Introduction

What is Java ?

Java is a programming language and a platform.

Java is a high level, robust, secured and object-oriented programming language.

Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has its own runtime environment (JRE) and API, it is called platform.

Where it is used ?

According to Sun, 3 billion devices run java. There are many devices where Java is currently used. Some of them are as follows:

1. Desktop Applications such as acrobat reader, media player, antivirus etc.
2. Web Applications such as irctc.co.in, javatpoint.com etc.
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System
6. Smart Card
7. Robotics
8. Games etc.

Types of Java Applications :

There are mainly 4 types of applications that can be created using java programming:

1) **Standalone Application**

Standalone applications are also known as desktop applications or window-based applications. These are traditional software that we need to install on every machine.Example of standalone applications are: Media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.

2) **Web Application**

An application that runs on the server side and creates dynamic page, is called web application.Currently, Servlet, JSP, Struts, Spring, Hibernate, JSF etc. technologies are used for creating web applications in java.

Java Basic

3) **Enterprise Application**

An application that is distributed in nature, such as banking applications etc. is called enterprise application. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.

4) **Mobile Application**

An application that is created for mobile devices. Currently Android and Java ME are used for creating mobile applications.

Java Platforms / Editions

There are 4 platforms or editions of Java:

1) **Java SE (Java Standard Edition)**

It is a java programming platform. It includes Java programming APIs(application programming interface) such as java.lang, java.io, java.net, java.util, java.sql, java.math etc. It includes core topics like OOPs, String, Regex, Exception, Inner classes, Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection etc.

Java Basic

2) **Java EE (Java Enterprise Edition)**

It is an enterprise platform which is mainly used to develop web and enterprise applications. It is built on the top of Java SE platform. It includes topics like Servlet, JSP, Web Services, EJB, JPA etc.

3) **Java ME (Java Micro Edition)**

It is a micro platform which is mainly used to develop mobile applications.

Java Basic

4) **JavaFx**

It is used to develop rich internet applications. It uses light-weight user interface API.

History of Java

The history of Java is very interesting. Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at the time. The history of java starts from Green Team. Java team members (also known as Green Team), initiated this project to develop a language for digital devices such as set-top boxes, televisions etc. But, it was suited for internet programming. Later, Java technology was incorporated by Netscape.

The principles for creating Java programming were "Simple, Robust, Portable, Platform-independent, Secured, High Performance, Multithreaded, Architecture Neutral, Object-Oriented, Interpreted and Dynamic".

History of Java contd.....

Currently, Java is used in internet programming, mobile devices, games, e-business solutions etc. There are given the major points that describes the history of java.

Java History from Oak to Java

1) James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. The small team of sun engineers called Green Team.

2) Originally designed for small, embedded systems in electronic appliances like set-top boxes.

3) Firstly, it was called "Greentalk" by James Gosling and file extension was .gt.

4) After that, it was called Oak and was developed as a part of the Green project.

History of Java contd.....

Why Java named as "Oak" ?

Oak is a symbol of strength and choosen as a national tree of many countries like U.S.A., France, Germany, Romania etc.

In 1995, Oak was renamed as "Java" because it was already a trademark by Oak Technologies.

History of Java contd.....

Why Java Programming named as "Java" ?

Why had they choosen java name for java language? The team gathered to choose a new name. The suggested words were "dynamic", "revolutionary", "Silk", "jolt", "DNA" etc. They wanted something that reflected the essence of the technology: revolutionary, dynamic, lively, cool, unique, and easy to spell and fun to say.

According to James Gosling "Java was one of the top choices along with Silk". Since java was so unique, most of the team members preferred java.

History of Java contd.....

Java is an island of Indonesia where first coffee was produced (called java coffee).

Notice that Java is just a name not an acronym.

Originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995.

In 1995, Time magazine called Java one of the Ten Best Products of 1995.

