classes
String Type	Character ,Arrays	Character ,Arrays, Objects	Objects
Use of Exception handling	C does uses exception handling for exception generated while execution of program.	C++ makes use of exception handling for handling various types of exceptions generated while execution of program.	Java makes use of exception handling in a very effective way to handle exceptions generated while execution of program.

Operator overloading	There is no such concept of operator overloading in C	C++ uses concept of operator overloading	There is no such concept of operator overloading in Java
Multithreading and Interfaces	Not Supported	Not Supported	Supported
Inheritance	No Inheritance	Supported	Multiple Inheritance not Supported

What is java ?

- Java is a **programming language**.
- Java is a high level, robust, object-oriented and secure programming language.
- **Platform**: Any hardware or software environment in which a program runs, is known as a platform.
- It is object oriented programming program is composed of a collection objects that communicate with each other
- Named after coffee name Java. Previously named as Oak.
- Developed by James Gosling and grp by Sun Microsystems in 1995.

Features of java ?





Features of java ?

- Simple
- Object-Oriented
- Portable
- Platform independent
- Secured
- Robust
- Architecture neutral
- Interpreted
- High Performance
- Multithreaded
- Distributed
- Dynamic



Simple

- Java is very easy to learn, and its syntax is simple, clean and easy to understand. According to Sun, Java language is a simple programming language because
- Java syntax is based on C++ (so easier for programmers to learn it after C++).

Object Oriented

- Java is an **object-oriented** programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behaviour.
- Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.
- Basic concepts of OOPs are:
 - Object
 - Class
 - Inheritance
 - Polymorphism
 - Abstraction
 - Encapsulation

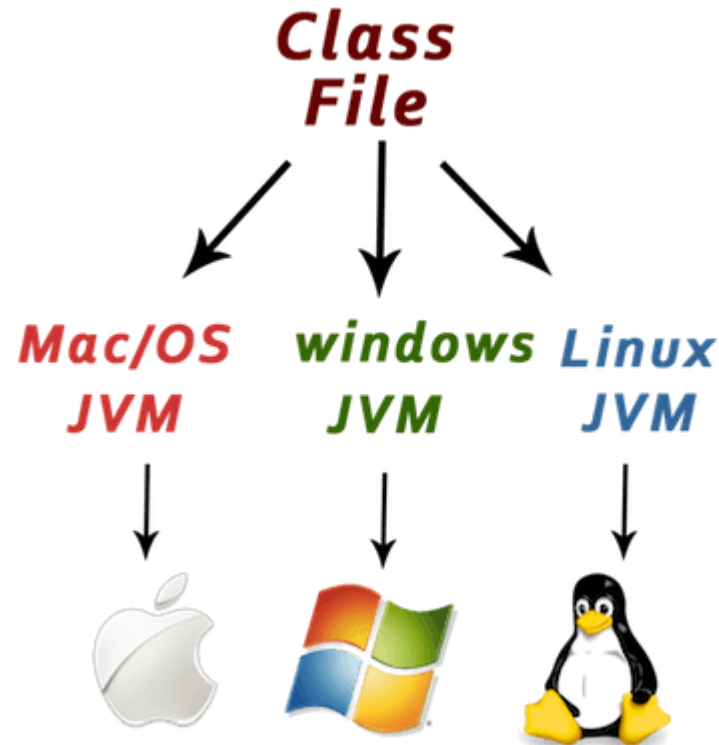


Portable

Java is portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.

Platform Independent

- Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines
- while Java is **a write once, run anywhere language**.





Secured

- Java is best known for its security. With Java, we can develop virus-free systems. Java is secured because:
- **No explicit pointer**
- **Java Programs run inside a virtual machine**



Robust

- Robust simply means strong. Java is robust because:
- It uses strong memory management.
- There is a lack of pointers that avoids security problems.
- There are exception handling and the type checking mechanism in Java. All these points make Java robust.



Architecture neutral

- Java is architecture neutral because there are no implementation dependent features, for example, the size of primitive types is fixed.



High Performance

- Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code. It is still a little bit slower than a compiled language (e.g., C++).
- Multithreading enhances the overall speed of Java programs.



Distributed

- Java is distributed because it facilitates users to create distributed applications in Java.



Multithreaded

- A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads.
- The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi-media, Web applications, etc.



