INTRODUCTION TO JAVA PROGRAMMING

JAVA – OVERVIEW

- A high-level programming language.
- Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.
- With the advancement of Java and its widespread popularity, multiple configurations were built to suite various types of platforms. Ex: J2EE for Enterprise Applications, J2ME for Mobile Applications.
- Java is guaranteed to be **Write Once, Run Anywhere.**

INTRODUCTION TO JAVA

- James Gosling and his team members initiated the Java language project in June 1991.

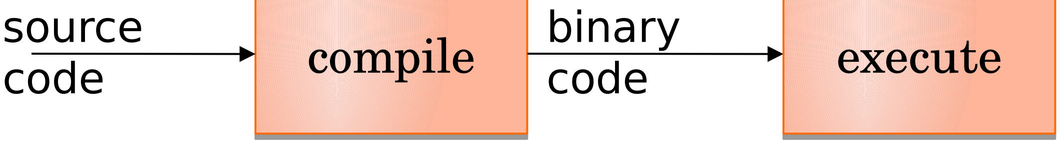
- The idea was to develop a language which was platform-independent.

- The language took 18 months to develop and had an initial name as Oak which was renamed to Java in 1995, due to copyright issues.

- Originally developed by James Gosling at Sun Microsystems(which has since merged into Oracle Corporation) and released in 1995.

COMPILERS, INTERPRETERS, AND THE JVM

Compiled Languages (e.g. C, C++)


source code

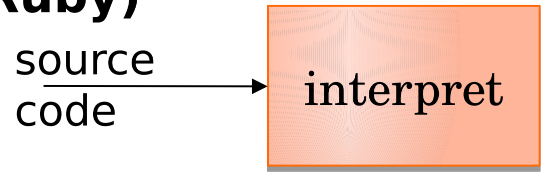
compile

binary code

execute

Compiler is unique to each platform

Interpreted Languages (e.g. JavaScript, Perl, Ruby)