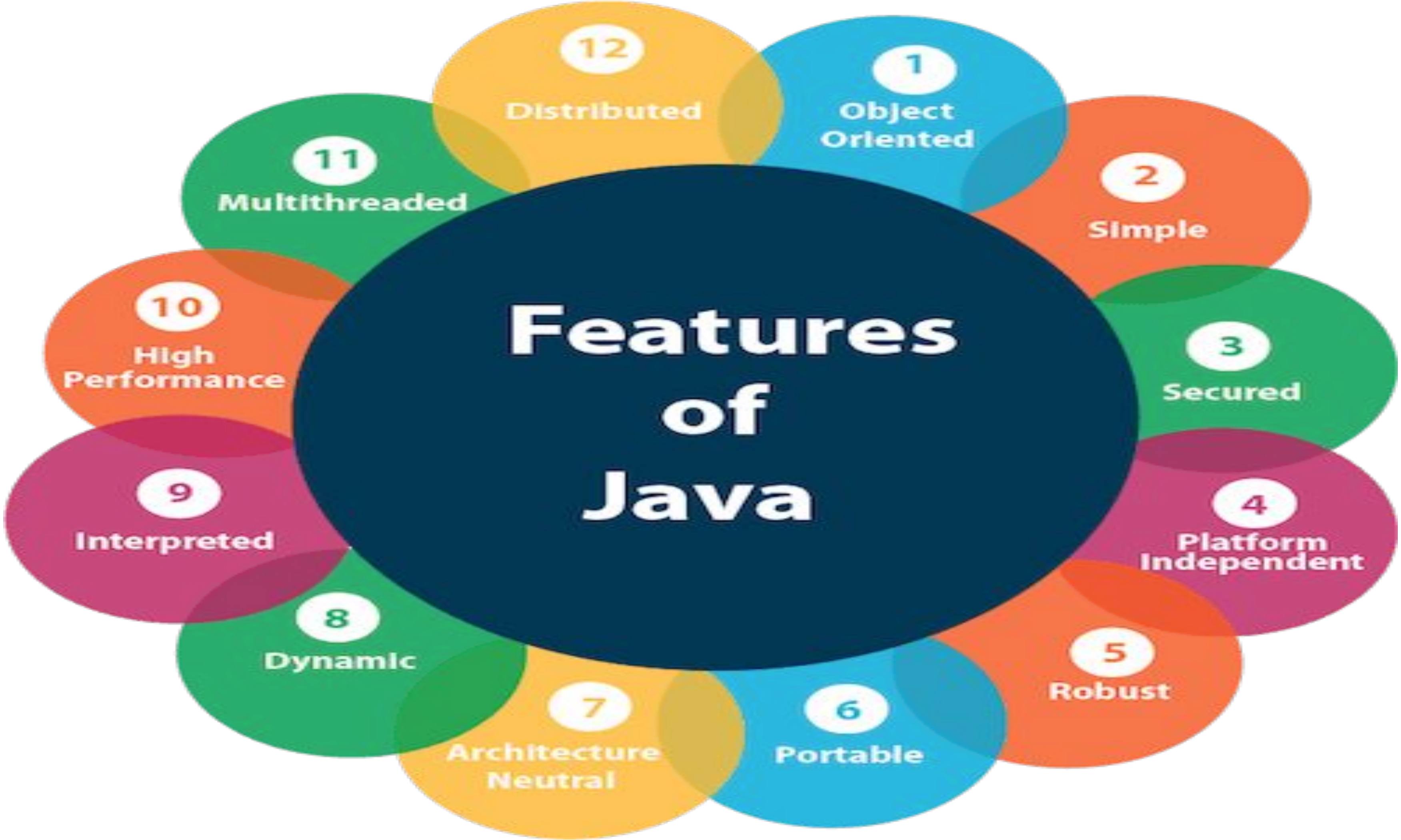
JDK 1.0 released in(January 23, 1996).

Features of Java

Features of Java

The main objective of Java programming language creation was to make it portable, simple and secure programming language. Apart from this, there are also some awesome features which play important role in the popularity of this language. The features of Java are also known as java buzzwords.

Features of Java contd.....



Features of Java contd.....

1. Object-oriented

Java is 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.

Features of Java contd.....

Basic concepts of OOPs are:

Object
Class
Inheritance
Polymorphism
Abstraction
Encapsulation

Features of Java contd.....

2. 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++).

