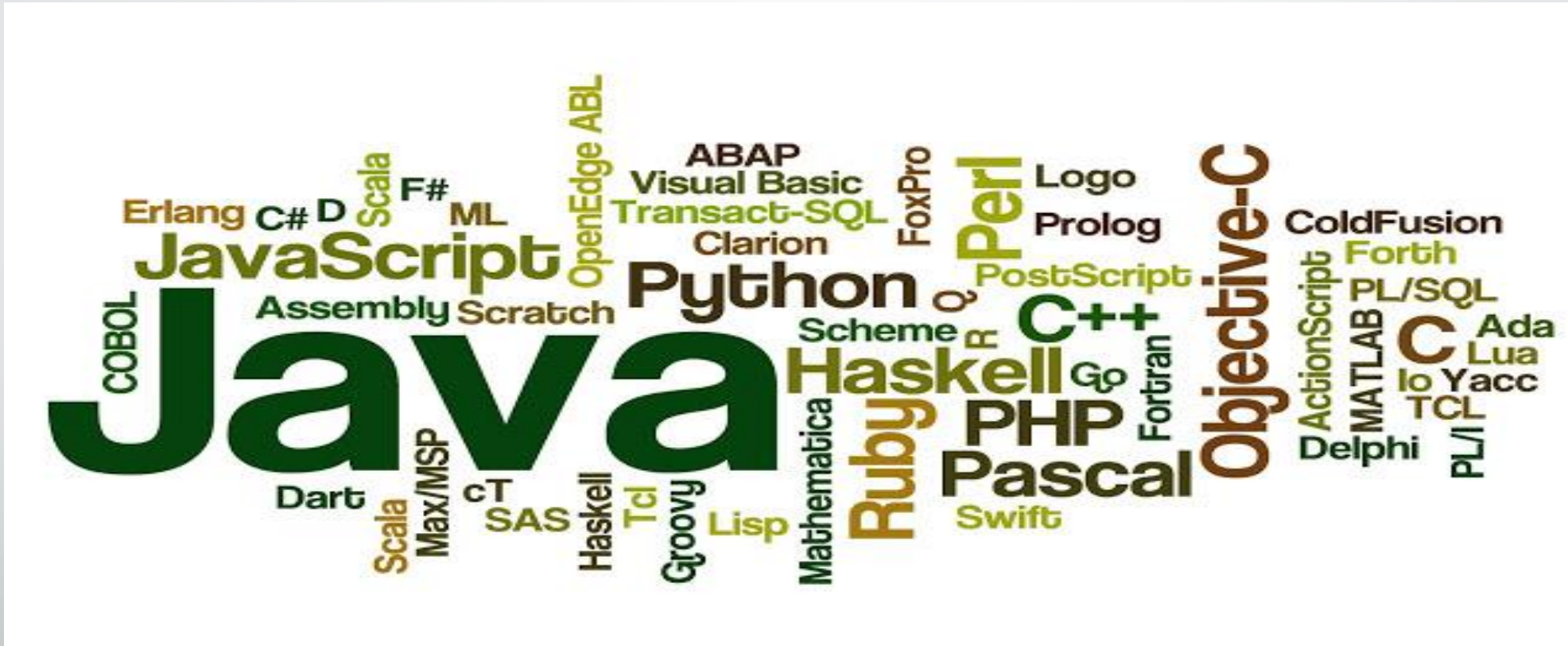


# Welcome To Java Programming



# History of Java

- **Java history** is interesting to know. The history of java starts from Green Team. Java team members (also known as **Green Team**), initiated a revolutionary task to develop a language for digital devices such as set-top boxes, televisions etc.
- For the green team members, it was an advance concept at that time. But, it was suited for internet programming. Later, Java technology as incorporated by Netscape.
- 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

# James Gosling



# Why Oak name for java language?

- 1) **James Gosling, Mike Sheridan, and Patrick Naughton , John** 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.