source code

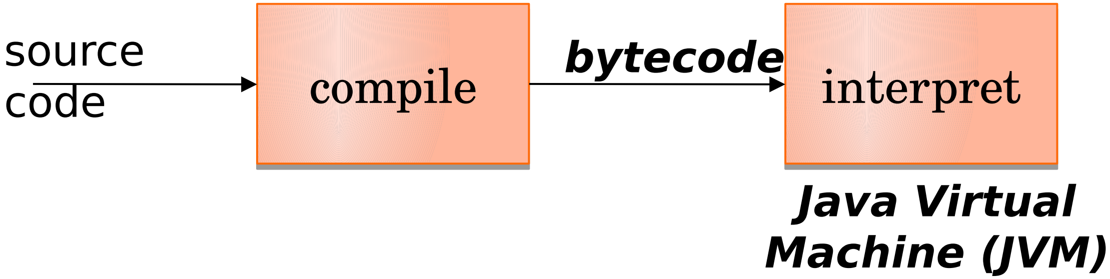
interpret

Interpreter translates code into binary and executes it

Small, easy to write

Interpreter is unique to each platform

Java


source code

compile

bytecode

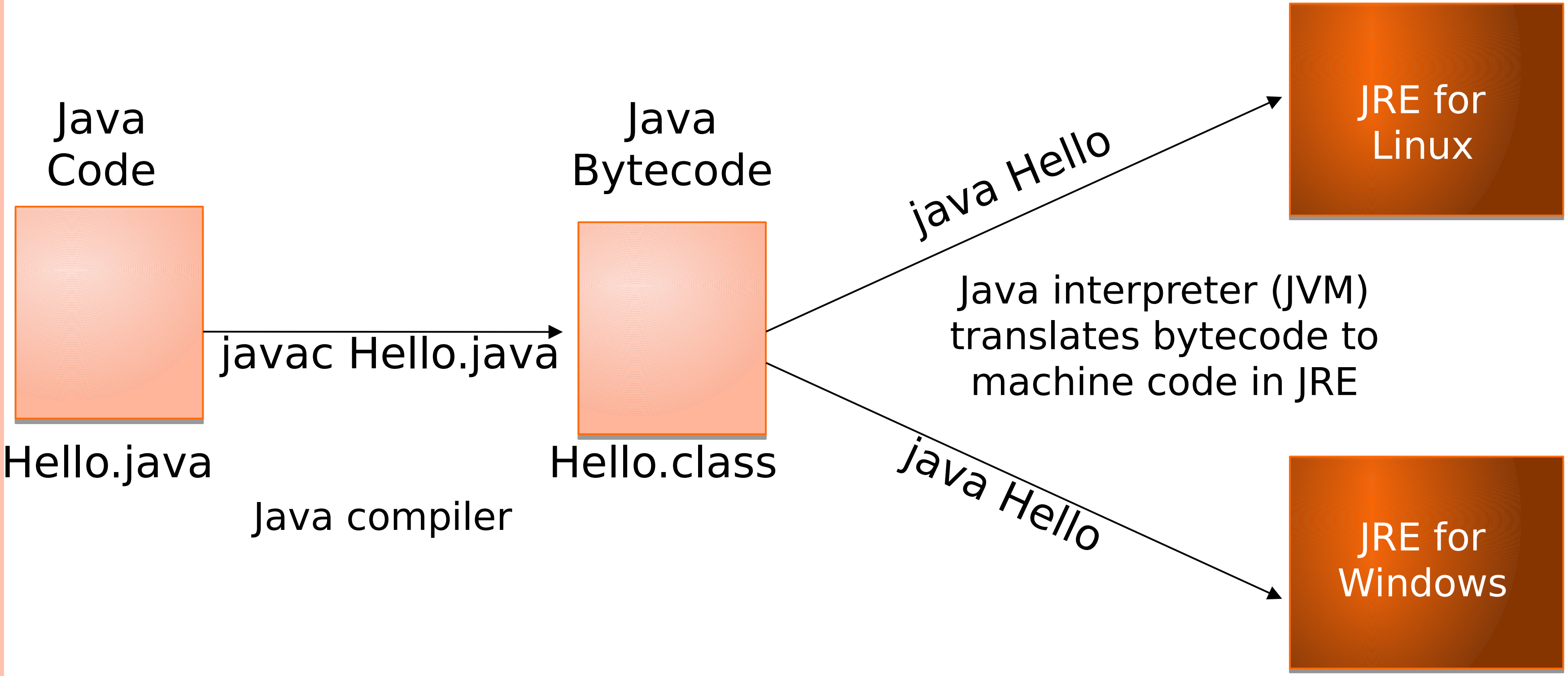
interpret

Java Virtual Machine (JVM)

Bytecode is platform independent

JVM is unique to each platform

COMPILING AND RUNNING JAVA



Java Code

Hello.java

javac Hello.java

Java compiler

Java Bytecode

Hello.class

java Hello

JRE for Linux

Java interpreter (JVM) translates bytecode to machine code in JRE

java Hello

JRE for Windows

PHASES OF PROGRAM CREATION AND EXECUTION

- **Edit :**
  - Creating a Java program consists of using an editor program.
  - Java source code files are saved with the file extension “**.java**”.
  - Source files can be created using a simple text editor, or an IDE (Integrated Development Environment), such as JCreator, Eclipse, JBuilder, etc.
  - IDEs provide tools to support the development process, including editors for writing programs and debugging for locating logic errors in programs.

PHASES OF PROGRAM CREATION AND EXECUTION

Compile

- During this phase, the programmer compiles the program using a command at the command line, or tools from the IDE.

- At this step, the Java source code is translated into **bytecodes**.

- If you are running the Java compiler from the command prompt, the command to enter is:
  - **Command: javac Hello.java***

- Running this command would compile the Java source file, Hello.java, and generate a bytecode class file named, Hello.class.
- Compiler is available as part of the Java Development Kit (JDK)

PHASES OF PROGRAM CREATION AND EXECUTION

- **Load**
  - The program must be placed in memory before it can execute.
  - In loading, the class loader takes the “.class” files, created in the previous step, and transfers them to primary memory.
  - The class loader also loads the .class files provided by Java, that your program uses.

PHASES OF PROGRAM CREATION AND EXECUTION

- **Verify**
  - As the classes are loaded, a bytecode verifier examines the bytecodes to ensure they are valid and don't violate any security restrictions.

- **Execute**
  - During the last phase, the JVM executes a programs bytecodes, performing the actions specified by the program.
    - **Command: java Hello**
  - A version of Java Runtime Environment (JRE) which incorporates a JVM is required to execute the bytecode and the Java library packages.

JDK, JRE AND JVM

- JRE is an acronym for Java Runtime Environment.
- It is used to provide runtime environment.