Java has removed many confusing and rarely-used features e.g. explicit pointers, operator overloading etc.

There is no need to remove unreferenced objects because there is Automatic Garbage Collection in java.

Features of Java contd.....

3. Secured

Java is best known for its security. With Java, we can develop virus-free systems. Java is secured because:

i) **No explicit pointer**
ii) **Java Programs run inside virtual machine sandbox**

iii) **Classloader**: Classloader in Java is a part of the Java Runtime Environment(JRE) which is used to dynamically load Java classes into the Java Virtual Machine. It adds security by separating the package for the classes of the local file system from those that are imported from network sources.

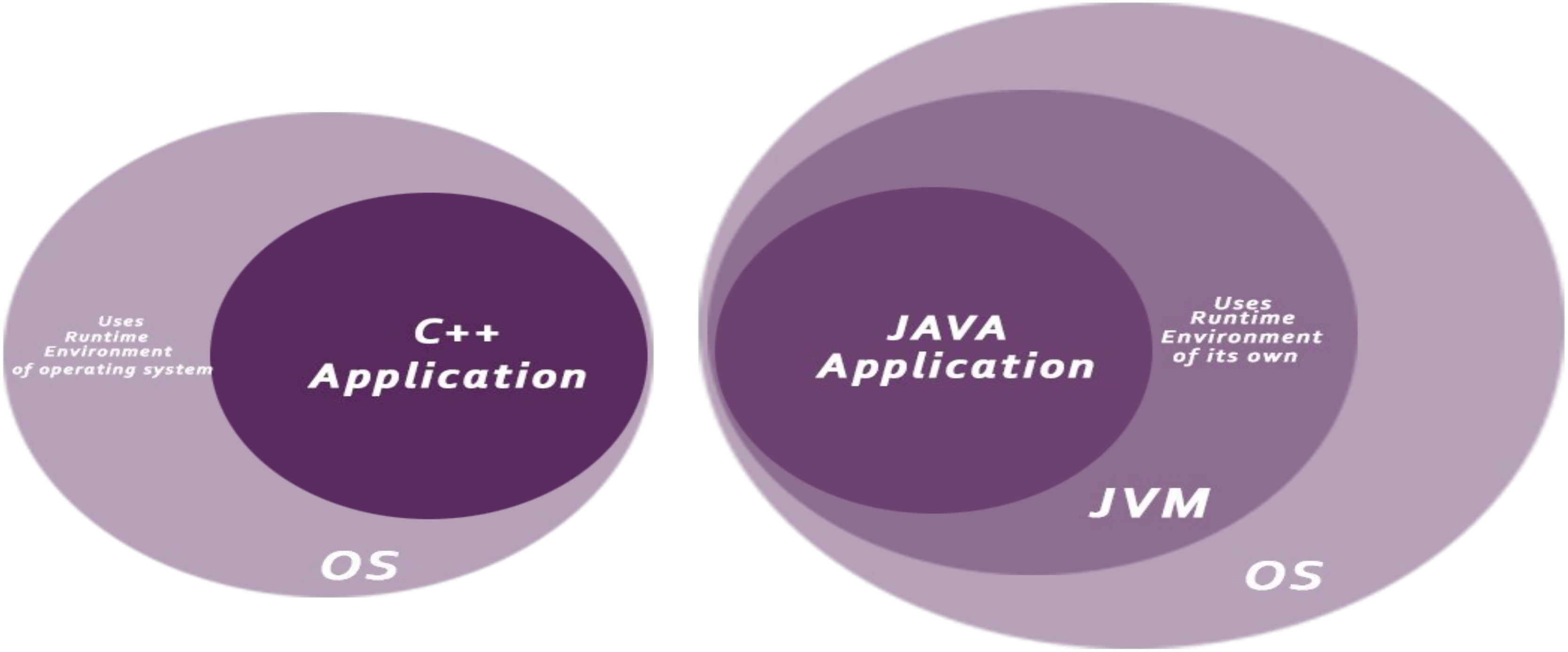
Features of Java contd.....

iv) Bytecode Verifier : It checks the code fragments for illegal code that can violate access right to objects.

v) Security Manager : It determines what resources a class can access such as reading and writing to the local disk.