- 5) **Why Oak?** Oak is a symbol of strength and chosen as a national tree of many countries like U.S.A., France, Germany, Romania etc.
- 6) In 1995, Oak was renamed as "**Java**" because it was already a trademark by Oak Technologies.

# Why Java name for java language?

- 7) **Why they choosed 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.
- 8) Java is an island of Indonesia where first coffee was produced (called java coffee).
- 9) Notice that Java is just a name not an acronym.
- 10) Originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995.
- 11) In 1995, Time magazine called **Java one of the Ten Best Products of 1995**.
- 12) JDK 1.0 released in (January 23, 1996).

# Java Version History

- There are many java versions that has been released. Current stable release of Java is Java SE 8.
- JDK Alpha and Beta (1995)
- JDK 1.0 (23rd Jan, 1996)
- JDK 1.1 (19th Feb, 1997)
- J2SE 1.2 (8th Dec, 1998)
- J2SE 1.3 (8th May, 2000)
- J2SE 1.4 (6th Feb, 2002)
- J2SE 5.0 (30th Sep, 2004)
- Java SE 6 (11th Dec, 2006)
- Java SE 7 (28th July, 2011)
- Java SE 8 (18th March, 2014)

# Java Tutorial

- Java Tutorial or Core Java Tutorial or Java Programming Tutorial is a widely used robust technology. Let's start learning of java from basic questions like what is java tutorial, core java, where it is used, what type of applications are created in java and why use java.



# 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:

- Desktop Applications such as acrobat reader, media player, antivirus etc.
- Web Applications such as irctc.co.in, javatpoint.com etc.
- Enterprise Applications such as banking applications.
- Mobile
- Embedded System
- Smart Card
- Robotics
- Games etc.

# Types of Java Applications

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

- **1) Standalone Application**

It is also known as desktop application or window-based application. An application that we need to install on every machine such as 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, jsf etc. technologies are used for creating web applications in java.

- **3) Enterprise Application**

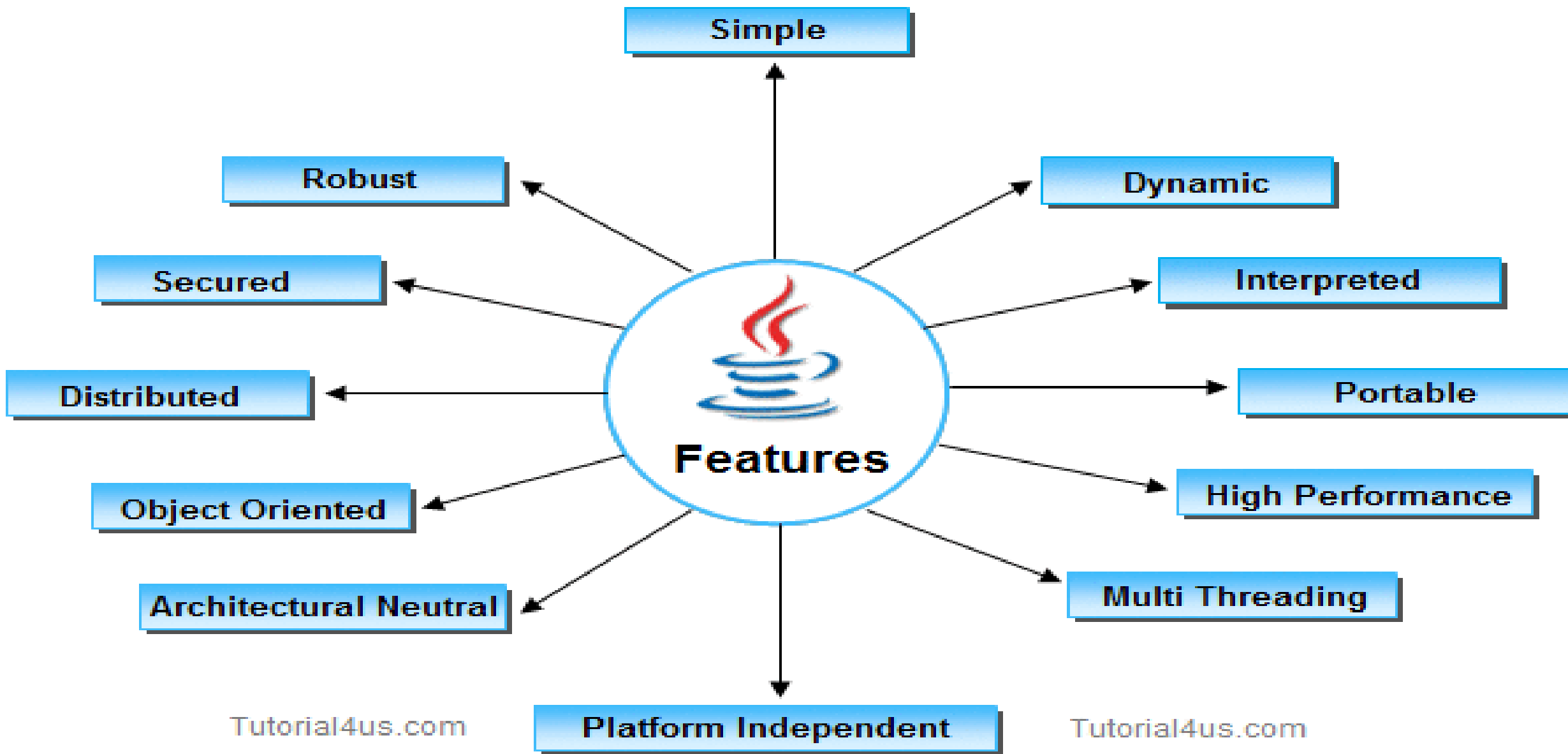
An application that is distributed in nature, such as banking applications etc. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.

