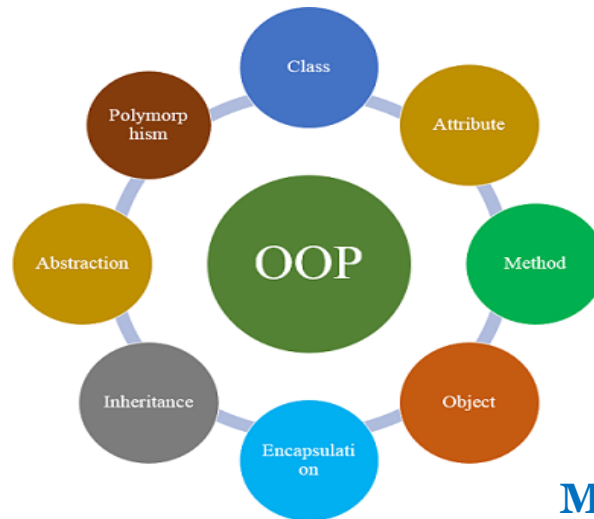


Chapter One

Introduction to Object-Oriented Programming



Mulugeta G. and Sufian K.

Definition of Programming

- **Programming**

- Is a **set of instructions** that instruct the machine how to perform a task and what to perform.
- Is a task to implement a solution to a problem in the form of computer language.
- It can be done using a variety of programming languages such as C, C++, C#, Python, Java, etc.

Programming Paradigms

- **Paradigm :**
 - A method to solve some problem or do some task.
- **Programming paradigm**
 - is an approach to solve problem using some programming language
 - it is a method to solve a problem using tools and techniques that are available to us following some approach.
- There are lots for programming language that are known but all of them need to follow some strategy when they are implemented and this **methodology/strategy** is paradigms.

Programming Paradigms

- Different programming languages follow different approaches to solve programming problems
- A programming paradigm is an approach to solve programming problems
- Common programming paradigms
 - ✓ Procedural Programming
 - ✓ Declarative Programming
 - ✓ Object-Oriented Programming

Procedural Programming

Procedural Programming

- Procedural programming follows a top-down approach during the designing of a program.
- It gives importance to the **concept of the function** and **divides the large programs into smaller** parts or called as **functions**.
- It follows a step-by-step approach in order to break down a task into a set of variables and routines via a sequence of instructions.
- Examples: ALGOL, COBOL, BASIC, PASCAL, FORTRAN, C.