These security are provided by java language. Some security can also be provided by application developer through SSL, JAAS, Cryptography etc.

Features of Java contd.....



Features of Java contd.....

4. 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. A platform is the hardware or software environment in which a program runs.

There are two types of platforms software-based and hardware-based. Java provides software-based platform.

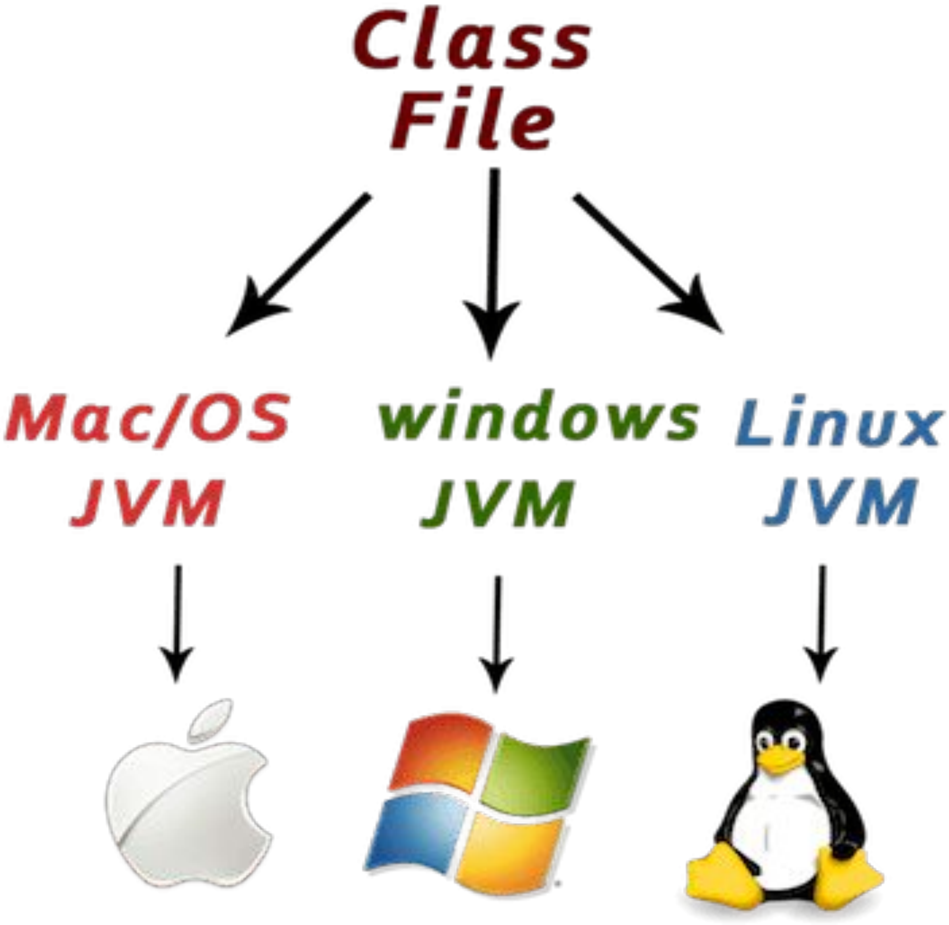
The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on the top of other hardware-based platforms. It has two components:

Features of Java contd.....

a. Runtime Environment
b. API(Application Programming Interface)

Java code can be run on multiple platforms e.g. Windows, Linux, Sun Solaris, Mac/OS etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere(WORA).

Features of Java contd.....



Features of Java contd.....

5. Robust

Robust simply means strong. Java is robust because:

a. It uses strong memory management.

b. There are lack of pointers that avoids security problems.

c. There is automatic garbage collection in java which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.

d. There is exception handling and type checking mechanism in java. All these points makes java robust.

Features of Java contd.....

6. Portable

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

7. Architecture-neutral

Java is architecture neutral because there is no implementation dependent features e.g. size of primitive types is fixed.

In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. But in java, it occupies 4 bytes of memory for both 32 and 64 bit architectures.

Features of Java contd.....

8. 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).

Features of Java contd.....

