### What is java?

Java is a high-level object oriented programming language and computing platform developed by Sun Microsystems and released in 1995. With more than 9 million developers worldwide, java enables us to develop and deploy applications and services.

From laptops to data centers, game consoles to super computers and cell phones to the Internet, java is everywhere.

### Why java?

**Due to platform independent and portability**

Develop software on one platform and deploy it on virtually any other platform. Create programs that can run in a web browser and access available web services.

Develop server-side applications like online stores, polls, image galleries, form processing and more. Create powerful applications for mobile phones and other embedded devices.

**Platform Independent: Java programs are witten once and run on any platform(OS)**

### Programming paradigm

**It is the way of using methodologies to solve a specific problem. why different paradigms?**

**To deal with complexity of the softwares.**

### Structured Paradigm Vs Object Oriented Paradigm

structured paradigm or procedural paradigm instructions are written in linear blocks or using functions.

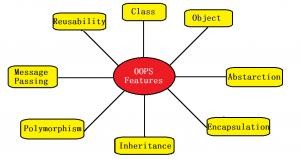
In procedural or structured paradigm, “**Code allows access to data”.**

In object oriented paradigm, the program is written using objects and classes. A problem can be solved

through interactions between objects. In object oriented paradigm, “**Data allows access to code”.**

### Object Oriented Paradigm:

It is a problem solving approach which modularize the wholeproblem into well defined and identifiable objects and the relationship among them.



### Object

Object is a real world thing or entity that is tangible or intangible. An object consists of properties/state and/or behavior. The set of values assigned to the properties of an object is known as the state of an object.

Every object, that takes some space in the world. Object can be physical or conceptual.

In ATM system

physical object: ATM,card,customer,bank conceptual: Transaction

For example consider a student object whose name is sam. Student object contains properties or characteristics like name, age, gender, date of birth etc and behavior like eat, sleep, register, write exam etc.

### Class

A class can be defined as a collection of similar objects.Similar means objects having same properties and behavior. class can be defined as a mold or template for creating objects. The state values that varies from one object to another object.

The class that does not take any space. It is an abstract thing.

### Messages

The objects communicate with one other by passing messages. A message is a request to carry out a task. A task can be anything from displaying a message to retrieving data from database and process it.

**Message passing** is calling the method of one object by another object.

### Abstraction

Abstraction as hiding the complex details and presenting only with the relevant or necessary details. Abstraction is used to manage complexity while writing programs.

Encapsulation and inheritance are used to implement abstraction while writing programs. Abstraction represents the external behavior of the object (what the object does) and encapsulation represents how it is implemented internally by the object.

**Data hiding** can be possible by the help of abstaction. Data hiding is required to make the data secure.

### Encapsulation

Encapsulation is the fundamental principle to achieve abstraction in object oriented programming. Abstraction is conceptual and encapsulation is a way to implement abstraction.

Encapsulation is defined as wrapping up or combining the data and code into a single unit. In the definition data refers to variables and code refers to methods (functions).

Every object contains private data (variables) and public methods which allows other code to access the private data.

### Inheritance

Another way to implement abstraction is by using inheritance.

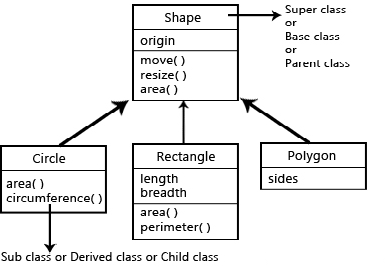
Inheritance is defined as one object derives or inherits the properties and behavior of another object.

In inheritance, classes are arranged as hierarchies.

common properties and behavior are moved into a common class also known as

super class or base class or parent class. Specific properties and behavior are moved into specialized classes also known as sub class or derived class or child class.

**Reusability and extensibility** is the advantages of inheritance.



### Polymorphism

Polymorphism means many forms.

It is defined as one interface multiple implementations. It means that based on the situation or different factors, the same thing exhibits multiple behaviors or the same thing can be used to carry out different tasks.

As a real world example for polymorphism, let’s consider the animal Chameleon. This animal changes its color to reflect the surroundings to hide from its prey. It is really cool! Based on the color of the surroundings the same thing (Chamelon) can change its color from one to another.

