SE Computer- Division 3

Course Name :Principles of Programming Language

Course Code: 210256

Course InCharge: Mrs. Alfiya Shahbad

Unit 3

Java as Object Oriented Programming Language-Overview

Fundamentals of JAVA, Arrays: one dimensional array, multi-dimensional array, alternative array declaration statements ,String Handling: String class methods

Classes and Methods: class fundamentals, declaring objects, assigning object reference variables, adding methods to a class, returning a value, constructors, this keyword, garbage collection, finalize() method,

overloading methods, argument passing, object as parameter, returning objects, access control, static, final, nested and inner classes, command line arguments, variable -length arguments.

Topic	Book To Refer
Fundamentals of JAVA, Arrays: one dimensional array, multi-dimensional array, alternative array declaration statements ,String Handling: String class methods Classes and Methods: class fundamentals, declaring objects, assigning object reference variables, adding methods to a class, returning a value, constructors, this keyword, garbage collection, finalize() method, overloading methods, argument passing, object as parameter, returning objects, access control, static, final, nested and inner classes, command line arguments, variable -length arguments.	**Herbert Schildt, "The Complete Reference Java", 9th Ed, TMH,ISBN: 978-0-07-180856-9.** **Programming With Java, 3rd Edition, E. Balaguruswamy**

Topic	Book To Refer
String Handling: String class methods Classes and Methods: class fundamentals, declaring objects, assigning object reference variables, adding methods to a class, returning a value, constructors, this keyword, garbage collection, finalize() method, overloading methods, argument passing, object as parameter, returning objects, access control, static, final, nested and inner classes, command line arguments, variable -length arguments.	**Herbert Schildt, "The Complete Reference Java", 9th Ed, TMH,ISBN: 978-0-07-180856-9.**

Topic	Book To Refer
String Handling: String class methods	**Herbert Schildt, "The Complete Reference Java", 9th Ed, TMH,ISBN: 978-0-07-180856-9.** **Page No- 413-431**

What is Java ?

What is Java?

Java is a programming language and a platform. Java is a **high level, robust, object-oriented and secure programming language.**

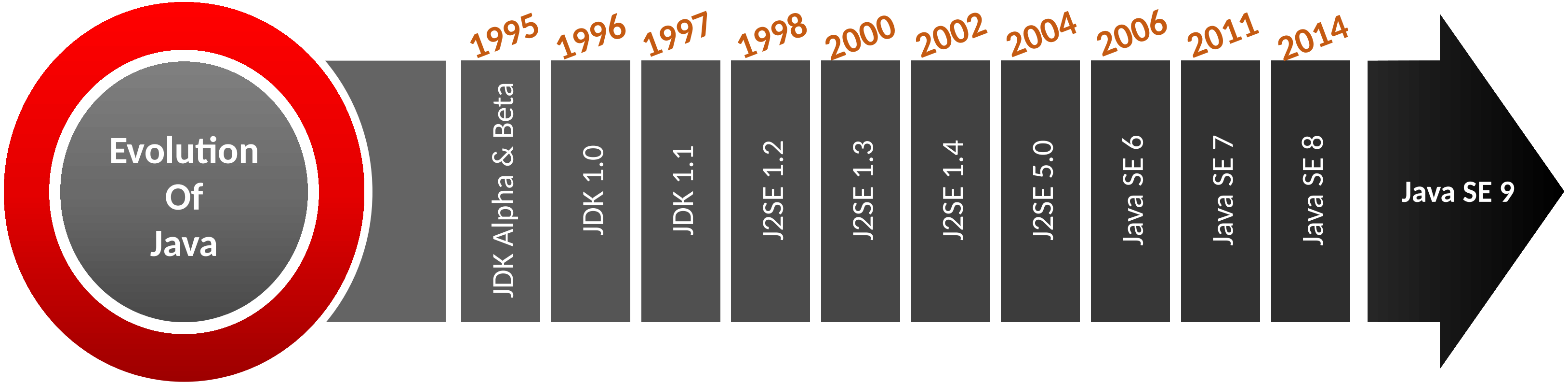
A Brief History of Java Programming Language

History of Java

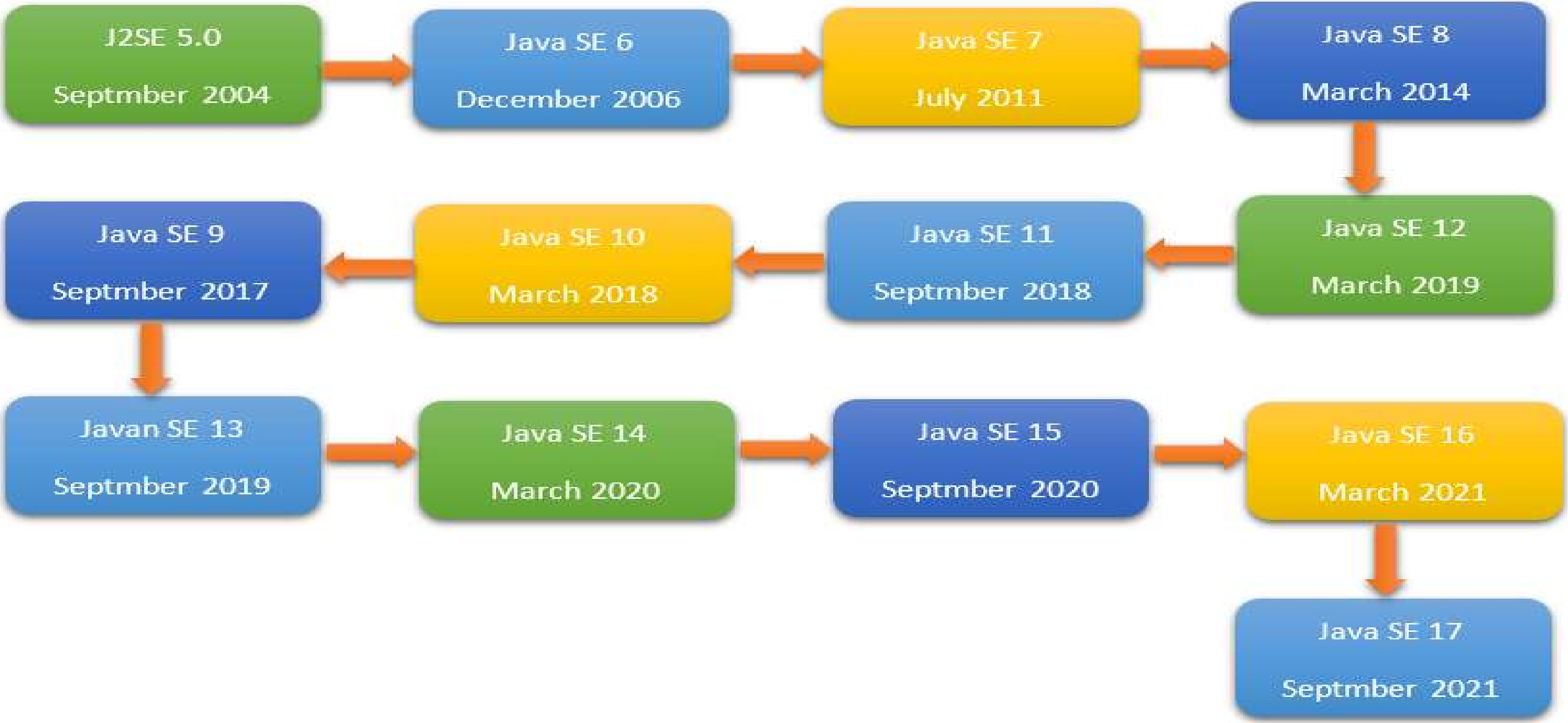
Java was developed by **Sun Microsystems (which is now the subsidiary of Oracle**) in the year 1995. **James Gosling is known as the father of Ja**va. Before Java, its name was **Oak**. Since Oak was already a registered company, **so James Gosling and his team changed the Oak name to Java.**