10. 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++). Java is an interpreted language that is why it is slower than compiled languages e.g. C, C++ etc.

11. Multi-threaded

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.

Features of Java contd.....

12. Distributed

Java is distributed because it facilitates users to create distributed applications in java. RMI and EJB are used for creating distributed applications. This feature of Java makes us able to access files by calling the methods from any machine on the internet.

C++ vs Java

C++ vs Java

There are many differences and similarities between C++ programming language and Java. A list of top differences between C++ and Java are given below:

C++ vs Java contd.....

Comparison Index	C++	Java
Platform-independent	C++ is platform-dependent.	Java is platform-independent.
Mainly used for	C++ is mainly used for system programming.	Java is mainly used for application programming. It is widely used in window, web-based, enterprise and mobile applications.

C++ vs Java contd…..

Comparison Index	C++	Java
Design Goal	C++ was designed for systems and applications programming. It was an extension of C programming language.	Java was designed and created as an interpreter for printing systems but later extended as a support network computing. It was designed with a goal of being easy to use and accessible to a wider audience.
Goto	C++ supports goto statement.	Java doesn't support goto statement.

C++ vs Java contd.....

Comparison Index	C++	Java
Multiple inheritance	C++ supports multiple inheritance.	Java doesn't support multiple inheritance through class. It can be achieved by interfaces in java.
Operator Overloading	C++ supports operator overloading.	Java doesn't support operator overloading.
Pointers	C++ supports pointers. You can write pointer program in C++.	Java supports pointer internally. But you can't write the pointer program in java. It means java has restricted pointer support in java.

C++ vs Java contd.....

Comparison Index	C++	Java
Compiler and Interpreter	C++ uses compiler only. C++ is compiled and run using compiler which converts source code into machine code so, C++ is platform dependent.	Java uses compiler and interpreter both. Java source code is converted into byte code at compilation time. The interpreter executes this byte code at run time and produces output. Java is interpreted that is why it is platform independent.
Call by Value and Call by reference	C++ supports both call by value and call by reference.	Java supports call by value only. There is no call by reference in java.
Structure and Union	C++ supports structures and unions.	Java doesn't support structures and unions.

Java Example

```
public class Simple
{
public static void main(String args[])
    {
      System.out.println("hello world");
    }
}
```

Java Example

Parameters used in first java program

Let's see what is the meaning of class, public, static, void, main, String[], System.out.println().

- **class** keyword is used to declare a class in java.
- **public** keyword is an access modifier which represents visibility, it means it is visible to all.

- **static** is a keyword, if we declare any method as static, it is known as static method. The core advantage of static method is that there is no need to create object to invoke the static method. The main method is executed by the JVM, so it doesn't require to create object to invoke the main method. So it saves memory.

Java Example

- **void** is the return type of the method, it means it doesn't return any value.

- **main** represents the starting point of the program.

- **String[]** args is used for command line argument. We will learn it later.

- **System.out.println()** is used print statement.

We will learn about the internal working of System.out.println statement later.

Basic Java

Difference between JDK, JRE and JVM

Firstly, let's see the basic differences between the JDK, JRE and JVM.

JVM (Java Virtual Machine) is an abstract machine. It is called virtual machine because it doesn't physically exist. It is a specification that provides runtime environment in which java bytecode can be executed. It can also run those programs which are written in other languages and compiled to Java bytecode.

Basic Java

JVMs are available for many hardware and software platforms. JVM, JRE and JDK are platform dependent because configuration of each OS are different from each other. But, Java is platform independent. There are three notions of the JVM: specification, implementation, and instance.