- **4) Mobile Application**

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

# Features of Java

The main features of java is it is object oriented programming language. Features of any language are nothing but the set of services or facilities provided by the language vendors to the industry programmers. Some important features are;



# Some Important Features of Java are;

- Simple
- Object oriented
- Secure
- Platform independent
- Multithreaded
- Architectural neutral
- Distributed
- Dynamic
- Portable
- High performance

# Reference



<http://www.tutorial4us.com/java/features-of-java>



## Compiled and Interpreted

- In the second stage Java interpreter generates machine code
- That can be directly executed by the machine



Fig. 2.1 Compiled and Interpreted

# Platform Independent and Portable

- The most significant feature of Java is its portability
- Java programs can be easily moved from one computer system to another ,anywhere and at anytime

# Platform Independent and Portable

Contd..

- Changes and upgrades in
  - Operating systems
  - Processors
  - System resources

will not force any changes in Java programs

# Platform Independent and Portable

Contd..

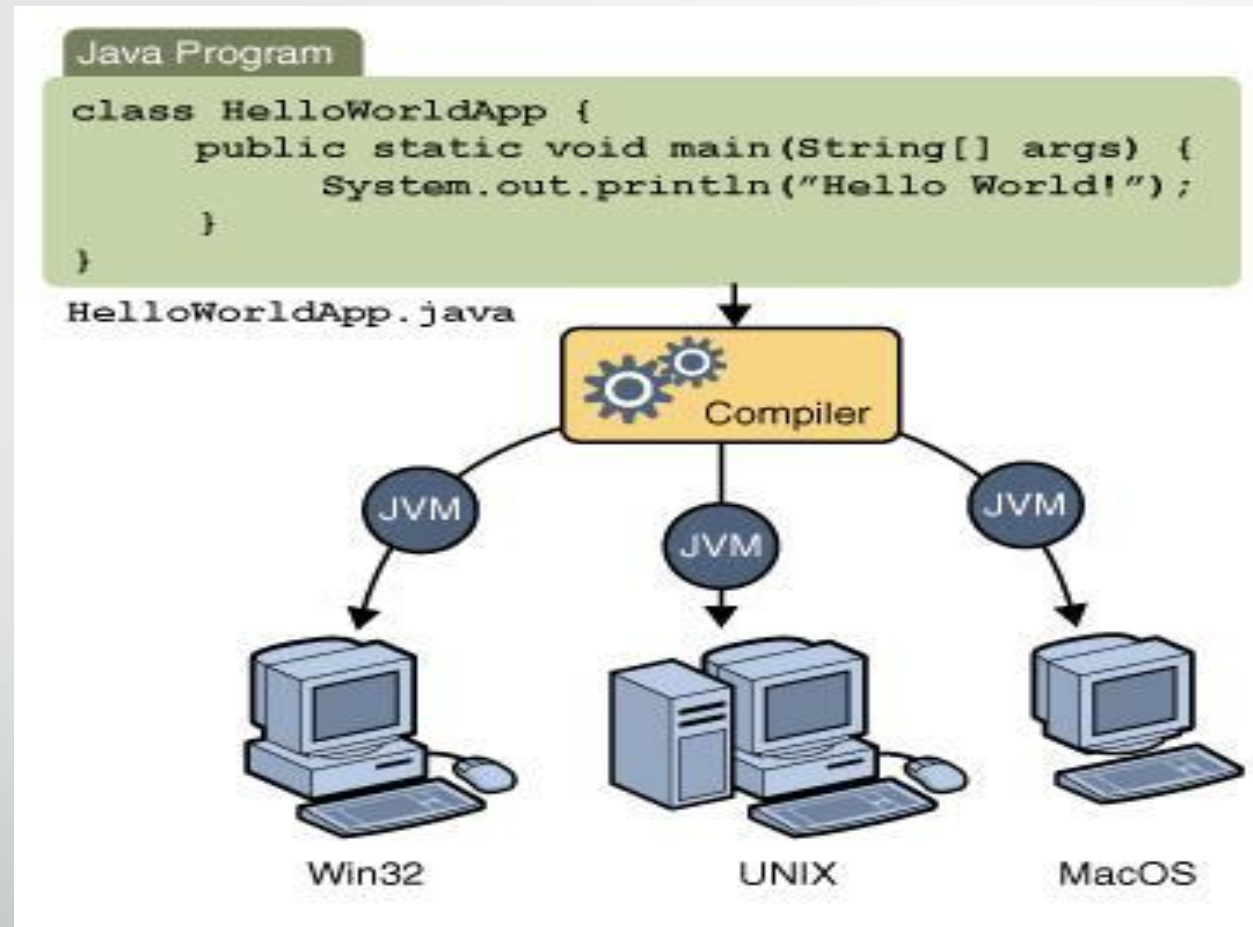


Fig. 2.2 Once compiled, Java class file can be run on any platform

# Object Oriented

- Java is a true object oriented language
- Almost everything in Java is an object
- All programs code and data reside within objects and classes
- Java comes with an extensive set of classes arranged in packages

# Robust and Secure

- Java is a robust language
- It provides many safeguards to ensure reliable code
- It has strict compile time and runtime for data types

# Robust and Secure

Contd..

- It is designed as a garbage collected language
- Java also provides concept of exception handling which identifies errors and eliminates

# Robust and Secure

Contd..

- Security becomes an important issue for a language that is used for programming on internet
- Threat of viruses and abuse of resources
- Java systems not only verify all memory access, but also ensure that no viruses are communicated with an applet



# Robust and Secure

Contd..

- The absence of pointers in Java ensures that programs cannot gain access to memory location without proper initialization

# Distributed

- Java is designed as a distributed language for creating applications on networks
- It has the ability to share both data and programs

# Distributed

Contd..

- Java applications can open and access remote object as easily as they can do in a local system
- Enables multiple programmers at multiple locations work together on a single project

## Simple, Small and Familiar

- Java is small and simple language. Many features of C and C++ are not part of Java
  - eg. : Java does not provide
    - Pointers
    - Preprocessors header file
    - goto statements
    - Operator overloading
    - Multiple inheritance

# Simple, Small and Familiar

Contd..