Dynamic

- Java is a dynamic language. It supports dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.
- Java supports dynamic compilation and automatic memory management (garbage collection)



Object oriented concepts

-
- Class
- Object
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation
- Message Passing



Objects and classes in Java

- An object in Java is the physical as well as a logical entity, whereas, a class in Java is a logical entity only.
- An entity that has state and behaviour is known as an object e.g., chair, bike, marker, pen, table, car, etc. An object has three characteristics:
- **State: represents** the data (value) of an object.
- **Behaviour:** represents the behaviour (functionality) of an object such as deposit, withdraw, etc.
- **Identity:** An object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. However, it is used internally by the JVM to identify each object uniquely.



Objects in Java

- **An object is an instance of a class.** A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.
- **Object Definitions:**
 - An object is *a real-world entity*.
 - An object is *a runtime entity*.
 - The object is *an entity which has state and behavior*.
 - The object is *an instance of a class*.



classes in Java

- A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.
- A class in Java can contain:
- **Fields**
- **Methods**
- **Constructors**
- **Blocks**
- **Nested class and interface**



Inheritance in Java

- **Inheritance in Java** is a mechanism in which one object acquires all the properties and behaviours of a parent object. It is an important part of OOPs (Object Oriented programming system).
- The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.
- Inheritance represents the **IS-A relationship** which is also known as a *parent-child* relationship.

Polymorphism in Java

- **Polymorphism in Java** is a concept by which we can perform a *single action in different ways*. Polymorphism is derived from 2 Greek words: poly and morphs. The word "poly" means many and "morphs" means forms. So polymorphism means many forms.
- E.g. Person
- There are two types of polymorphism in Java: compile-time polymorphism and runtime polymorphism.
- We can perform polymorphism in java by method overloading and method overriding.



Abstraction in Java

- **Abstraction** is a process of hiding the implementation details and showing only functionality to the user.
- Another way, it shows only essential things to the user and hides the internal details, for **Example**: sending SMS where you type the text and send the message. You don't know the internal processing about the message delivery.



Encapsulation in Java

- **Encapsulation in Java** is a *process of wrapping code and data together into a single unit*.
- Variables or data of a class is hidden from any other class and can be accessed only through any member function of own class in which they are declared.
- Example, a capsule which is mixed of several medicines.



Message Passing in Java

- Message passing in Java is like sending an object i.e. message from one thread to another thread.



Coupling and cohesion in Java

- **Coupling** refers to the interdependencies between modules, while **cohesion** describes how related the functions within a single module are.
- **coupling** is the degree of interdependence between software modules; a measure of how closely connected two routines or modules are; the strength of the relationships between modules.



Coupling and cohesion in Java

- Coupling is usually contrasted with cohesion. Low coupling often correlates with high cohesion, and vice versa.
- Low coupling is often a sign of a well-structured computer system and a good design, and when combined with high cohesion, supports the general goals of high readability and maintainability.



Getting Started with Java Programming

- A Simple Java Application
- Compiling Programs
- Executing Applications



A Simple Java Program

- A Simple Java Application
- Compiling Programs
- Executing Applications



A Simple Java Program

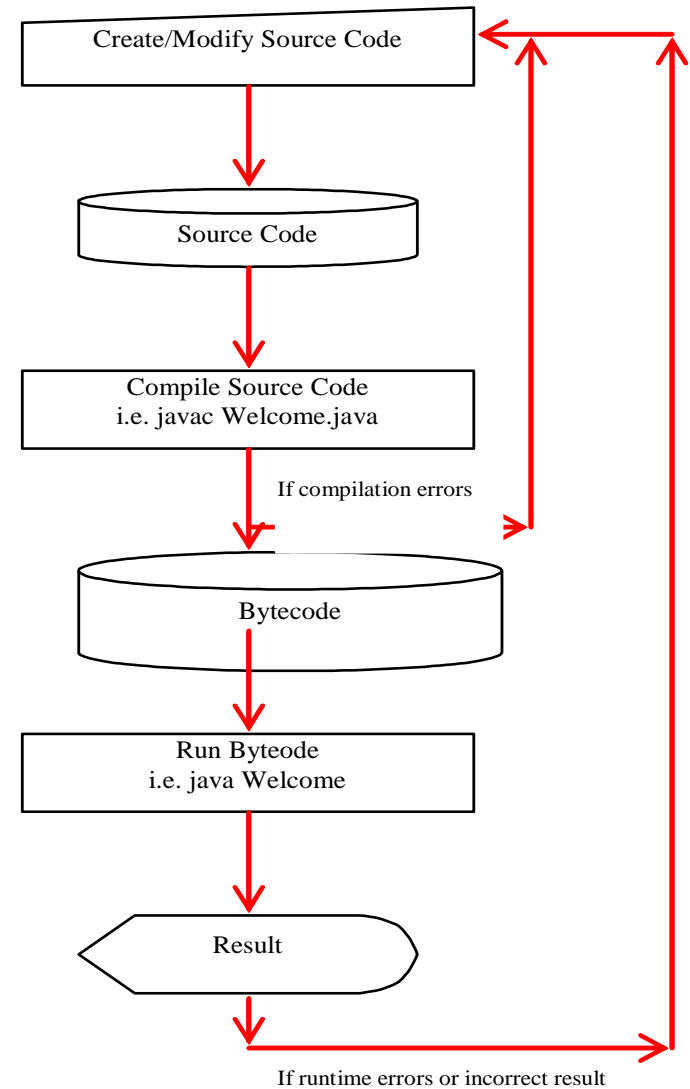
```
//This application program prints Welcome to  
Java!
```