The JVM performs following main tasks:

Loads code
Verifies code
Executes code
Provides runtime environment

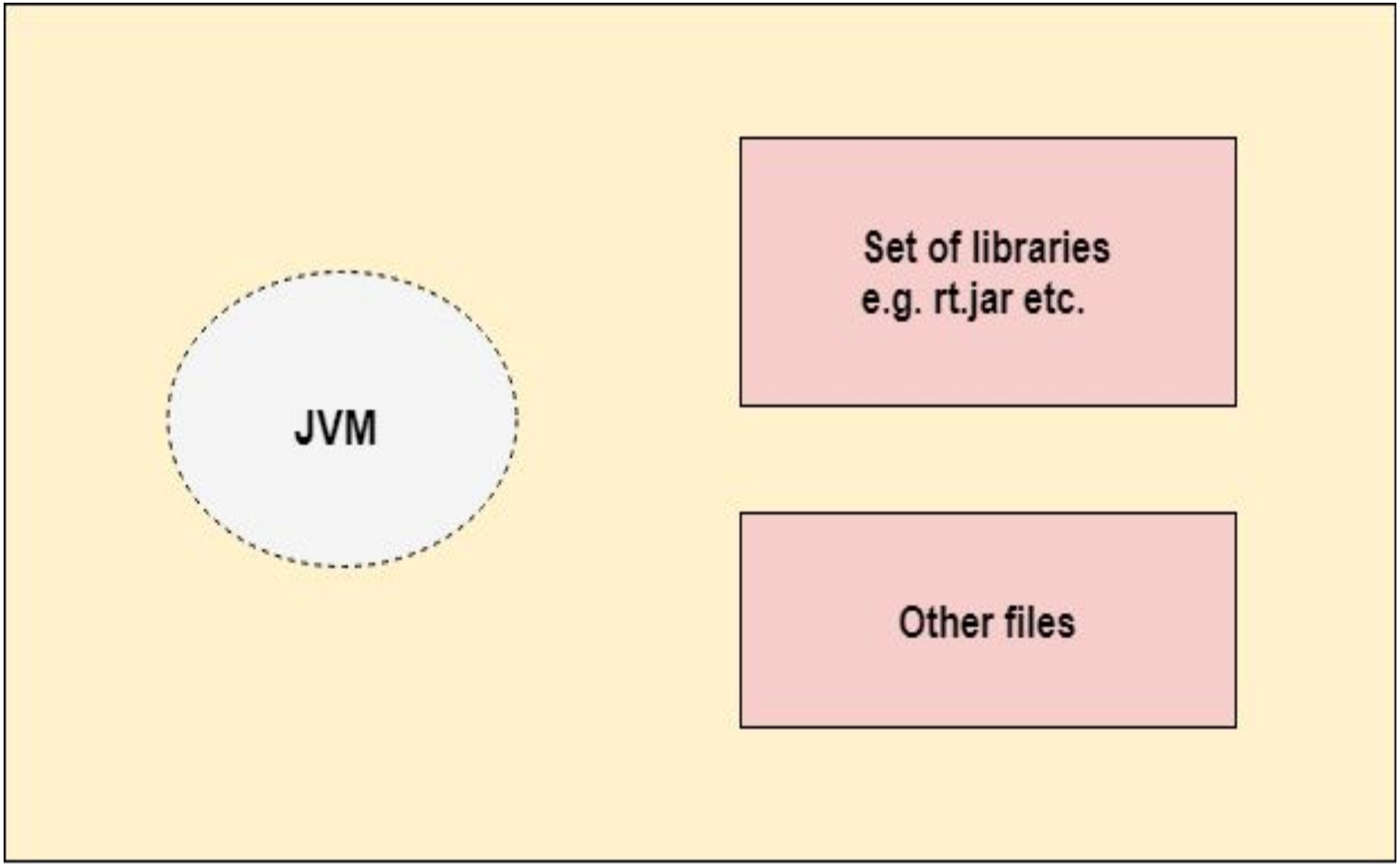
Basic Java

JRE

JRE is an acronym for Java Runtime Environment. It is also written as Java RTE. The Java Runtime Environment is a set of software tools which are used for developing java applications. It is used to provide runtime environment. It is the implementation of JVM. It physically exists. It contains set of libraries + other files that JVM uses at runtime.

Implementation of JVMs are also actively released by other companies besides Sun Micro Systems.

Basic Java



JRE

Basic Java

JDK

JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop java applications and applets. It physically exists. It contains JRE + development tools.