- Familiarity is another feature of Java
- It is modeled on C and C++ languages
- Java is a simplified version of C++

# Multithreaded and Interactive

- Multithreaded means handling multiple tasks simultaneously
- Java supports multithreaded programs
- Need not wait for the application to finish one task before beginning another
- Greatly improves the interactive performance of graphical applications

# High Performance

- Java performance is impressive for an interpreted language
  - Due to the use of intermediate byte code
- Java architecture is also designed to reduce overheads during runtime
- The incorporation of multithreading enhances the overall execution speed of Java programs

# Dynamic and Extensible

- Java is a dynamic language
- Java is capable of dynamically linking new
  - Class libraries
  - Methods
  - Objects
- Java programs supports functions written in other languages such as C and C++
  - These functions are known as “native methods”



# Dynamic and Extensible

Contd..

- This facility enables the programmers to use the efficient available in these languages
- Native methods are dynamically linked at runtime

# Java and Internet

- Java is strongly associated with the Internet. Internet users can use Java to create **applet programs and run them locally using a "Java-enabled browser"** such as **HotJava**. They can also use a Java-enabled browser to download an applet located on a computer anywhere in the Internet and run it on his local computer. In fact, Java applets have made the Internet a true extension of the storage system of the local computer.
- Internet users can also setup their websites containing java applets that could be used by other remote users of Internet. This feature made Java most popular programming language for Internet

# Java and World Wide Web

- World Wide Web (WWW) is an open-ended information retrieval system designed to be used in the Internet's distributed environment. This system contains Web pages that provide both information and controls. Web system is open-ended and we can navigate to a new document in any direction. This is made possible with the help of a language called *Hypertext Markup Language* (HTML). Web pages contain HTML tags that enable us to find, retrieve, manipulate and display documents worldwide.
- Java was meant to be used in distributed environments such as Internet. Since, both the Web and Java share the same philosophy, Java could be easily incorporated into the Web system. Before Java, the World Wide Web was limited to the display of still images and texts. However, the incorporation of Java into Web pages has made it capable of supporting animation, graphics, games, and a wide range of special effects.

# Virtual machine

- Before understanding what is JVM let us first know what virtual machine is.

A virtual machine is a layer of abstraction that gives a program one simplified interface for interacting with a variety of physical computers and their operating systems.