Declarative Programming

Declarative Programming

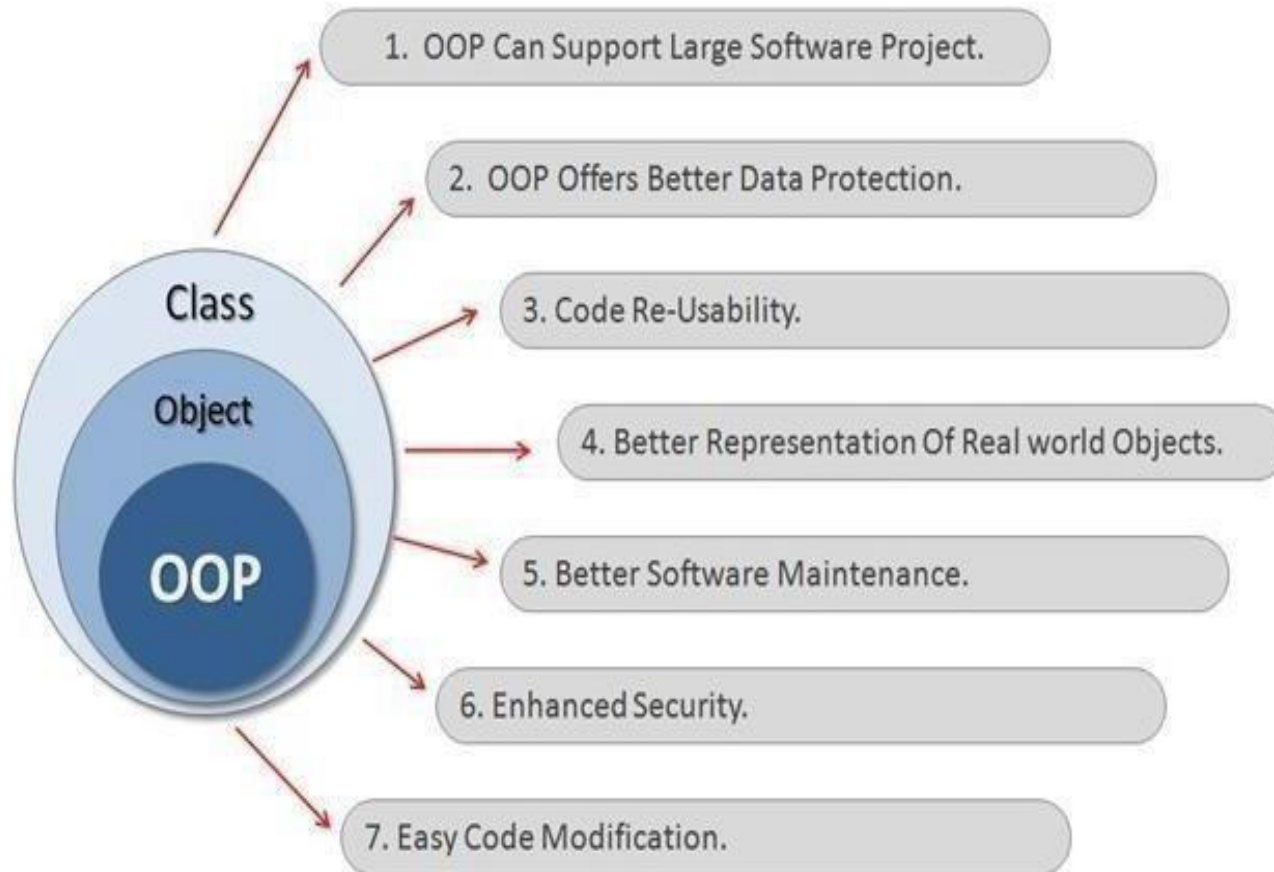
- Focus on what the program should accomplish without explicitly describing how to achieve it.
- You specify what you want to achieve, rather than how to achieve it.
- Focuses on describing the desired result.
- Example: prolog

Object-oriented Programming Paradigm

Object-oriented Programming Paradigm

- is based on the concept of object.
- An object contains data in the form of fields that are known as **attributes** and the procedures are known as **methods**.
- Programs are divided into what are known as objects.
- It follows the **bottom-up** flow of execution.
- It introduces concepts like data abstraction, inheritance, and overloading of functions and operator overloading.
- Example: Java, C++, Python, C#, Perl, Kotlin, Ruby

Advantages of OOP Paradigm



History of Programming

ALGO L

International group 1960

BCPL

Created by Martin Richards in 1967

B

Created by Ken Thompson in 1970

C

Created by Dennis Ritchie in 1972

C++

It is an Extension of C developed by Bjarne in 1980
Created by Dennis Ritchie in 1972

Java

A future of C++ and developed in 1991 by James Gosling

Introduction

- **Java:**

- Is a programming language and a platform
- Developed by **sun microsystems** [James Gosling]
- a high level, robust, secured and object-oriented programming language.
- Based on C/C++
- Have a Widespread acceptance

- **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.

History of Java

- Java started out as a research project.
- James Gosling is generally credited as the inventor of the Java programming language.
- He was the first designer of Java and implemented its original compiler and virtual machine.
- He is also known as the Father of Java.



James Gosling

Why “Java” name

- 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.
 - Java is an **island of Indonesia** where first coffee was produced (called **java coffee**).
 - Notice that Java is just a name not an acronym.

Java Logo



Java Version History

- There are many java versions that has been released. Current stable release of Java is Java SE 12.