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

Java Version History



Java Version History



J2SE 5.0

Septmber 2004

Java SE 6

December 2006

Java SE 7

July 2011

Java SE 8

March 2014

Java SE 12

March 2019

Java SE 11

Septmber 2018

Java SE 10

March 2018

Java SE 9

Septmber 2017

Javan SE 13

Septmber 2019

Java SE 14

March 2020

Java SE 15

Septmber 2020

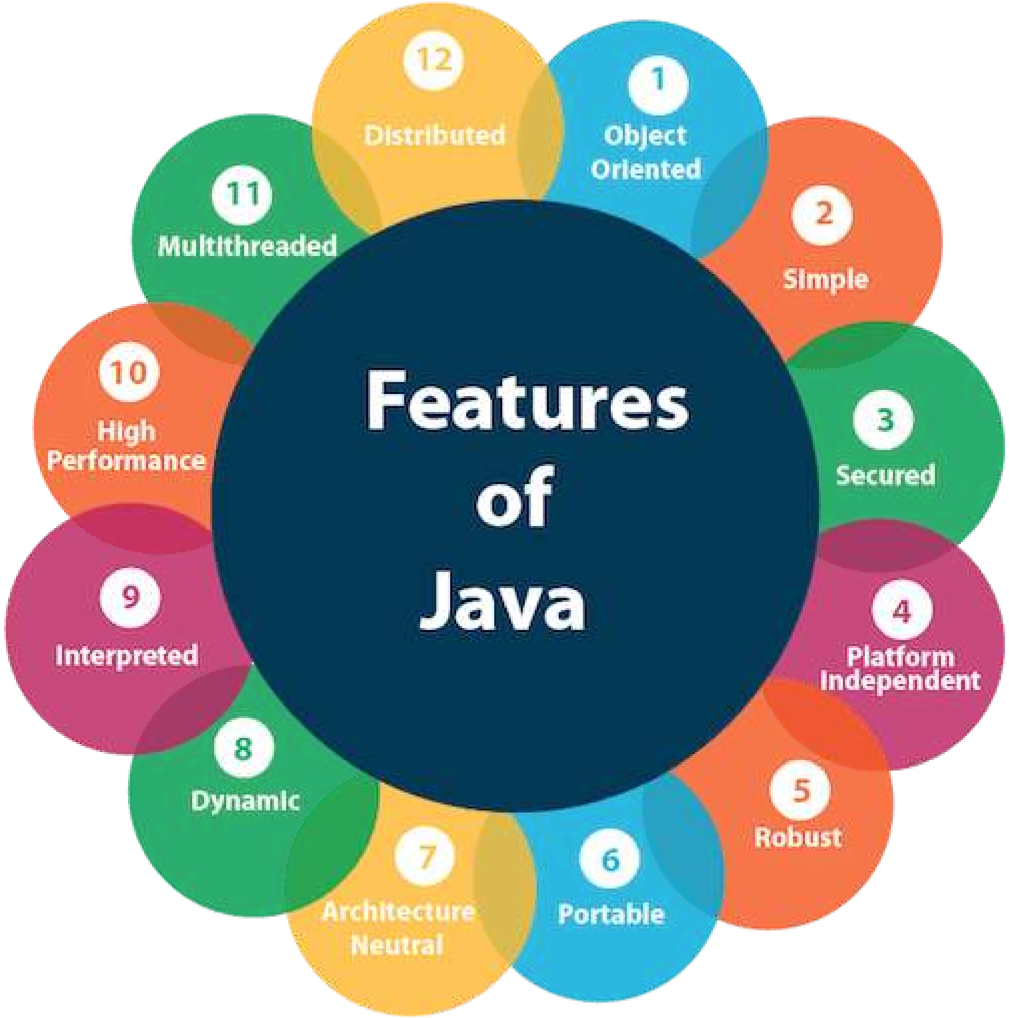
Java SE 16

March 2021

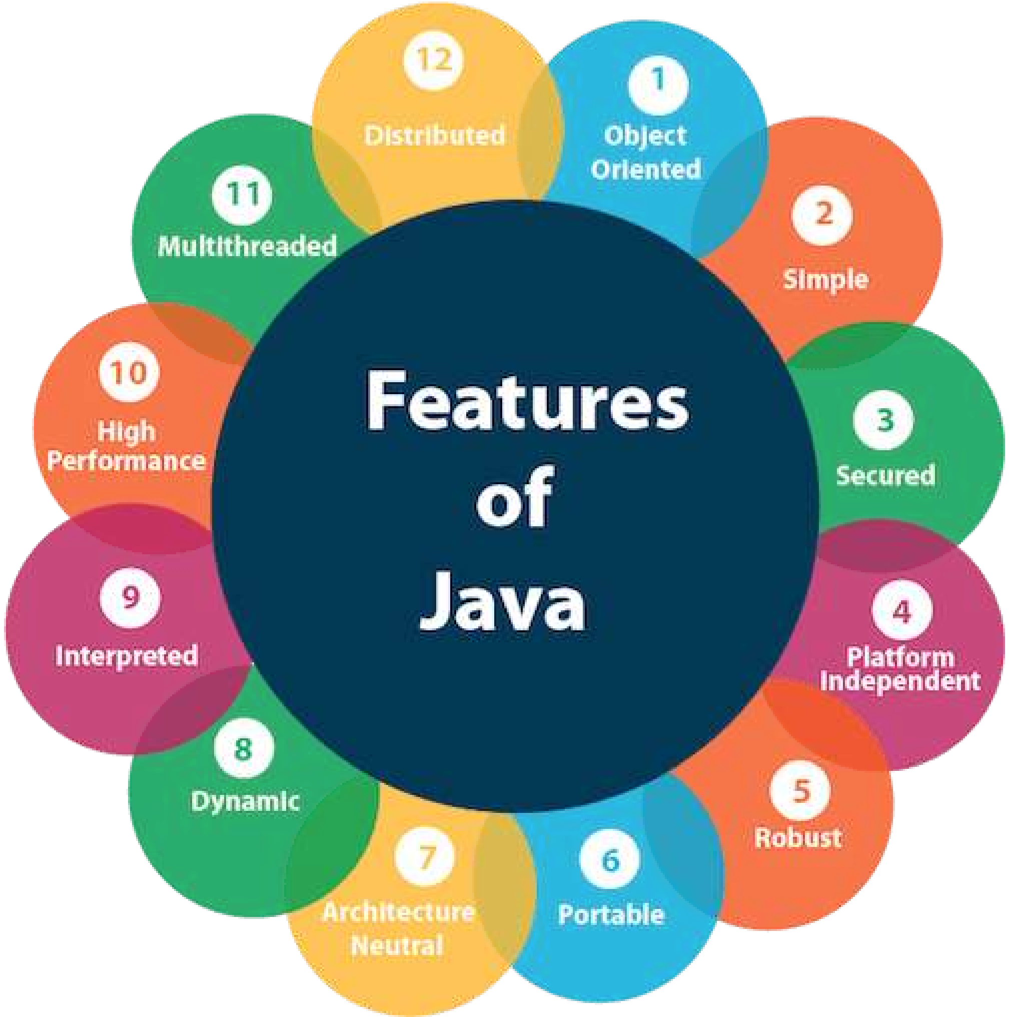
Java SE 17

Septmber 2021

Features of Java



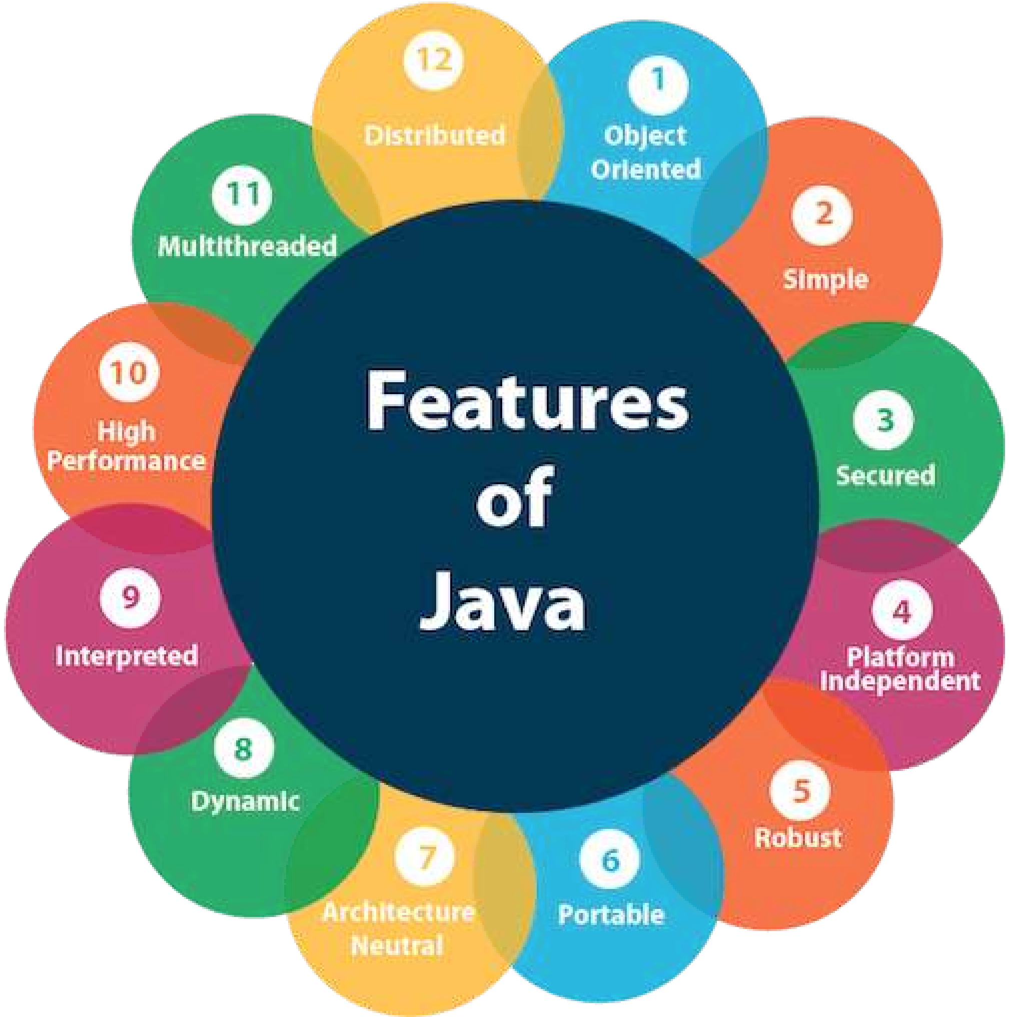