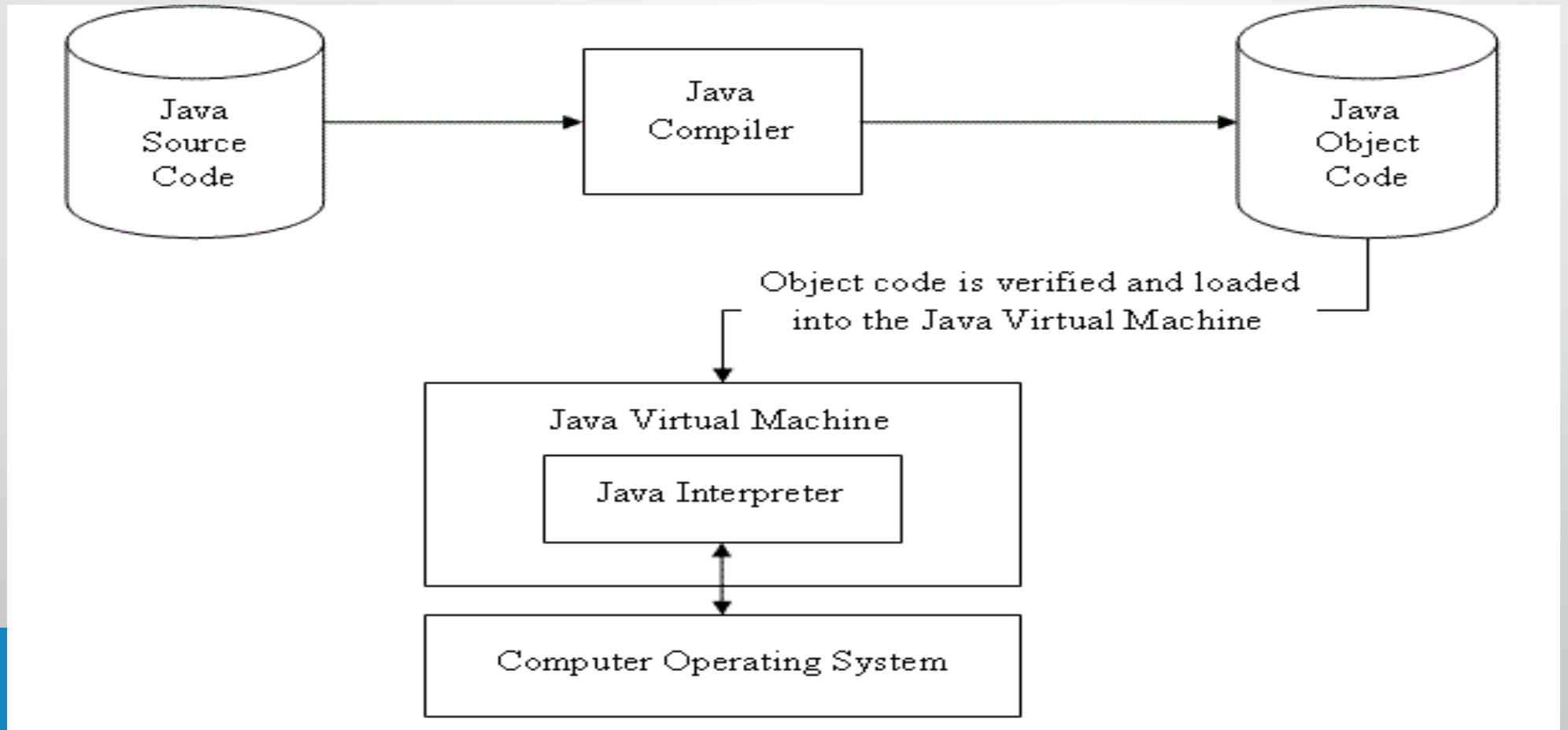
# Java virtual machine

- As the name indicates, JVM is not a real hardware machine but a software layer which resembles an hardware platform.
- JVM converts Java byte code into machine language and executes it.

The byte code can be executed on any platform where there exist JVM.

- JVM's are available for many hardware and software platforms.
- The use of the same byte code for all JVM's on all platforms allows Java to be described as a "write once, run anywhere" programming language.
- Thus, the JVM is a crucial component of the Java platform.

# Diagram of JVM



# Components of JVM

1. Byte code verifier
2. Class loader
3. Execution engine
4. Garbage collector
5. Security Manager



The following figure shows a block diagram of the JVM that includes its major subsystems and memory areas.