-
- 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 SE 9 (21st September, 2017)....
 - Java SE 22 (19th March 2024)
-

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 machines such as media player, antivirus etc.
- AWT and Swing are used in java for creating standalone applications.

2. Web Application

- runs on the server side and used to create dynamic pages
- 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, JSP, Servlet and JPA are used for creating enterprise applications.

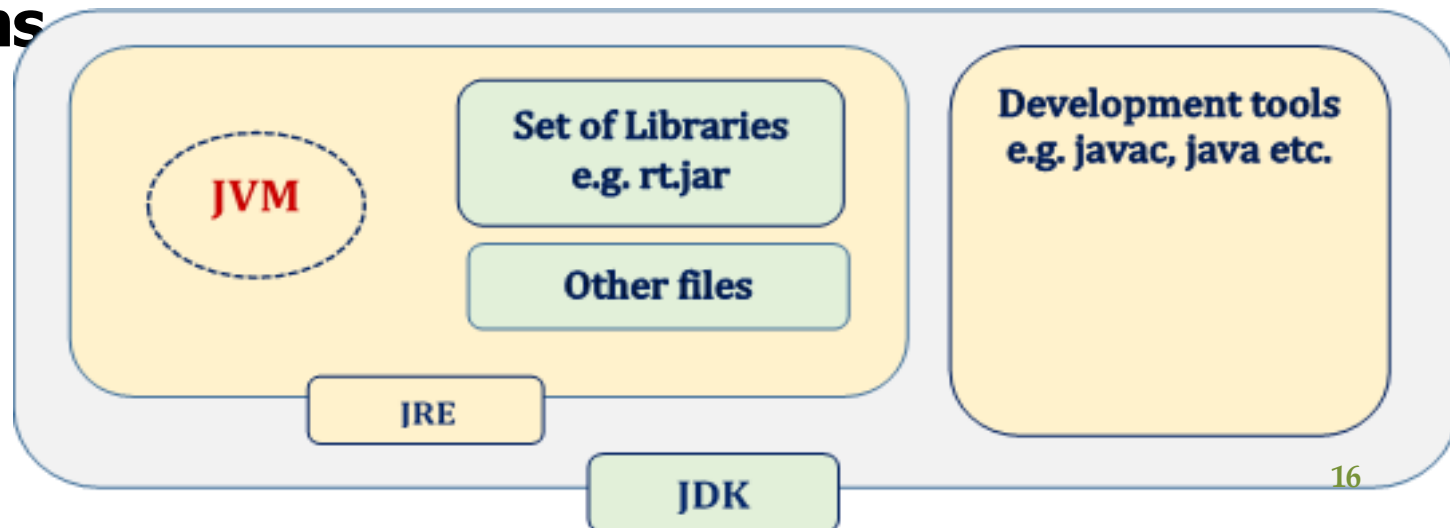
4. Mobile Application

- An application that is created for mobile devices.
- Currently Android and Java ME [Micro Edition] are used for creating mobile applications.

Java Terminology

Java Development Kit:

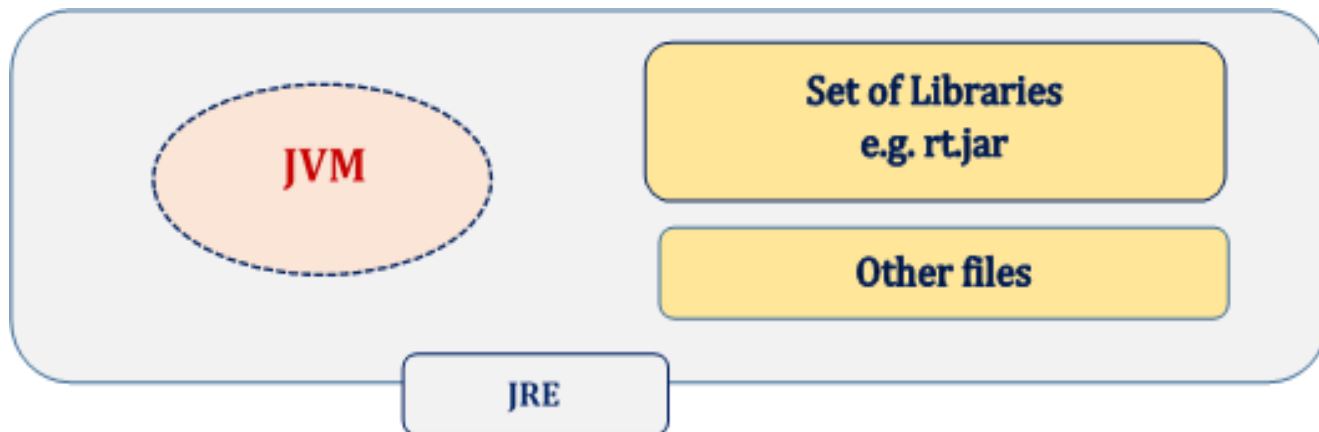
- It contains one(or more) **JRE's** along with the various development tools like
 - the Java source compilers,
 - bundling and deployment tools,
 - debuggers, development libraries, etc.
- Used to develop java **Application** and **Applets**
- complete development kit for writing and running Java programs



Cont'd...)

➤ Java Runtime Environment:

- It is used to provide runtime environment.
 - It is the implementation of JVM.
 - This is the software on your computer that actually runs Java programs.
- **JRE** = It contains **set of libraries** + **other files** that JVM uses at runtime.



Cont'd ...

- **Java Virtual Machine:**
 - is an abstract machine
 - It is a specification that provides runtime environment in which java bytecode can be executed.
 - It interprets the **byte code** into the **machine code** depending upon the underlying OS and hardware combination.
 - allows you to run Java applications on any device or operating system that has a compatible JVM installed.
- **The JVM performs following main tasks:**
 - Loads code
 - Verifies code
 - Executes code
 - Provides runtime environment