### Dynamic binding

**binding is the linking of function calling with function body.**

Binding of method calling to method body occures at run time. Method overriding is the concept of dynamic binding.

function declaration

function definition function calling

int add(int x,int y)

{

return x+y;

}

printf("%d",add(12,56)); //static binding ie occuring at compile time

### Persistency

permanently storing object in files or databases.

**Books to read** daniel liang complete reference black book

How platform Independent ?

Java Features

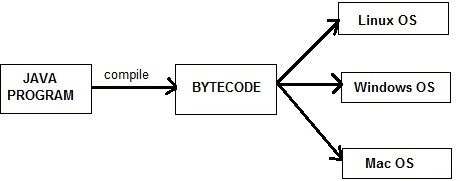
1. **Simple:** Java is very easy to write programs using Java’s syntax for the

experienced programmers who know C and C++ . Java’s syntax was derived from C and object orientation features were derived from C++.

1. **Object Oriented**: Java is a object oriented programming language which allowsthe users to write a program in terms of classes and objects. All the object oriented features can be implemented in java. For every primitive types there are wrapper class like Integer,Boolean,Chatacter,Long,Float,Byte,String
2. **Platform Independent:** Same java program can be run any platform or operating system. The java source code is compiled into an intermediate code known as **bytecode,** which is not human readable. The bytecode can be run on any java compatible platform.

Java comapatible means Java virtual machine is there.

The bytecode makes java program platform independent with compiler point of view, ie there is a single java comiler for every platform.



1. **Architecture Neutral**: Java compiler generates bytecode which when run on any machine having any architecture gives the same output.For example, Java’s integer primitive type always means a signed 2’s complement 32-bit integer. Whereas in C/C++, integer can be unsigned and its size varies according to the system’s register size. Java’s goal is to enable programmers to develop programs which has WORA (Write Once, Run Anywhere) property.
2. **Portable**: Portability is achieved through Java’s architecture neutrality and

platform independent features.We can carry the bytecode and run on any java

compatible machine.

1. **Both compiled and Interpreted:** Java programs are compiled into bytecode.The bytecode generated by the Java compiler goes as input to the Java Virtual Machine (JVM) which is the runtime engine that interprets bytecode into machine code (0’s and 1’s) line by line. Interpretation simplifies debugging process.
2. **Robust:** Errors in Java programs does not crash the virtual machine or hurt the underlying operating system. Several built-in Java features help the users to create robust programs.

if(factor==2) printf("not prime); else if(factor>2) printf("prime")l

int a=8;int b=0; int c=a/b; printf("%d",c);

Java is a strongly typed language which requires the programmer to declare the type of a variable before using it. Java also has built-in memory management mechanism known as garbage collector.

Java also provides exception handling to handle runtime errors. Pointers available in C/C++ are not supported by Java which may lead to memory errors.

1. **Secure**: Bytecodes are not readable by human being, those are symbolic instruction. In JVM bytecode verifier program verifies the correctness of bytecode and then execute the program.

Java’s sandbox security model available in applets prevents the Java program

from accessing the resources of the user’s system. This prevents malicious

programs to gain access to sensitive information available on the user’s system. The memory management is completely done by jvm. No pointer is used. garbage collector is there to deallocate memory space by any object.

### Multithreaded :

Thread is **light weight** process which requires less resource(**memory)** for execution.

process **is heavy weight**.

Multithreading enables a program to carry out multiple tasks at the same time. For example, consider listening to music and writing something on a piece of paper at the same time.

Java provides multiprocess synchronization primitives so that a programmer can be free from the worries of providing complex synchronization solutions.

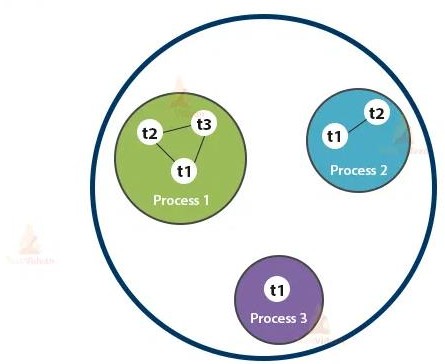
### Advantages:

The main advantage of multithreading is that the maximum utilization of resources is possible.

It doesn’t occupy memory for each thread. It shares a common memory area.

There is no need to wait for the application to finish one task before beginning another one.

There is a decreased cost of maintenance. Also, It is time-saving. It improves the performance of complex applications.