Features of Java



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 incorporate both data and behavior.
- 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

Features of Java

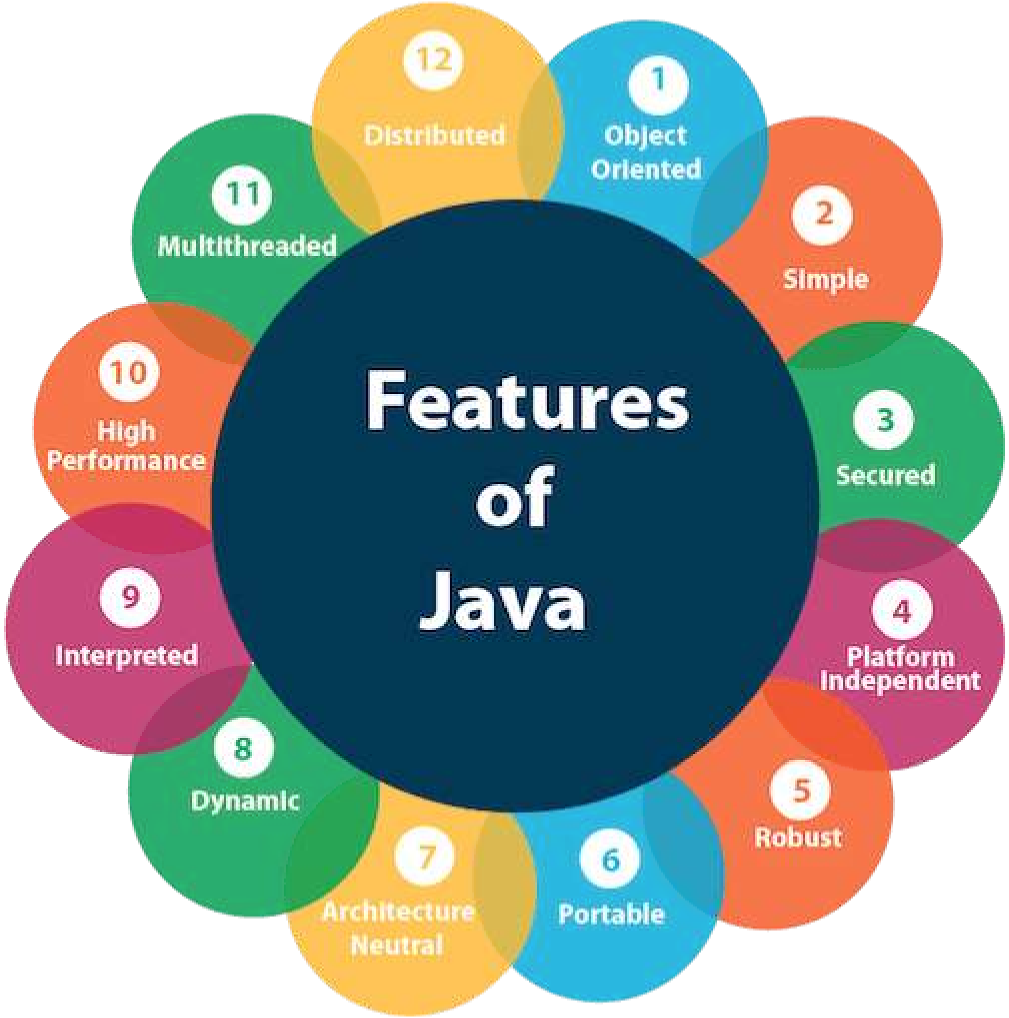


Simple

➢ Java is very easy to learn, and its syntax is simple, clean and easy to understand. According to Sun Microsystem, 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 complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.
- There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java.