Differences b/w C++ and Java

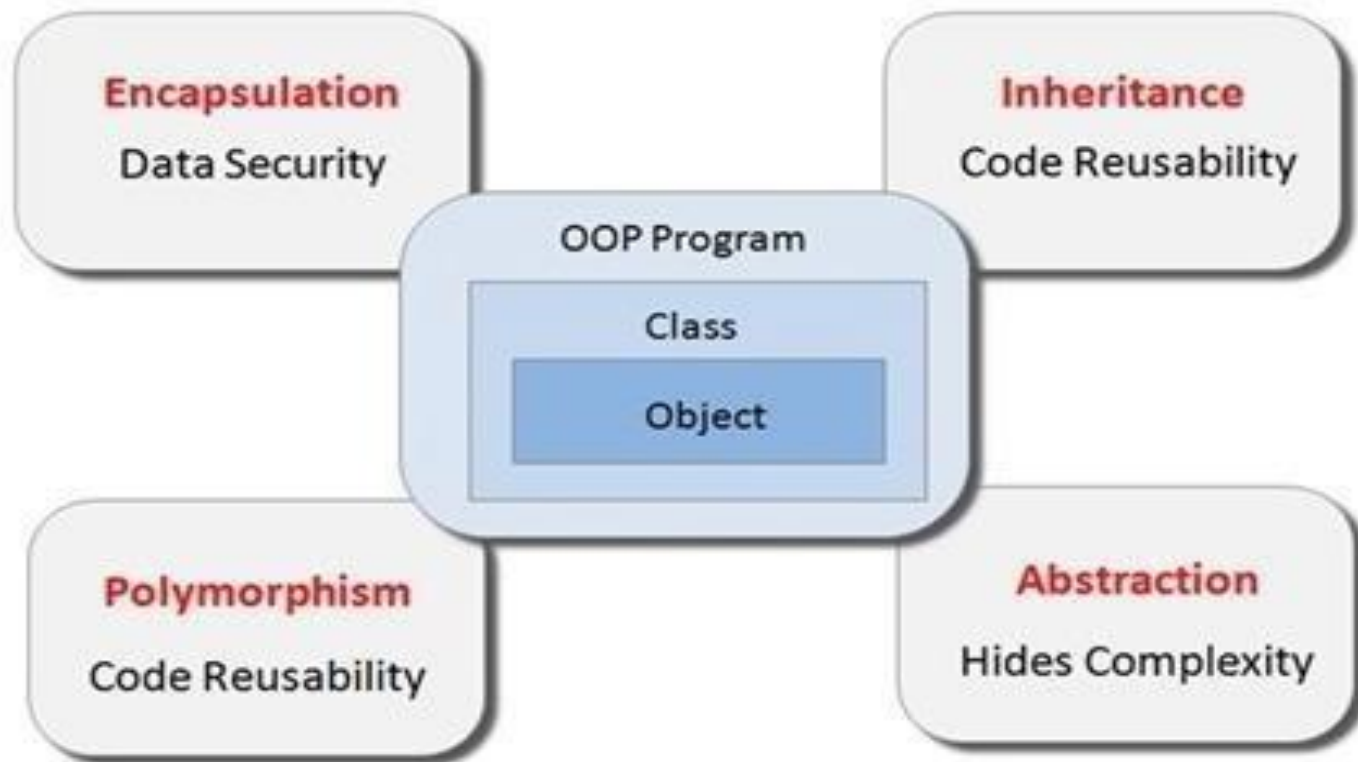
C++

1. **Global variable are supported.**
2. **Multiple inheritance is supported.**
3. **Constructors and Destructors supported.**
4. **In C++ pointers are supported.**
5. **C++ supporting ASCII character set.**

Java

1. **No Global variables. Everything must be inside the class only.**
2. **No direct multiple Inheritance.**
3. **Java supporting constructors only & instead of destructors garbage collection is supported.**
4. **No pointer arithmetic in Java.**
5. **Java supports Uni code Character set.**

Principles of Object-oriented Programming



Characteristics of java

- The following are characteristics of Java
 - Simple
 - Object-Oriented
 - Distributed
 - Interpreted
 - Robust
 - Platform Independent
 - Secure
 - Architecture-Neutral
 - Portable
 - High Performance
 - Multithreaded
 - Dynamic

Cont'd ...

- **Simple:**

According to Sun, Java language is simple because:

- Syntax is based on C++ (so easier for programmers to learn it after C++).
- Removed many confusing and/or rarely-used features e.g., explicit pointers, operator overloading etc.
- No need to remove unreferenced objects because there is Automatic Garbage Collection in java.

- **Object-Oriented:**

- OOP is a popular programming approach that is replacing traditional procedural programming techniques.
- OOP is a methodology that simplify software development and maintenance
- OOP provides great **flexibility, modularity, clarity, and reusability** through encapsulation, inheritance, and polymorphism.
- Basic concepts of OOP are: object, class, inheritance, polymorphism, abstraction and encapsulation

Cont'd ...

- **Distributed:**

- Distributed computing involves several computers working together on a **network**.
- Since **networking capability** is inherently integrated into Java, writing network programs is like **sending** and **receiving** data to and from a file.
- **RMI** and **EJB** are used for creating distributed applications by **calling the methods** from any machine on the internet

- **Interpreted:**

- The programs are compiled into the Java Virtual Machine code called **bytecode**.
- The bytecode is **machine-independent** and can run on any machine that has a **Java interpreter**, which is part of the Java Virtual Machine (**JVM**).

Cont'd ...