1. **High performance**: The interpreter makes execution of the program slower.Many virtual machines use a **JIT (Just-In-Time)** compiler that converts the bytecode into platform-specific instructions once as the program executes , which are for repitative task. It compiles only that method which is being called. This saves time and makes it more efficient. This helps to improve the performance of the Java programs.

Multithreading enhances the overall execution speed of Java programs.

1. **Distributed**: Java provides libraries for handling TCP/IP, FTP and other network protocols which enables programmers to easily develop distributed applications. Java also provides Remote Method Invocation (RMI) which enables a program to access methods across a network.
2. **Dynamic**: Java programs uses run-time type information which is used to verify and access objects at run time. This allows to dynamically link small fragments of bytecode during execution of the application.

### History

**James Gosling, Mike Sheridan, and Patrick Naughton** developed java.

Initially it was designed for electronic device by sun microsystems named as Oak. Then after it is named as Java in 1995. Now it is a product of Oracle corporation.

The first version 1.0 of Java came in 1996 when Sun Microsystems promised the principle of WORA (Write Once, Run Anywhere) property.

Then Java 2 (J2SE 1.2) was introduced in December 1998-1999. J2EE(Java Enterprise Editions) was for enterprise applications.

In 2006, Sun renamed new J2 versions as Java EE, Java ME, and Java SE. September 2018 marked the release of Java SE 11 (LTS).

March 2019 marked the release of Java SE 12 (LTS). On September 10th, 2019, Java SE 13 was released.

Java SE 14 came in March 2020, which is the latest version of Java.

### Edition of Java

Depending on type of application there are different edition of java available.

1. Java SE: Standard Edition

It is used to develop desktop application

1. Java EE : Enterprise Edition

It is used to develop enterprise edition.

1. Java ME : Micro Edition

It is used to develop mobile application

API : application programming interface set of packages

package: collection of interrelated classes and interface

program=data structure+logic/algorithm

### HOW TO MEASURE THE EFFICIENCY OF A PL

**to reduce development time**

### to reduce execution time

**which takes less resource for execution**

### softwares are inherently complex. flexibility more

**different kinds of users : abstraction is more**

**Difference between Structure oriented and Object Oriented Paradigm**

**structure oriented pl**

1. more emphasis is given to the algorith ie how to solve a problem
2. data moves freely from one function to another function(global sata) 3.reusability is less

4. extensibility is less object oriented paradigm

1. more emphasis is given to data rather than alg. what to do
2. data is hidden in the form of object
3. reusability is more 4.extensibility is more

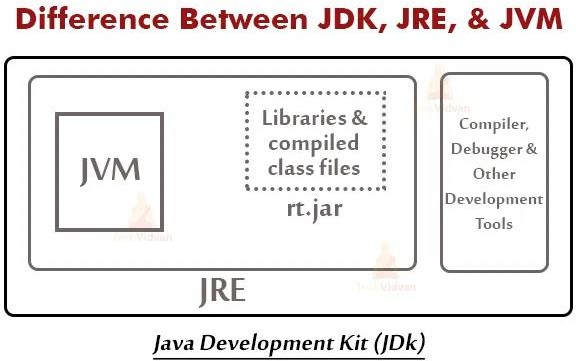
5.real world sw can be easily developed

### Software tools required to run java

**JRE or** java runtime Environment is the main sw required to execute the bytecode to its corresponding native code by the jvm present in it.

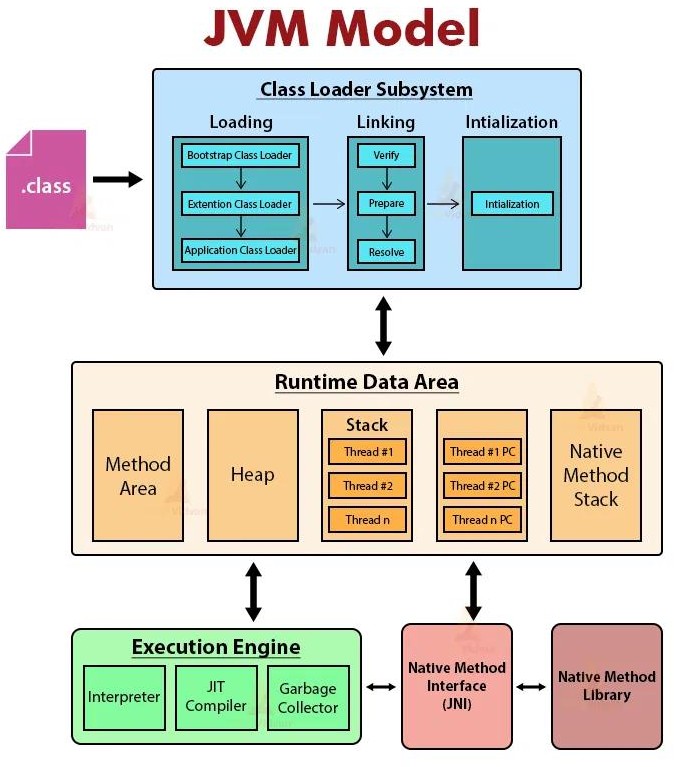
**JDK** is for the development environment whereas JRE is for the run time environment.Jdk is the combination of JRE and development tools like compiler.