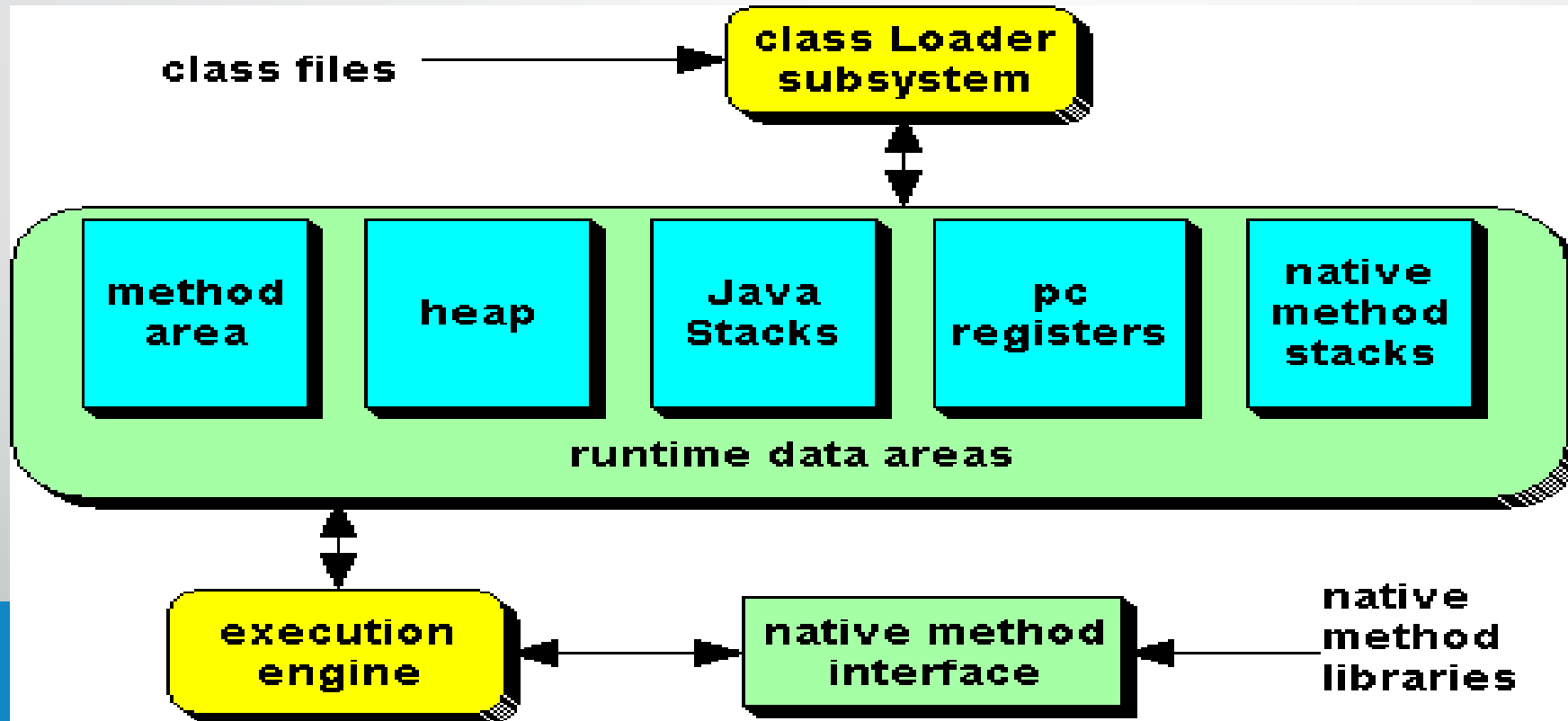


FIG: Memory Configuration of JVM

# Introduction to the JVM (Cont'd)

---

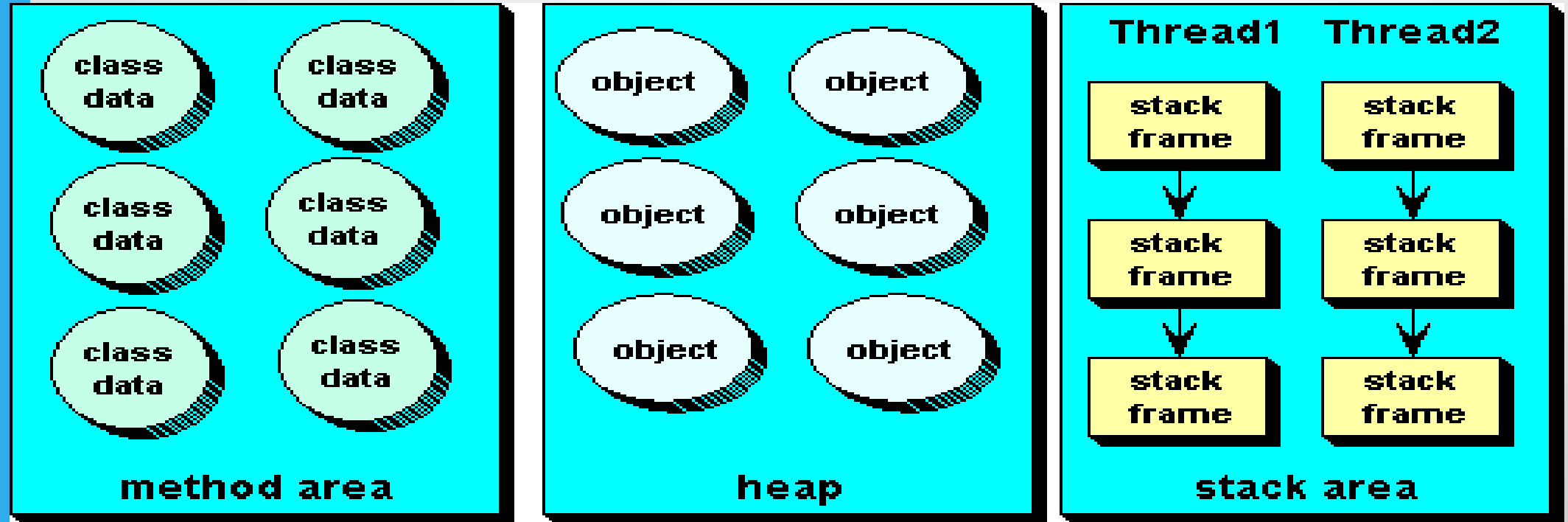
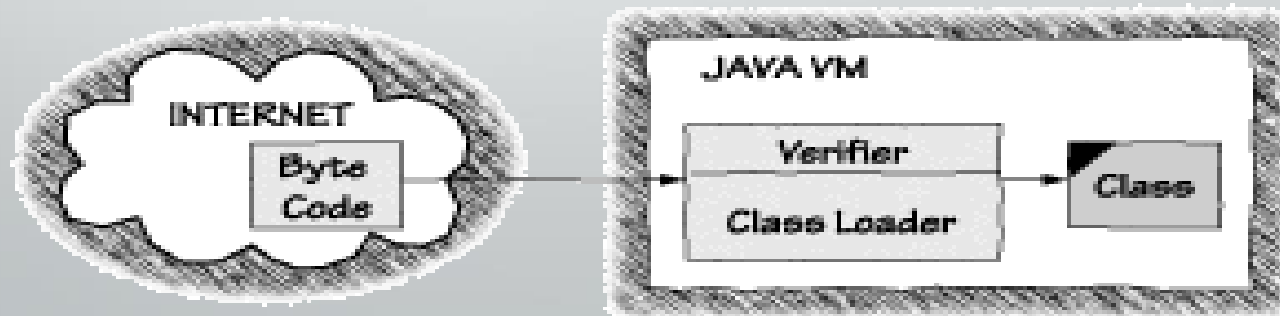


Figure 2: Content of Memory Blocks at runtime.

# Byte code verifier

- As the name suggests, bytecode verifier is used to verify the bytecode.
- Bytecode verifier checks for unusual code.
- Byte code verifier is a crucial component for security.



# Class loader

- Class loader loads java classes into java virtual machine.
- All Java virtual machines include one class loader that is embedded in the virtual machine.

The main feature of the class loader is that JVM.

- doesn't need to have any knowledge about the classes that will be loaded at runtime.
- Class loader reads bytecode and creates the instance of `java.lang.class`.

# Execution engine

- The execution engine helps JVM to convert bytecode into machine code.
- It has two parts:
  - a) Interpreter
  - b) Just-in-time-interpreter
- Execution engine is responsible for executing the instructions contained in the methods of loaded classes.

# Garbage collector

- Garbage collection is the process of automatically freeing objects that are no longer referenced by the program.

It periodically check for the object on heap , whose link is broken so it can collect garbage from heap.

- Garbage collection relieves java programmer from memory management.

# Security manager

- Security manager constantly monitors the code.
- It is special java object that is responsible for guarding security policies for java applications.
- It is always consulted before any potentially dangerous operation is requested by a java application.



# That's All