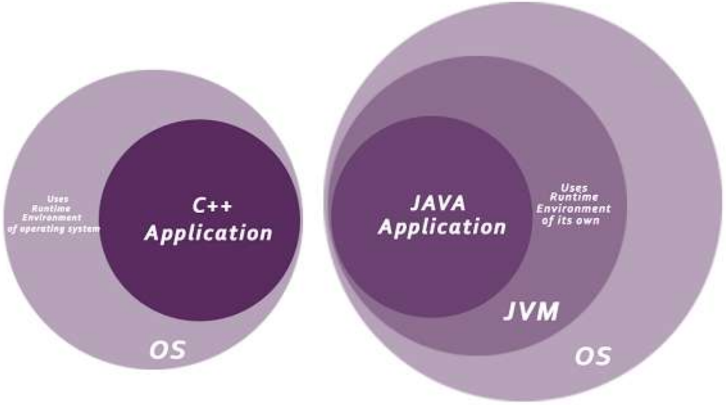
Features of Java



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 sandbox**



Uses Runtime Environment of operating system
C++ Application
OS
JAVA Application
Uses Runtime Environment of its own
JVM
OS

Features of Java

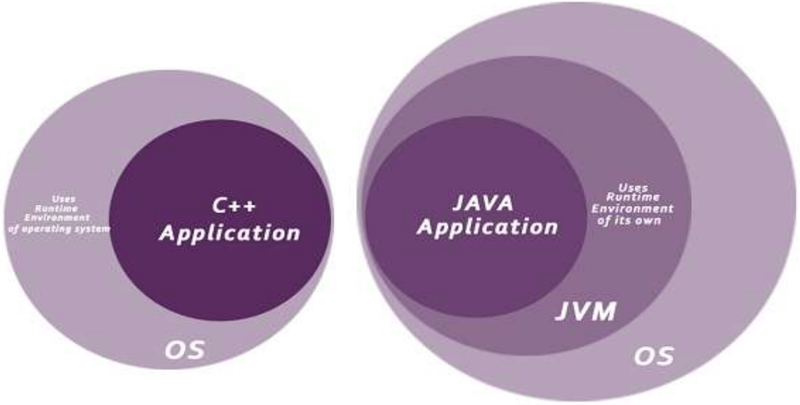
Secured

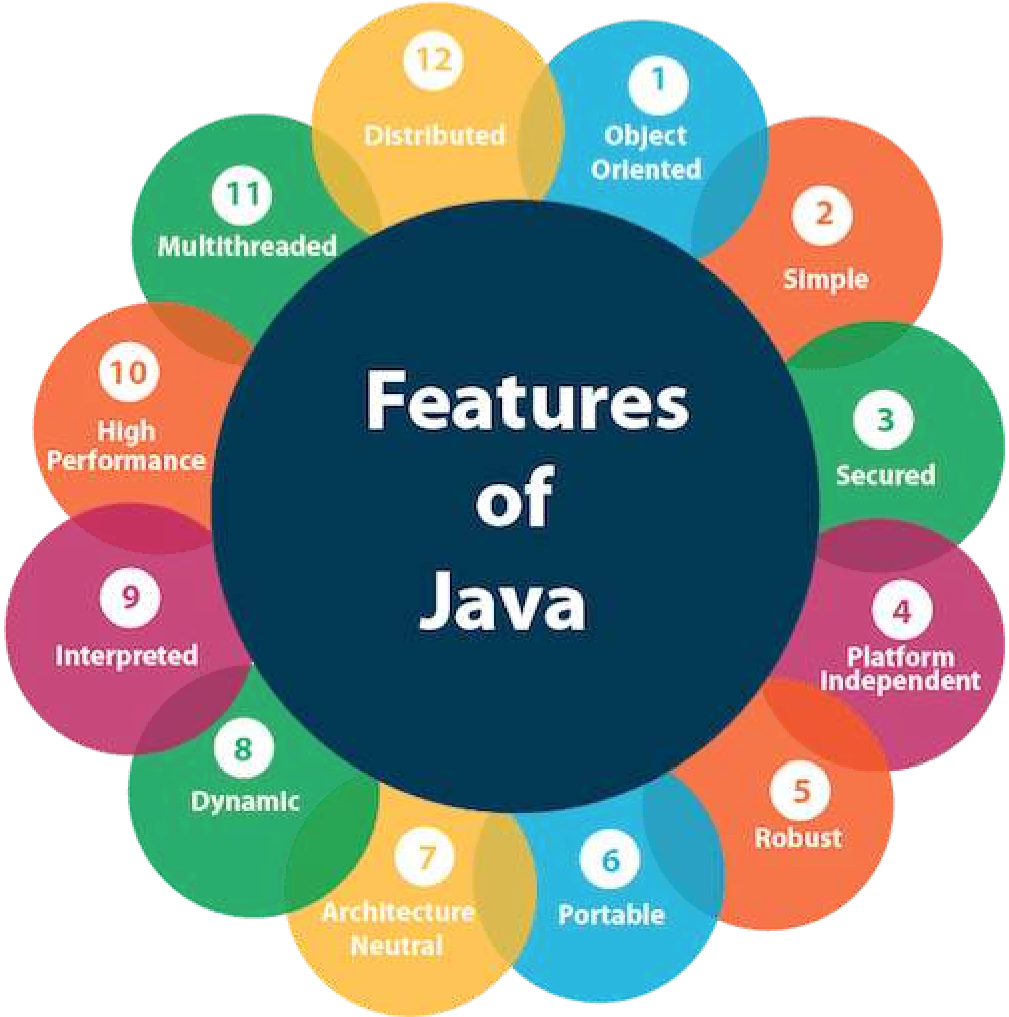
Java is best known for its security. With Java, we can develop virus-free systems. Java is secured because:

- **No explicit pointer**
- **Java Programs run inside a virtual machine sandbox**

- **Classloader:** Classloader in Java is a part of the Java Runtime Environment (JRE) which is used to load Java classes into the Java Virtual Machine dynamically. It adds security by separating the package for the classes of the local file system from those that are imported from network sources.
- **Bytecode Verifier:** It checks the code fragments for illegal code that can violate access rights to objects.
- **Security Manager:** It determines what resources a class can access such as reading and writing to the local disk.

Java language provides these securities by default. Some security can also be provided by an application developer explicitly through SSL, JAAS, Cryptography, etc.


Uses Runtime Environment of operating system
C++ Application
OS
JAVA Application
Uses Runtime Environment of its own
JVM
OS