### Difference Between JDK, JRE, And JVM



**JVM Architecture**

Java virtual machine is an abstract machine which takes bytecode as input and generates native code by its execution engine known as interpreter.



There are three main subsystems in JVM Architecture:

1. ClassLoader
2. Memory Area
3. Execution Engine

It is the component of JVM Architecture **that loads classes in memory.** Every JVM consists of a ClassLoader. Three built-in classloaders in Java are:

### Bootstrap ClassLoader

This is the classloader which is the super class of Extension classloader. It loads the rt.jar file.

### Extension ClassLoader

It is the ClassLoader which loads the jar files present in the directory. This is the child Bootstrap classLoader and parent of System classloader.

### Application ClassLoader

It is the classLoader which loads the class files from the classpath. This is the child of the Extension classloader.

The Three important functions of ClassLoader are Initialization, Loading, Linking.

### Initialization

This operation involves the assignment of all static variables with their specific values in the block of the program.

In this phase, there is an assignment of all static variables with their values defined in the code and static block.

### Loading

This operation loads files from secondary memory into the main memory (RAM) for execution. The Classloader reads the .class file, generates the corresponding binary data, and saves it in the method area.

JVM stores some information for each .class file in the method area. This information is:

The fully qualified name of the loaded class and its immediate parent class.

Whether .class file is related to interface or an enum or a class. Modifier, Variables and Method information, etc.

After loading the .class file, JVM creates an object of type Class to represent this file in the heap memory.

The programmers can use this Class object for getting class level information like the name of the class, parent name, methods, and variable information, etc.

To get this object reference we can use the getClass() method of Object class.

### Linking

This operation combines different files in the main program together. It performs verification, preparation, and (optionally) resolution.

**Verification:** The Verification phase checks the correctness of the .class file. It means that it checks that the file formation and generation is by a valid compiler or not. If the verification fails then we get a java.lang.Verify Exception.

**Preparation:** JVM allocates memory for class variables and initializes the memory to default values.

**Resolution:** Resolution is the process of replacing symbolic references with direct references. It uses searching into the method area to locate the referenced entity.

final double PI=3.141;

### JVM Memory area

1. Method Area– It stores the structure of each class like method data, field data, runtime pool, metadata.
2. Heap– Heap is the runtime area where object allocation takes place.
3. Stacks– Stacks store the partial results and local variables of a program. Whenever a thread is created, there is a simultaneous creation of JVM stack. When we invoke a method, a new frame creates and destroys at the same time

when the invocation process completes.

1. PC Registers – It stores the address of the currently executing JVM instruction.
2. Native Method Stacks – It includes all the native methods required in any application. It is not written in java.

### Execution Engine in Java

It is the component of JVM that reads data from memory locations and executes the instructions. It has three major components namely a virtual processor, an interpreter, and a JIT compiler.

1. Virtual Processor
2. Interpreter: Read the bytecode stream then execute the instructions.
3. Just-In-Time(JIT) compiler: It improves performance. JIT compiles parts of the byte code with similar functionality at the same time and reduces the amount of time needed for compilation.
4. Garbage collector: It is deamon thread or low priority thread which always runs behind the system to check whether there is any object not to be used.If found,then its memory is deallocated to the heap area.

It has two methods **mark() and sweep() Native Method interface**

It is a framework that helps in communication between different applications written in different languages such as c, c++, etc.

Native Method Interface allows Java code running in a JVM to call by libraries and native applications.

**How to write a simple java program:**java editors are needed to write a Java program.

Notepad

Netbeans //frameworks Eclipse