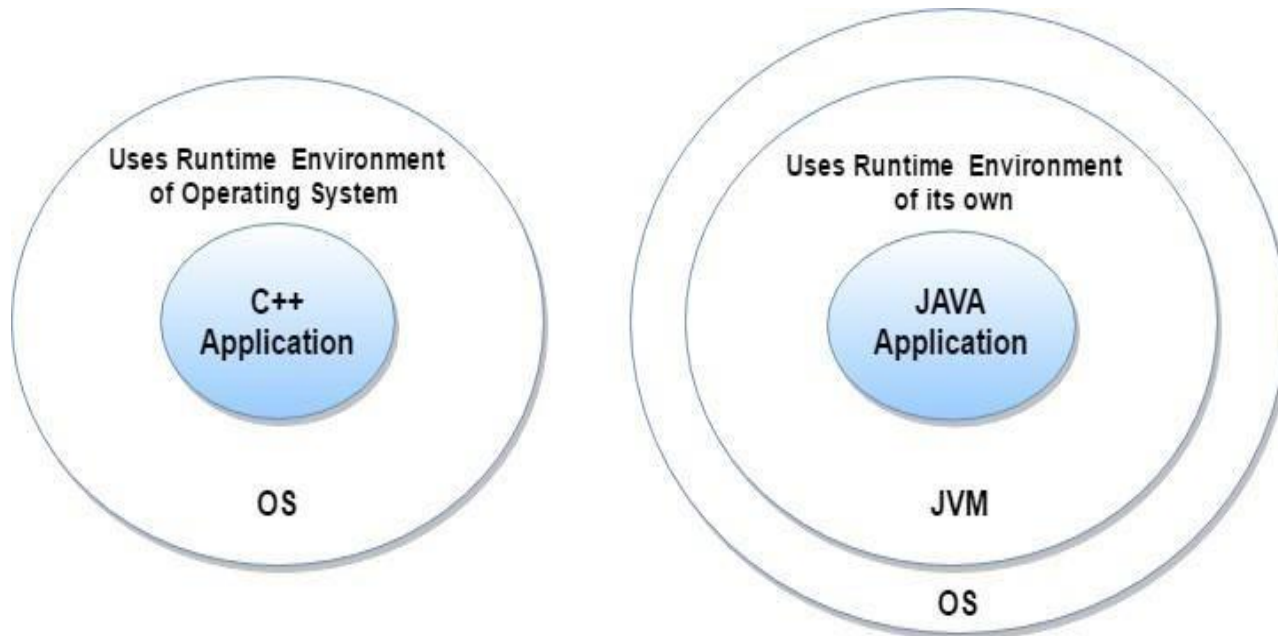
- **Robust:**
 - Robust simply means **strong**.
 - Java uses strong **memory management**.
 - There are **lack of pointers** that avoids security problem.
 - There is automatic garbage collection in java.
 - There is **exception handling** and **type checking mechanism** in java.
 - All these points makes **java robust**.

Cont'd ...

- **Secure:**

Java is secured because:

- **No explicit pointer**
- **Java Programs run inside virtual machine sandbox**
- **Java implements several security mechanisms to protect your system against harm caused by stray programs.**



Cont'd ...

- **Architecture neutral:**

- Write once, run anywhere
- With a Java Virtual Machine (JVM), you can write one program that will run on any platform.
- 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 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.

- **Portable:**

- We may carry the java bytecode to any platform.
- They can be run on any platform without being recompiled.

- **High Performance:**

- Java is faster than traditional interpretation since byte code is "close" to native code.

Cont'd ...

- **Multithreaded:**

- 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.
- important for multi-media, Web applications etc.
- Multithread programming is smoothly integrated in Java, whereas in other languages you have to call procedures specific to the operating system to enable multithreading.

Cont'd ...