- It is the implementation of JVM.
- It contains set of libraries + other files that JVM uses at runtime.

JDK, JRE AND JVM

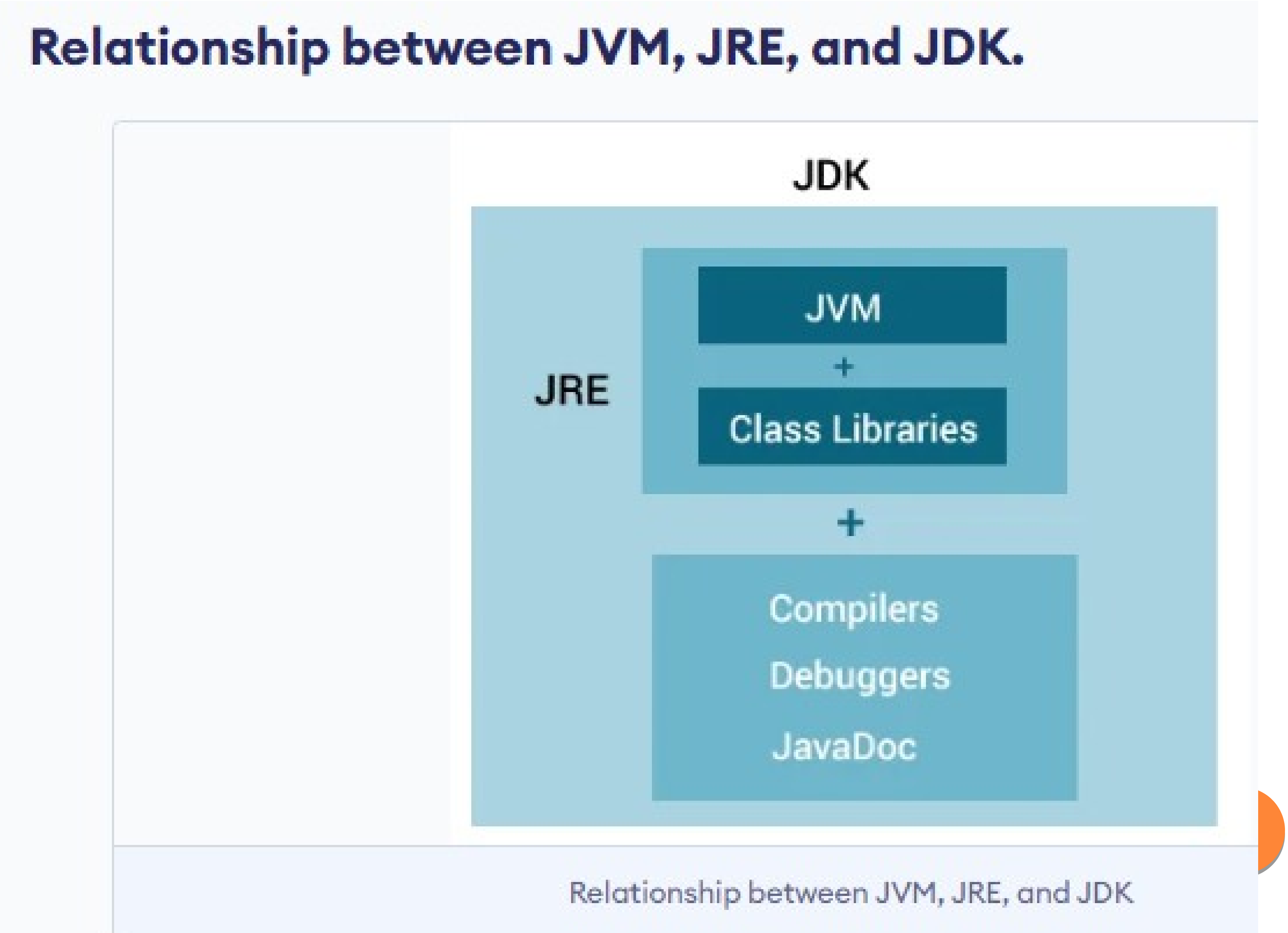
- JDK is an acronym for Java Development Kit.
- It contains JRE + development tools.
- Full featured Software Development Kit

- JDK Includes a complete JRE (Java Runtime Environment) plus tools for developing, debugging, and monitoring Java applications.

- JDK is required to develop and run Java applications and applets.