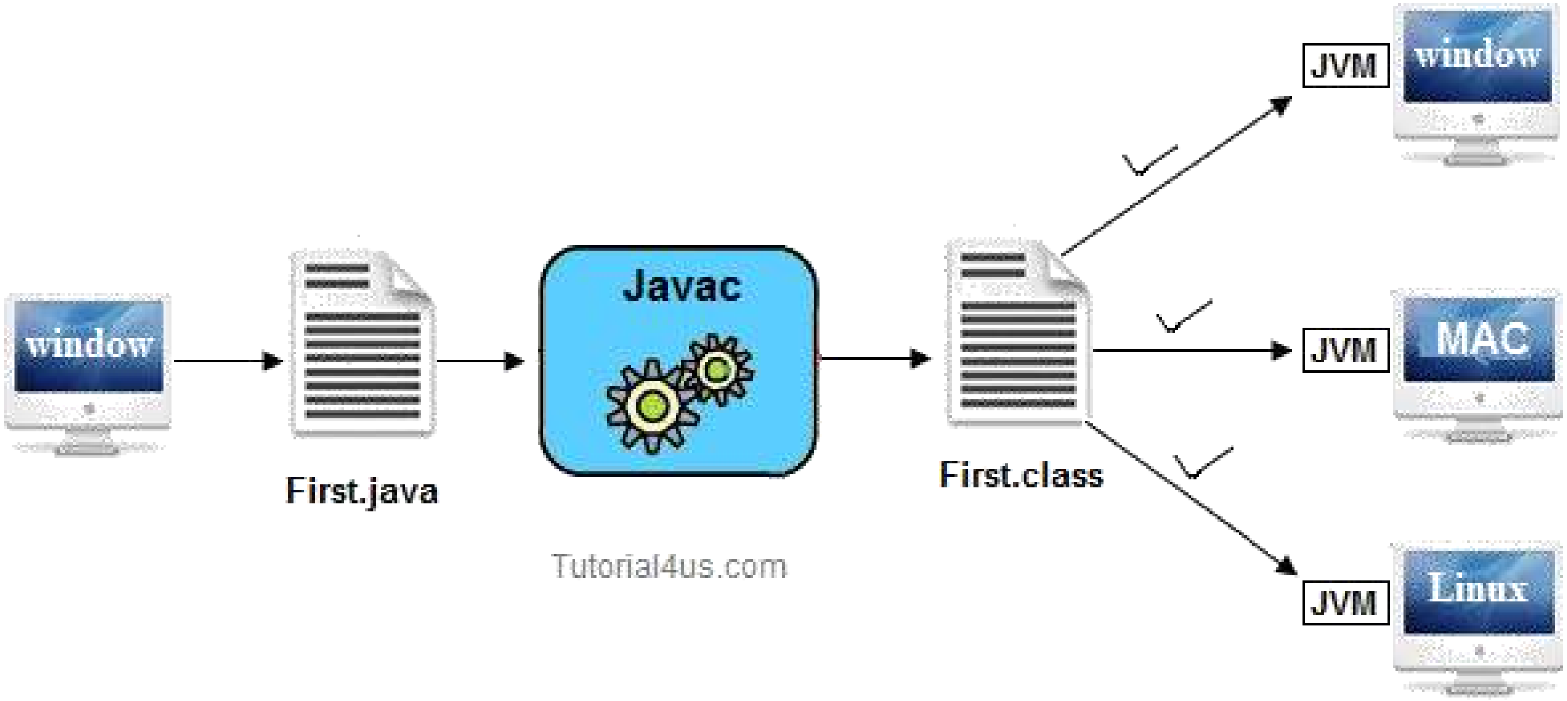


Features of Java

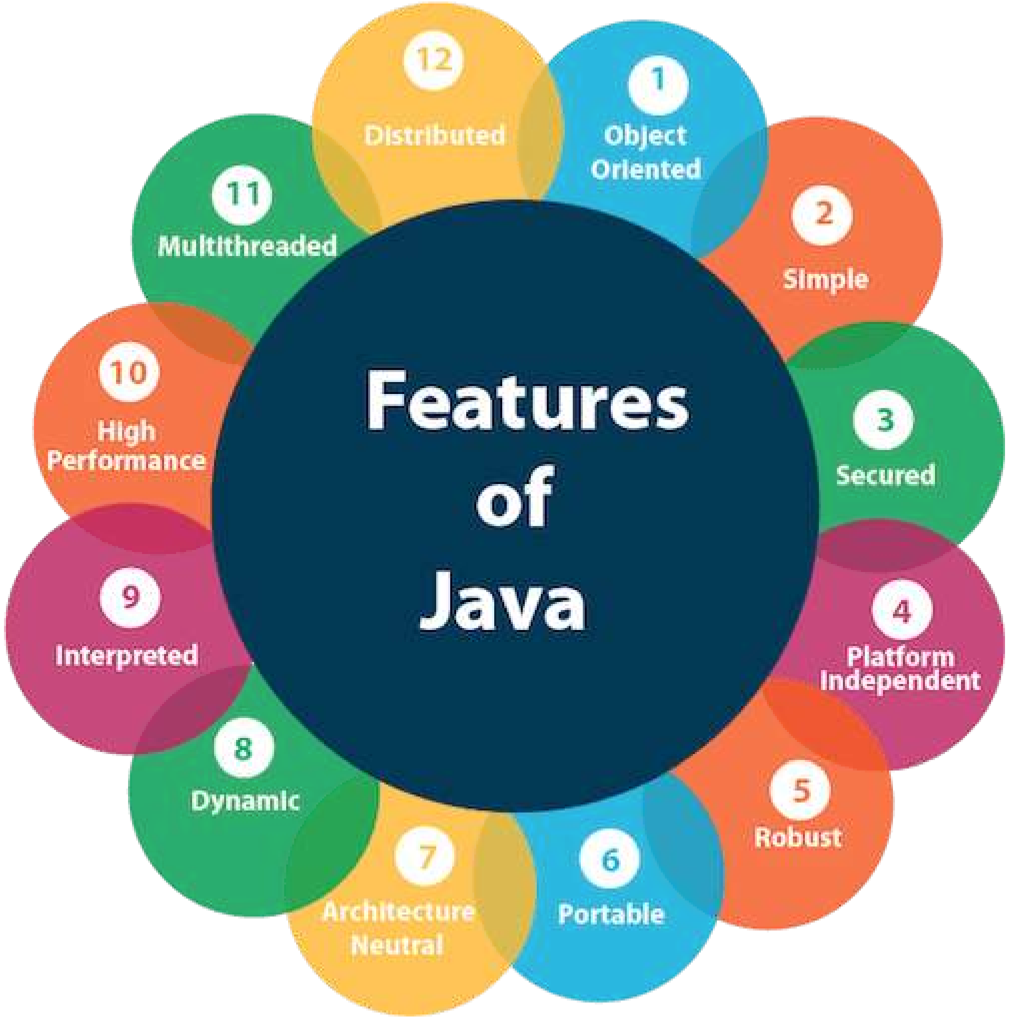
Platform Independent

- 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 a software-based platform.