- **Dynamic:**

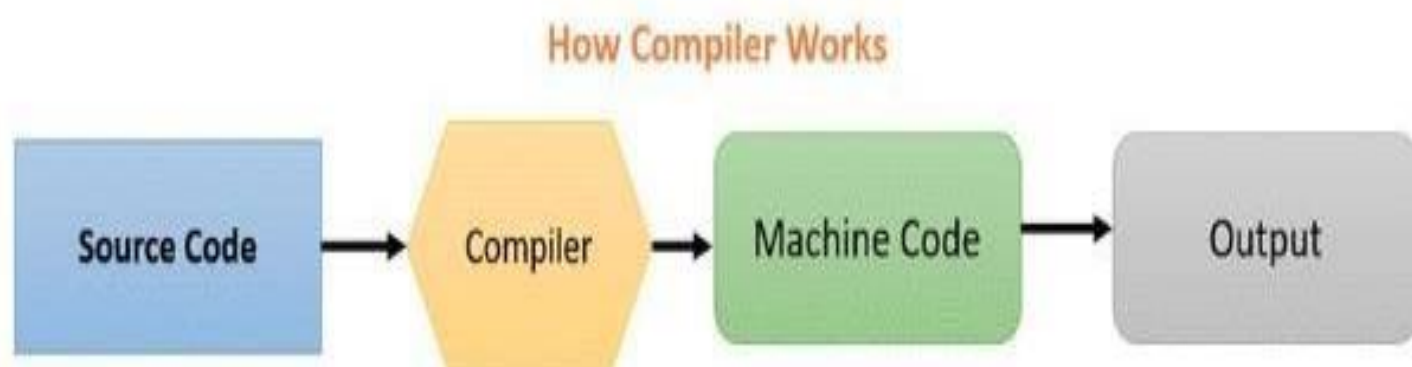
- **Java was designed to adapt to an evolving environment.**
- **New code can be loaded on the fly without recompilation.**
- **There is no need for developers to create, and for users to install, major new software versions.**
- **New features can be incorporated transparently as needed.**

The Programming Process

- **Computers do not understand human languages.**
- **In fact, at the lowest level, computers only understand sequences of numbers that represent operational codes (op codes for short).**
- **Programming languages are for humans to read and understand.**
- **The program (source code) must be translated into machine language so that the computer can execute the program (as the computer only understands machine language).**
- **The way that this translation occurs depends on whether the programming language is a compiled language or an interpreted language.**

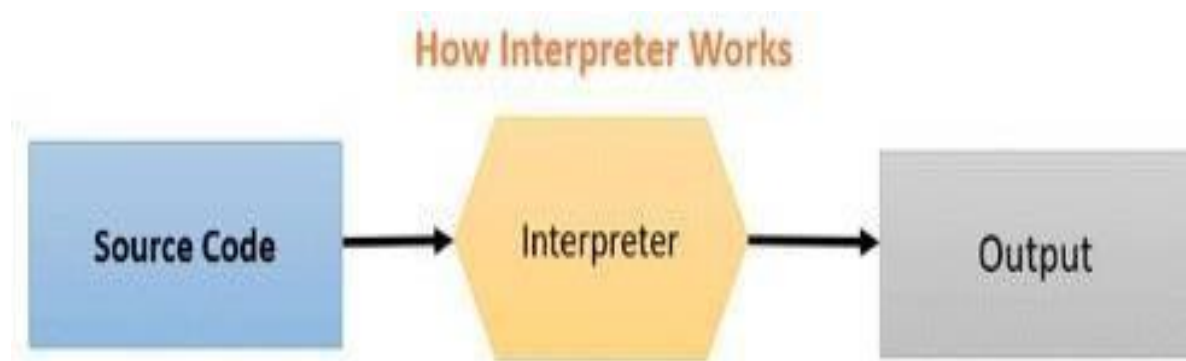
Compiler

- to translate Java source code into bytecode that the JVM can understand and execute.
- It translates the entire source code of a program into machine code or bytecode all at once, before execution starts.
- The compiler checks your Java code for syntax errors (incorrect code structure) during the compilation process.