JRE / JDK

- Usually, when we only care about running Java programs on our browser or computer we will only install **JRE**. It's all we need.
- On the other hand, if we are planning to do some Java programming, we will also need **JDK**.



Relationship between JVM, JRE, and JDK.

Relationship between JVM, JRE, and JDK

HOW IS JAVA PLATFORM INDEPENDENT?

- In the case of Java, it is the magic of Bytecode that makes it platform independent.

- This adds to an important feature in the JAVA language termed as portability.
- Every system has its own JVM which gets installed automatically when the jdk software is installed.

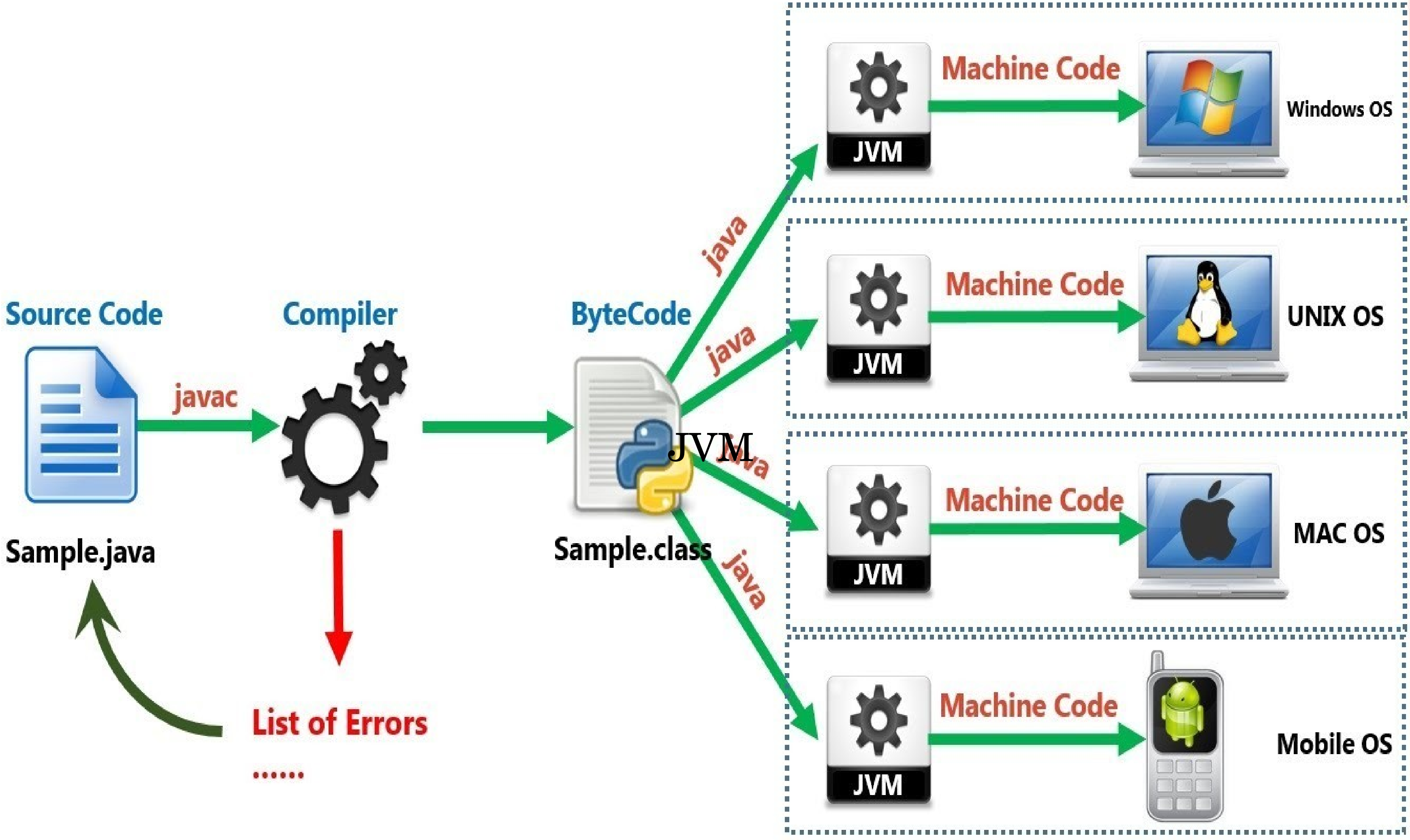
- For every operating system separate JVM is available which is capable to read the .class file or byte code.