Java is Platform Independent



Features of Java

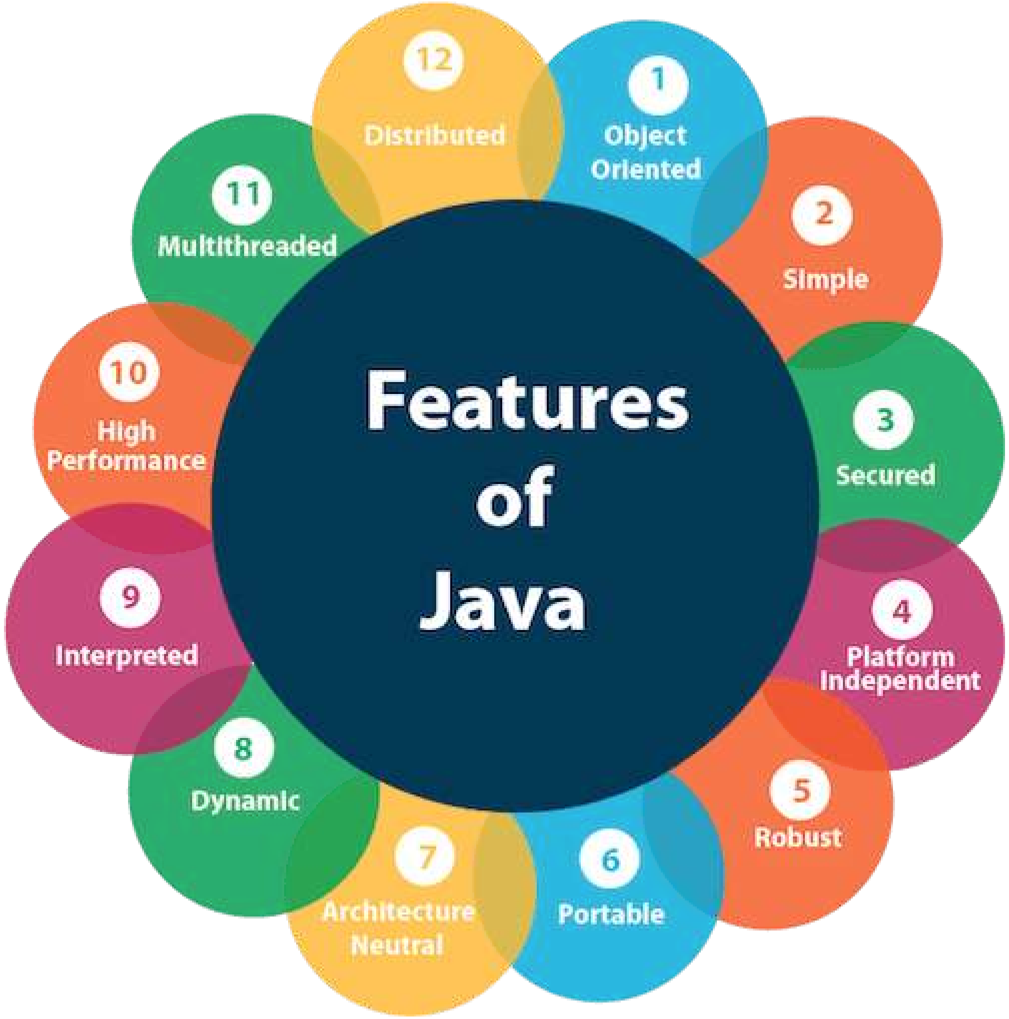


Platform Independent

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

1. Runtime Environment
2. API(Application Programming Interface)

- Java code can be executed on multiple platforms, for example, 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).

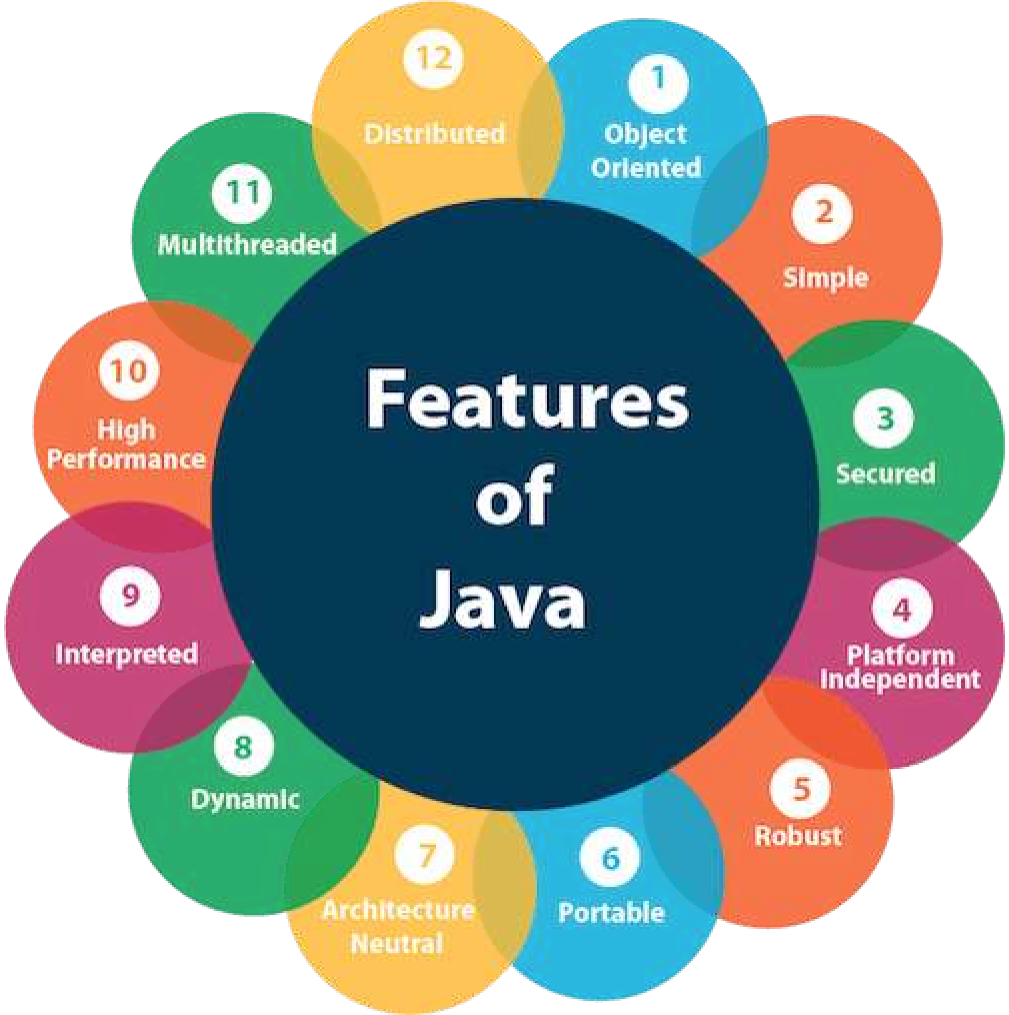


Features of Java

Robust

The English mining of Robust is strong. Java is robust because:

- It uses strong memory management.
- There is a lack of pointers that avoids security problems.
- Java provides automatic garbage collection which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.
- There are exception handling and the type checking mechanism in Java. All these points make Java robust.