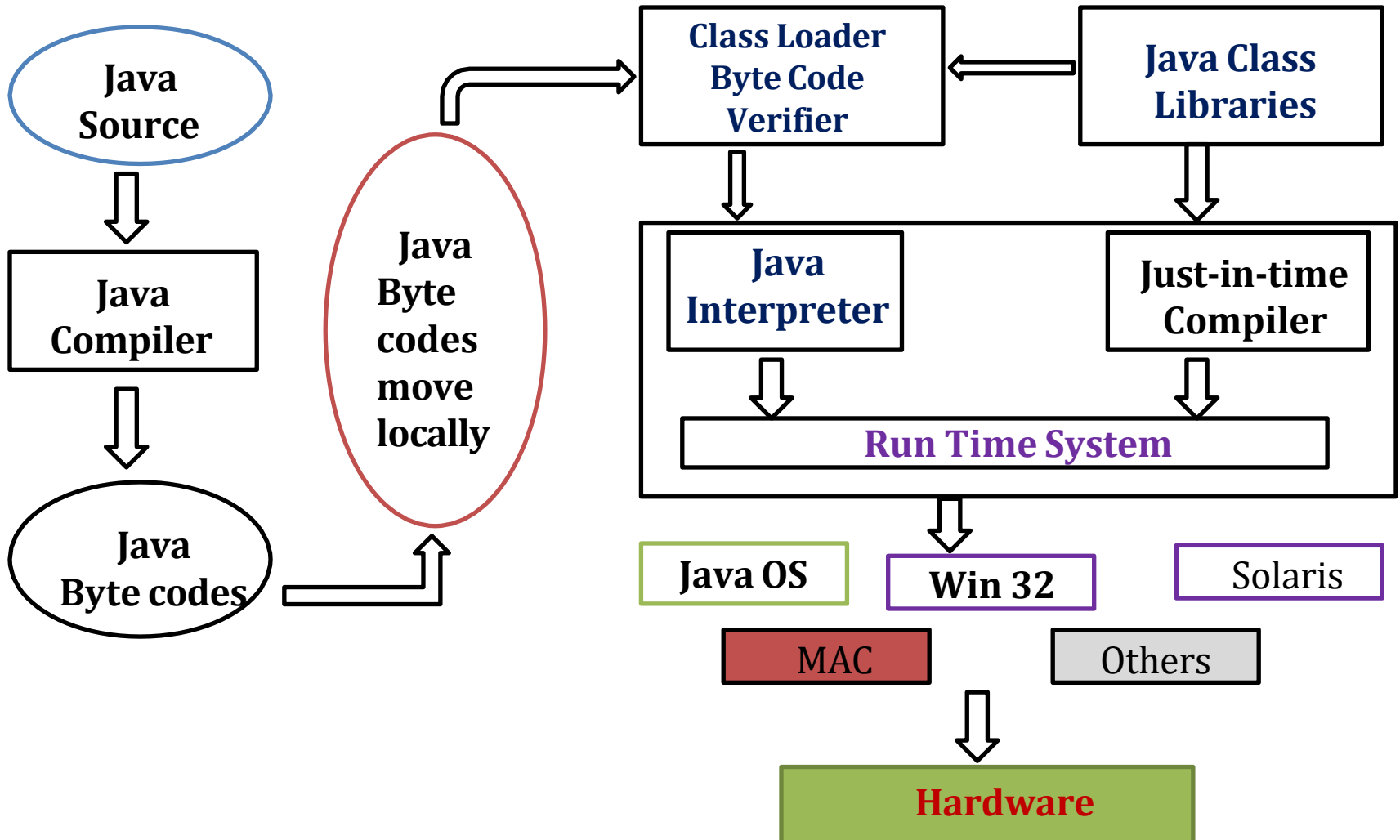


Interpreter

- **It translates and executes programming language statements one by one.**
- **An interpreter takes less time to interpret a source program as distinguished by a compiler.**
- **Instead of generating a separate output file, it reads and executes the program directly.**



Java execution procedure



Editors

- Java programs can be written using a **simple text-editor**.
- We can also use Java integrated development environment
- An **IDE** provides many functionalities such as auto-completion, debugger options,
- **Advantages of Using an IDE or Editors**
 - ❖ Require less time and efficient
 - ❖ Built-in documentation
 - ❖ Helps to create a company standard:
- **Popular Java Editors**
 - NetBeans, Eclipse, IntelliJ IDEA, BlueJ

Simple Java program

//This program prints Welcome to Java!

```
public class Welcome {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java!");  
    }  
}
```

Output:

Welcome to Java!

Thank You