- An important point to be noted is that while JAVA is platform-independent language, the JVM is platform-dependent.

- Different JVM is designed for different OS and byte code is able to run on different OS.

JVM

- Java is platform independent but JVM is platform dependent.
- In Java, the main point here is that the JVM depends on the operating system – so if you are running Mac OS X you will have a different JVM than if you are running Windows or some other operating system.

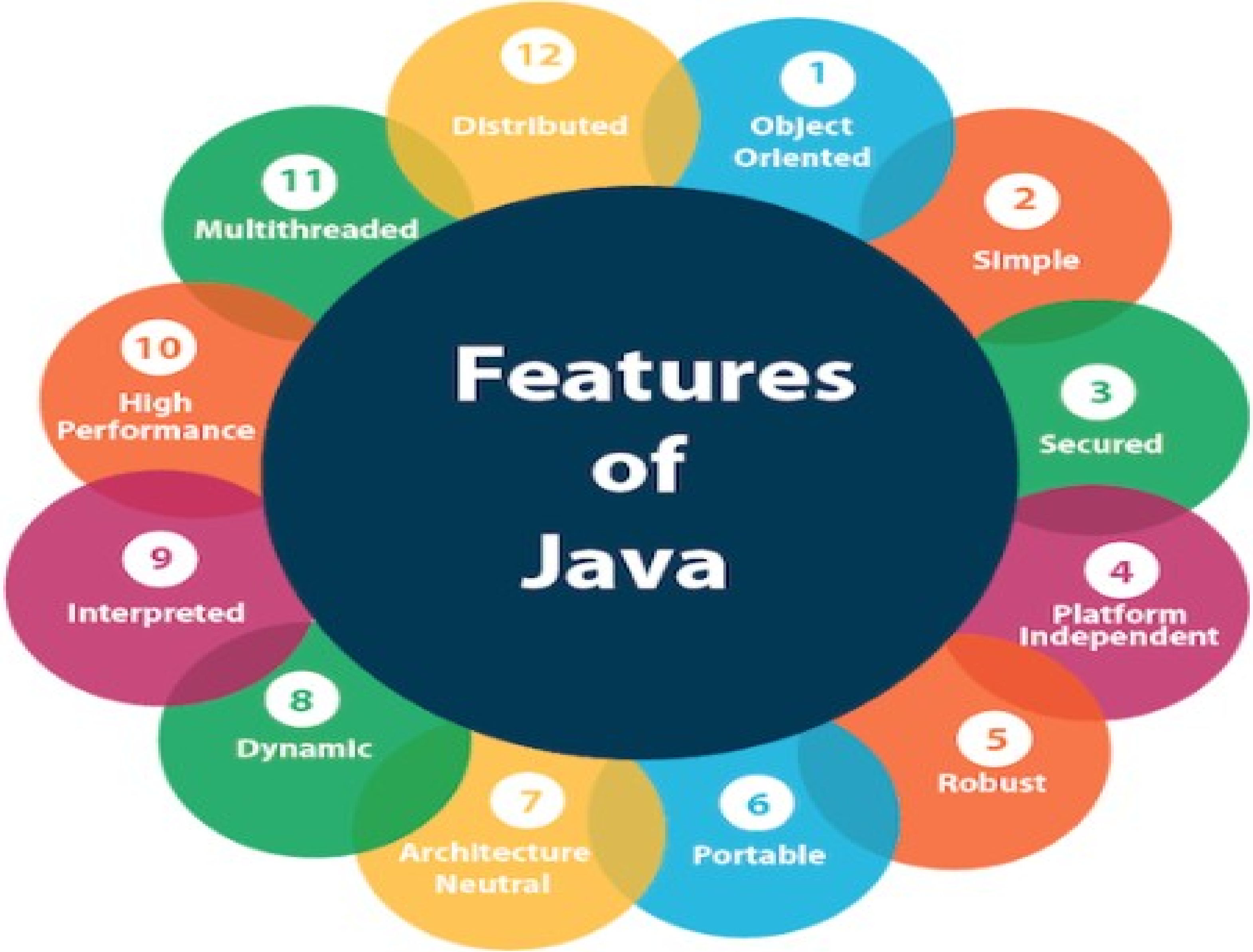

Source Code
Sample.java
javac
Compiler
List of Errors
......
ByteCode
Sample.class
java
java
java
java
JVM
Machine Code
Windows OS
JVM
Machine Code
UNIX OS
JVM
Machine Code
MAC OS
JVM
Machine Code
Mobile OS

JVM

WHAT IS JAVA?