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

Creating and Compiling Programs

■ On command line

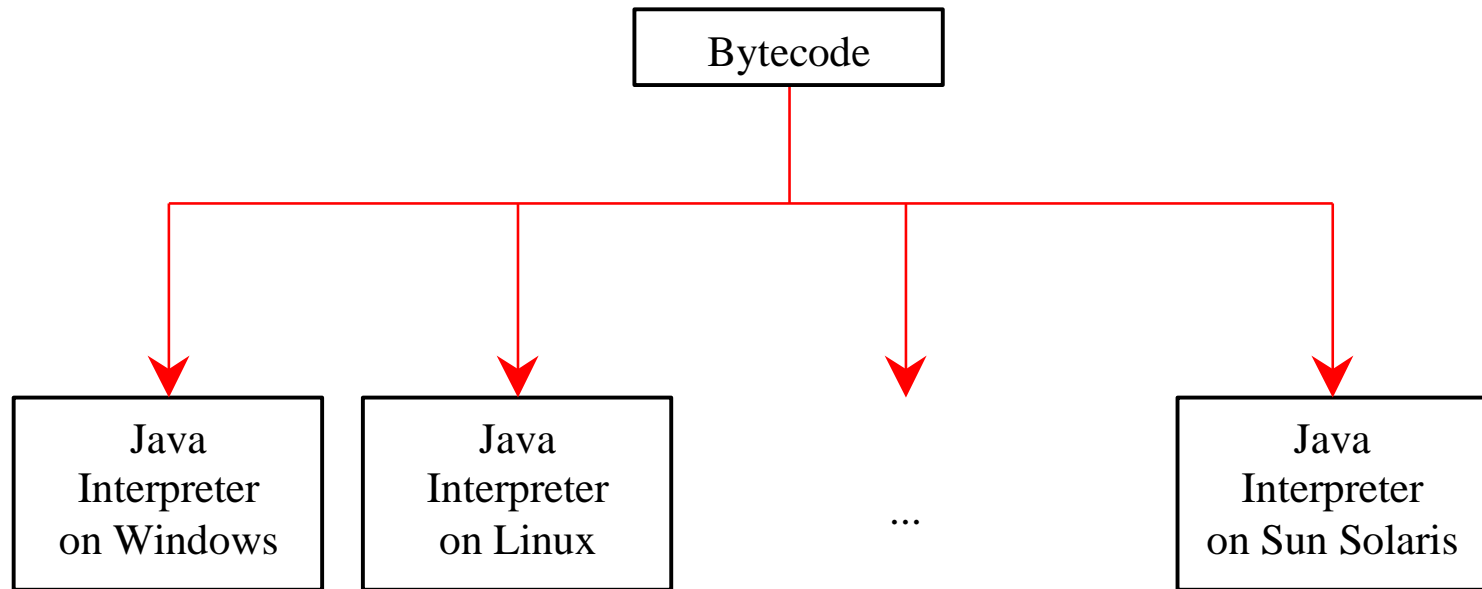
□ `javac file.java`



Executing Applications

■ On command line

□ `java classname`



How to install JAVA

- 1. Visit java.sun.com.
- 2. Find a link to download the Java SE 6 (the Java Standard Edition, version 6).
- If you find some other version. Just make sure that you have a version numbered 6 or higher.
- 3. Download the JDK, not the JRE.

To create your own Java programs, you need more than the JRE. You need the entire JDK.

- 4. Double-click the icon of the downloaded file and follow the wizard's instructions for installing the JDK.

Java Environment

JDK (Java Development kit)

JRE (Java Runtime env.)

JVM
(Java Virtual Machine)

+ library classes

+ development tools