Features of Java

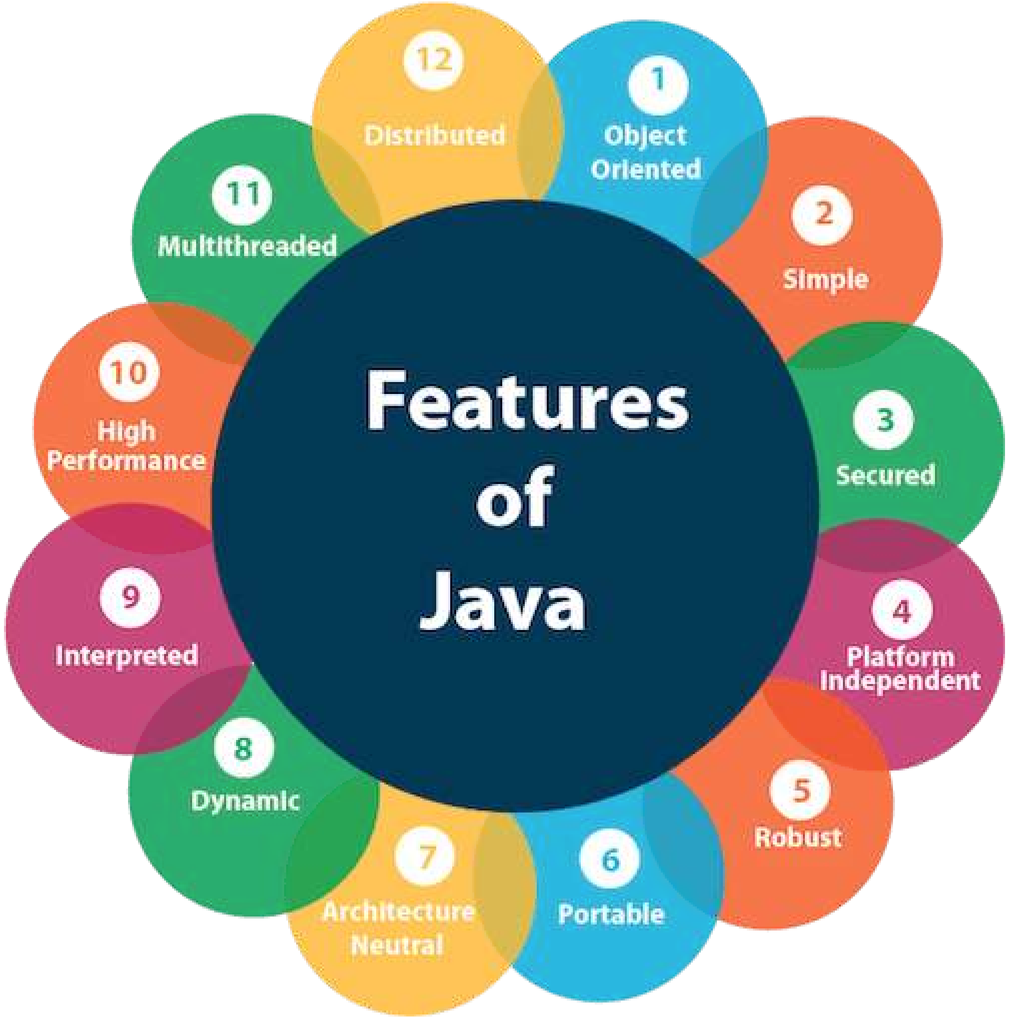
Portable

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

Architecture-neutral

Java is architecture neutral because there are no implementation dependent features, for example, the 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. However, it occupies 4 bytes of memory for both 32 and 64-bit architectures in Java.



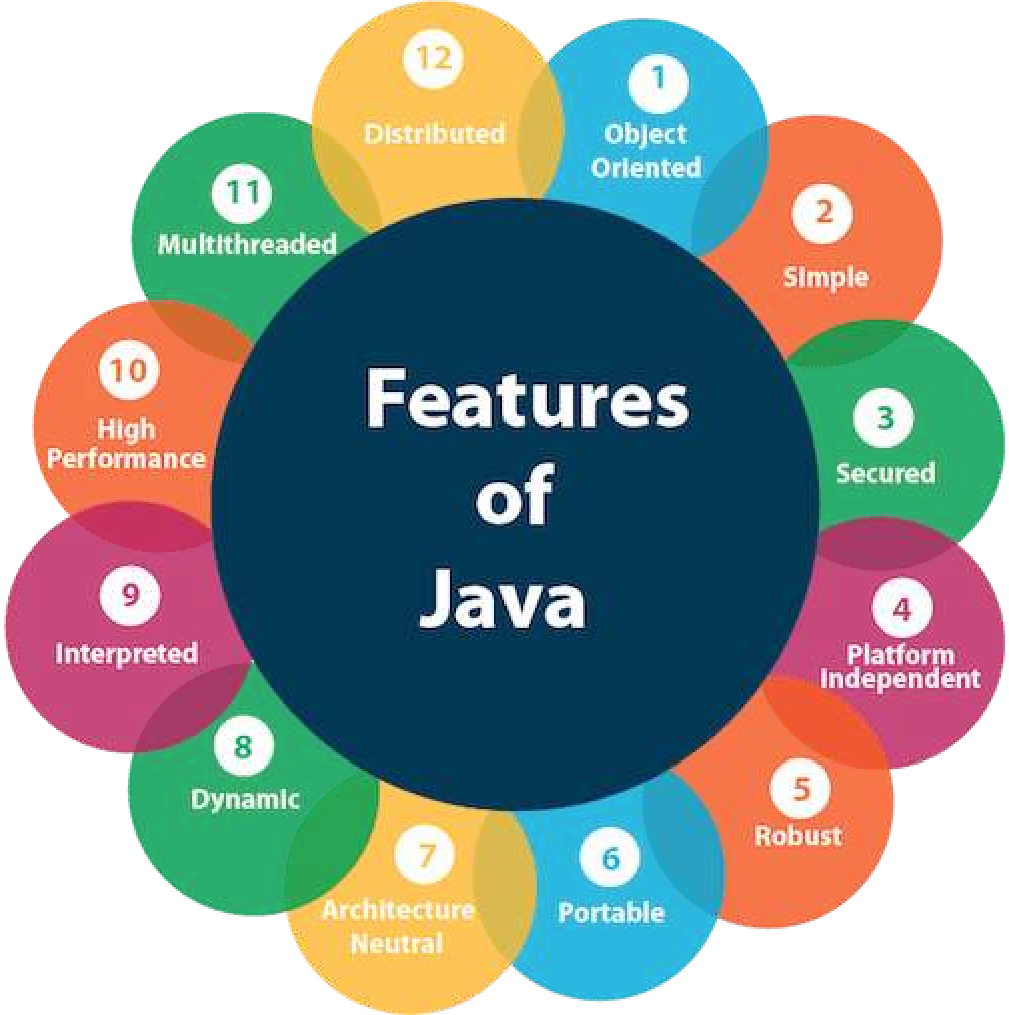
Features of Java

Dynamic

Java is a dynamic language. It supports the dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.

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.



Features of Java

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.

Distributed

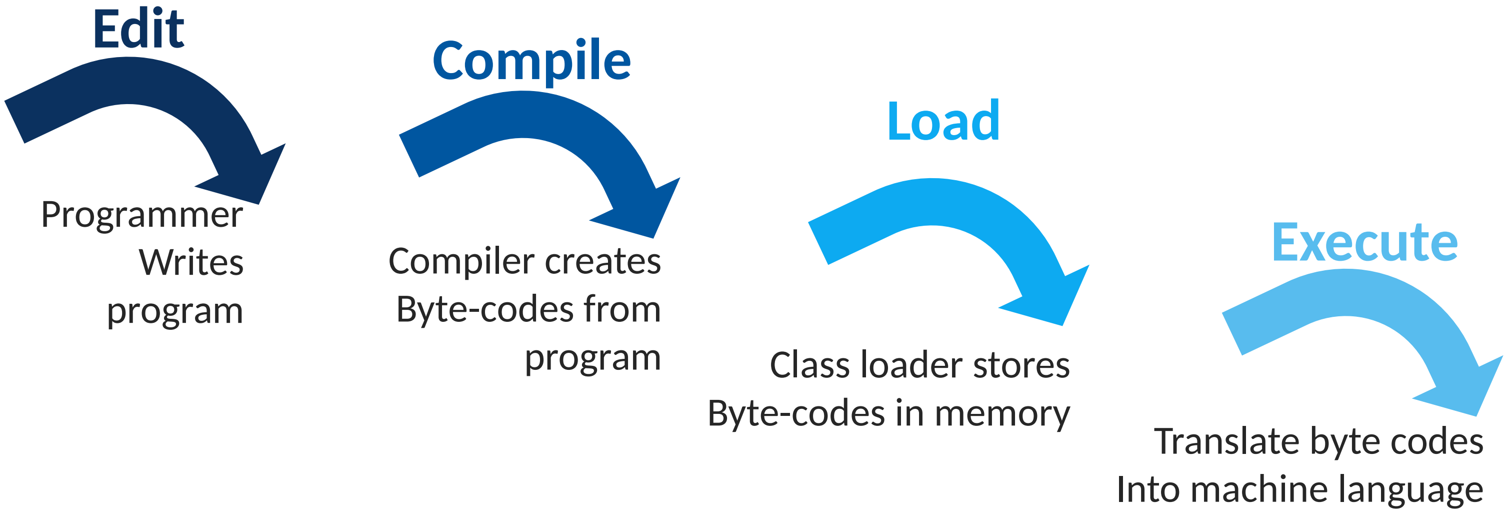
Java is distributed because it facilitates users to create distributed applications in Java. RMI and EJB are used for creating distributed applications. This feature of Java makes us able to access files by calling the methods from any machine on the internet.