- Java code that runs on one platform does not need to be recompiled to run on another platform, it's called **write once, run anywhere(WORA).**
- Java Virtual Machine (JVM) executes Java code but is written in platform-specific languages such as C/ C++ etc.
- JVM is not written in Java and hence cannot be platform independent and Java interpreter is actually a part of JVM.

FEATURES OF JAVA



FEATURES OF JAVA

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 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

Object-oriented

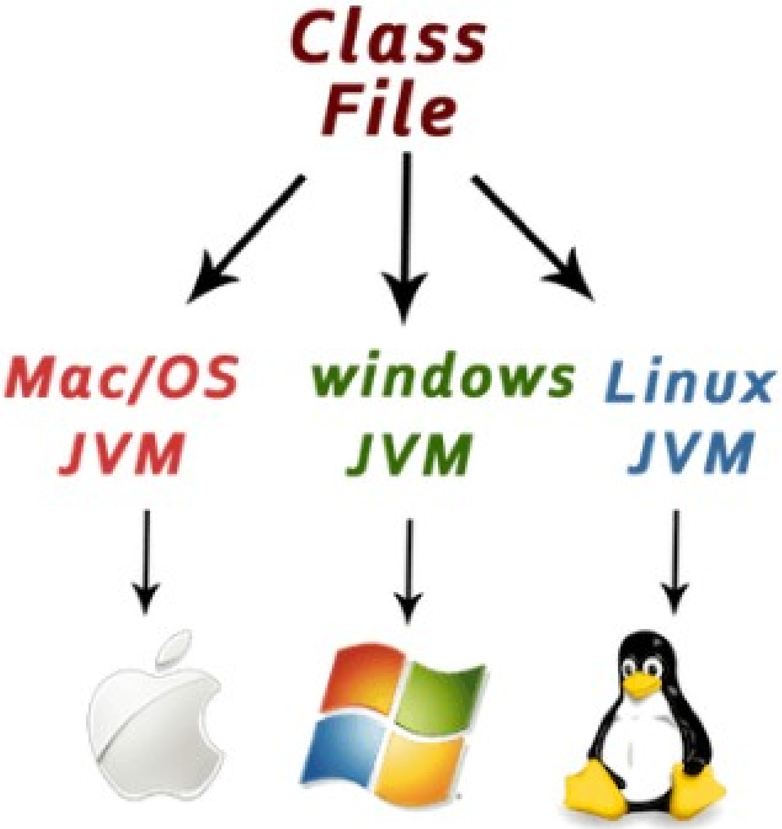
- Java is an object-oriented programming language.
- Everything in Java is an object.
- Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behavior.
- Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.

Basic concepts of OOPs are:

- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

FEATURES OF JAVA

Platform Independent



FEATURES OF JAVA

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 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.

- There are **exception handling and the type checking mechanism** in Java. All these points make Java robust.

Secured

- Java is best known for its security. Java is secured because:
- **No explicit pointer, Exception Handling, Compile-time checking, Memory management**
- JVM plays a vital role to provide security. It verifies the byte-code. The JVM provides guarantees that there is no unsafe operation going to execute.
- **Security API's**

  Java class libraries provide several API that leads to security. These APIs contain cryptographic algorithms and authentication protocols that lead to secure communication.
- **Byte Code**

  Every time when a user compiles the Java program, the Java compiler creates a class file with Bytecode, which are tested by the JVM at the time of program execution for viruses and other malicious files.

FEATURES OF JAVA

- **Portable:** Java does not have implementation dependent aspects and it yields or gives same result on any machine.
- **Architectural Neutral Language:** Java byte code is not machine dependent, it can run on any machine with any processor and with any OS.
- **High Performance:** Along with interpreter there will be JIT (Just In Time) compiler which enhances the speed of execution.
- **Multithreaded:** Executing different parts of program simultaneously is called multithreading. This is an essential feature to design server side programs.

JAVA FEATURES

- **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 designed for use on network; it has an extensive library which** works in agreement with TCP/IP

JAVA FEATURES

Dynamic

- Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment.
- Classes are stored in separate files and are loaded into the Java interpreter only when needed.
- This means that an application can decide as it is running what classes it needs and can load them when it needs them.

- 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++.