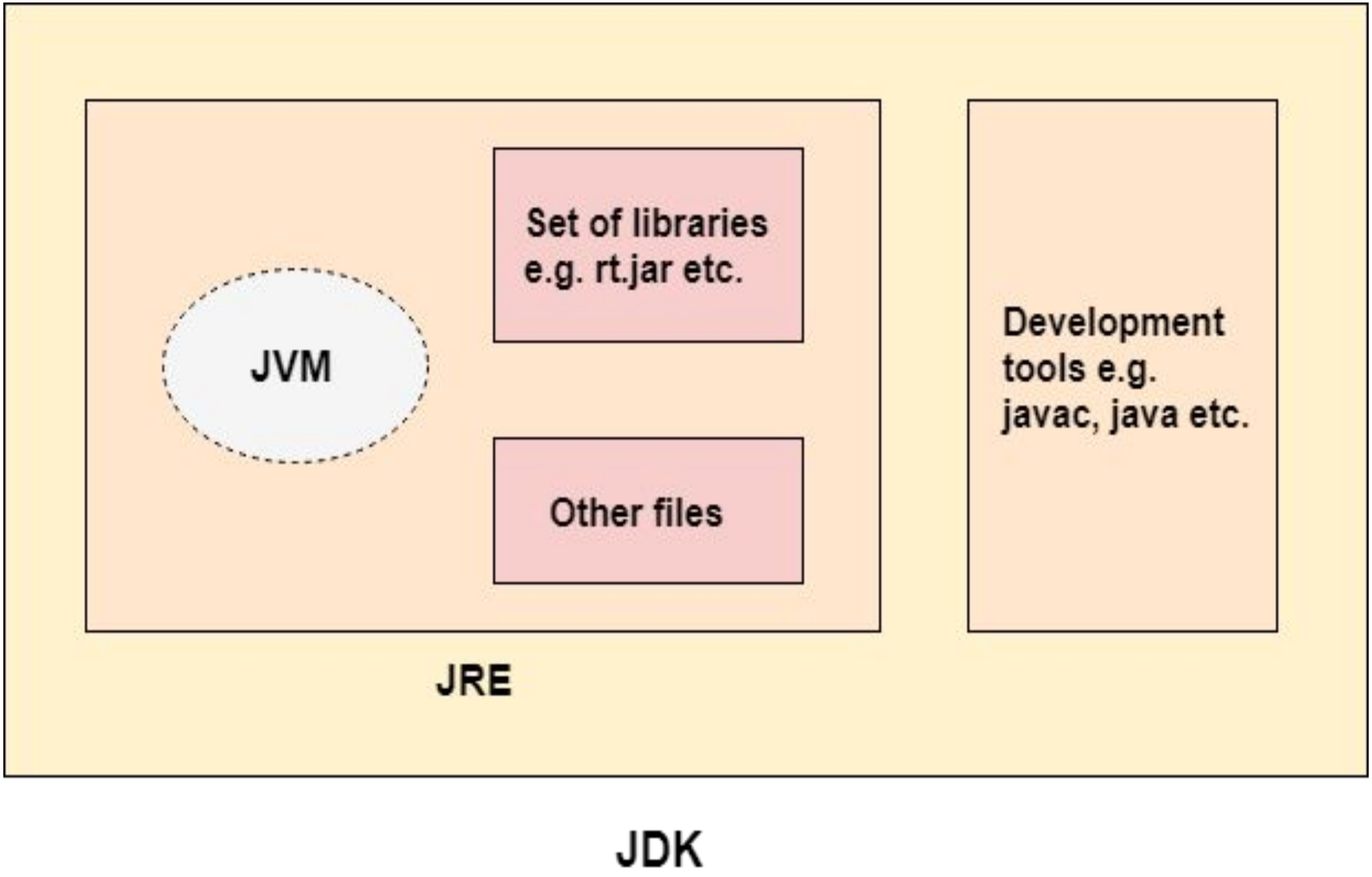
JDK is an implementation of any one of the below given Java Platforms released by Oracle corporation:

Standard Edition Java Platform
Enterprise Edition Java Platform
Micro Edition Java Platform

Basic Java

The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) etc. to complete the development of a Java Application.

Basic Java



Set of libraries e.g. rt.jar etc.

JVM

Development tools e.g. javac, java etc.

Other files

JRE

JDK