Java is Simple

- It is **free from pointer** due to this execution time of application is improved. [Whenever we write a Java program without pointers then internally it is converted into the equivalent pointer program].
- It has **Rich set of API** (application protocol interface).
- It has **Automatic Garbage Collector** which is always used to collect un-Referenced (unused) Memory location for improving performance of a Java program.
- It contains **user friendly syntax for developing any applications.**

Java Life Cycle

Java Programs Normally Undergo Four Phases



Edit

Programmer Writes program

Compile

Compiler creates Byte-codes from program

Load

Class loader stores Byte-codes in memory

Execute

Translate byte codes Into machine language

Is Java Purely Object Oriented?

Yes Then why?

No Then Why?

Difference between JDK, JRE and JVM

JDK

Java Development Kit

JDK is an acronym for Java Development Kit. It physically exists. It contains JRE + development tools.

JRE

Java Runtime Environment

JRE is used to provide runtime environment. It is the implementation of JVM. It physically exists.

JVM

Java Virtual Machine

JVM is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.

JDK is a software development kit whereas **JRE is a software bundle** that allows Java program to run, whereas **JVM is an environment** for executing bytecode.

Java Editions

J2SE

Java 2 Standard Edition

Java standard edition is use to develop client-side standalone applications or applets

J2ME

Java 2 Micro Edition

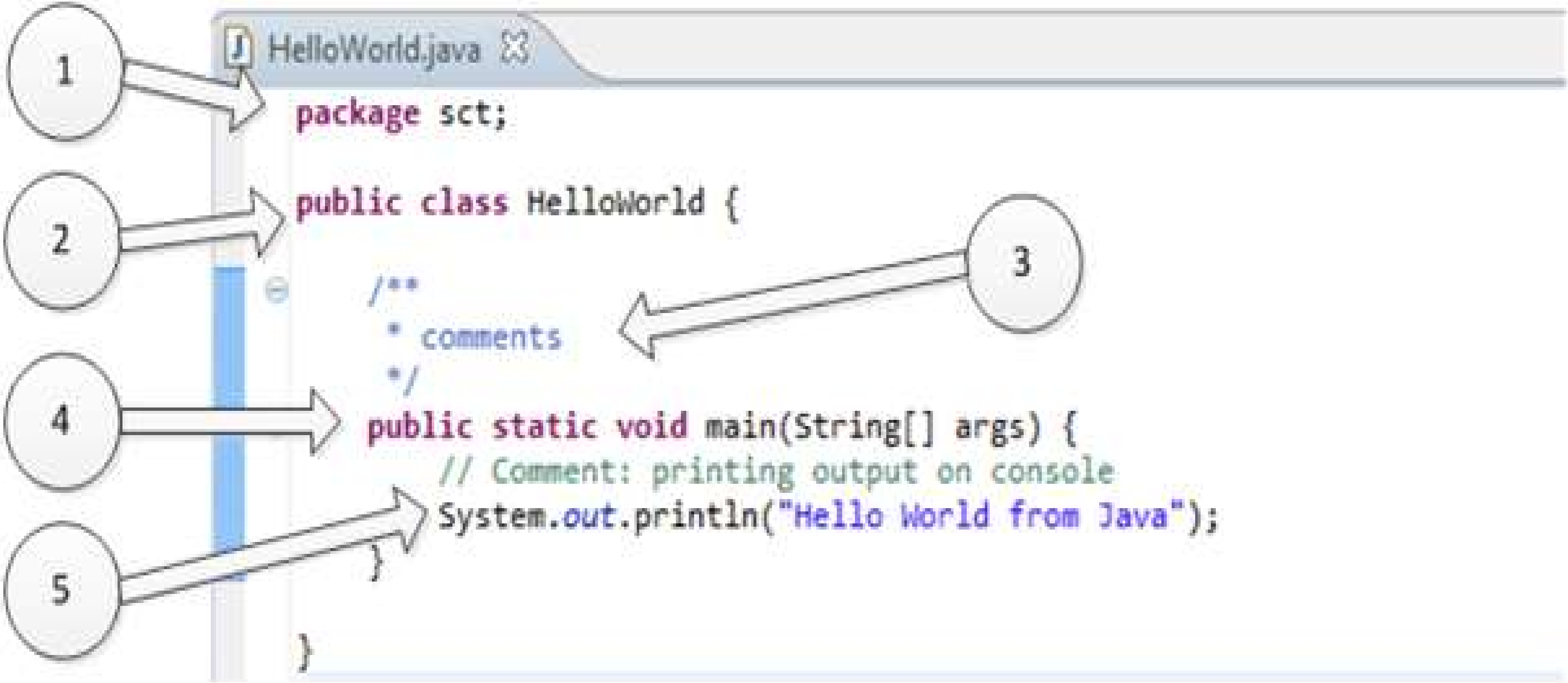
Java micro edition is use to develop applications for mobile devices such as cell phones

J2EE

Java 2 Enterprise Edition

Java enterprise edition is use to develop server-side applications such as Java servlets and Java Server Pages

First “Hello World” program using JAVA



```java
package sct;

public class HelloWorld {

    /**
     * comments
     */
    public static void main(String[] args) {
        // Comment: printing output on console
        System.out.println("Hello World from Java");
    }

}
```

“package sct”

- It is package declaration statement.
- defines a namespace in which classes are stored.
- to organize the classes based on functionality.
- If you omit the package statement, the class names are put into the **default package java.lang**, which has no name.
- Package statement cannot appear anywhere in the program. It must be the **first line of your program.**

2. “public class HelloWorld”

- This line has various aspects of java programming.
- **public:** This is access modifier keyword **which tells compiler access to class.** Various values of access modifiers can be **public, protected,private or default (no value).**
- **class:** This keyword used to declare a class. Name of class (HelloWorld) followed by this keyword.

3. “Comments”

- **Line comments:** It starts with **two forward slashes (//) and continues to the end of the current line.** Line comments do not require an ending symbol.
- **Block comments:** start with a forward slash and **an asterisk (/*) and end with an asterisk and a forward slash (*/).**Block comments can also extend **across as many lines as needed.**

4. “public static void main (String [ ] args)”:

- **public:** Access Modifier
- **static:** static is a reserved keyword which means that a method is accessible and usable even though no objects of the class exist.
- **void:** This keyword declares nothing would be returned from the method. The method can return any primitive or object.
- **Method** content inside curly braces. { }

5. System.out.println("Hello World from Java")

- **System**: It is the name of Java utility class.
- **out**:It is an object which belongs to System class.
- **println:** It is utility method name which is used to send any String to the console.
- **“Hello World from Java”:** It is String literal set as argument to println method.