**JAVA**

Microprocessor revolution enabled development of PCs

Impact in “Internet of Things”

**Sun Microsystems 1991**

Funded project led by **James Gosling**

C++ based object-oriented programming

Named **Java**

Can run on variety of computer systems

“Write once, run anywhere”

Garnered interest in businesses because of the **Internet**

Used to develop large-scale enterprise applications

* Enhance functionality of web servers
* Develop robotic software
* Application for consumer devices

Key language for developing Android devices

Sun Microsystems acquired by **Oracle in 2010**

Most used programming language for general use

10 million developers

Existing collections of classes and methods in the Java class library, aka Java API (Application Programming Interfaces)

**Java Dev Environment**

Five steps to create and execute Java apps

* Edit
* Compile
* Load
* Verify
* Execute

**Phase 1: Creating a Program**

Editing a file with an editor program

You type a Java program and then save on a storage device

Given a name ending with .java extension (indicating Java source code)

Editors:

* Linux, vi and emacs
* Windows, notepad
* MacOS, TextEdit

**Integrated development environments (IDEs)**

Provides tools that support software development

* Eclipse
* IntelliJ IDEA
* NetBeans

**Phase 2: Compiling a Java Program into Bytecodes**

Compile a program with command **javac**

Done in your OS terminal

* Linux, Shell (Terminal)
* Windows, Command Prompt
* MacOS, Terminal

Creating a file ending with .class

Java compiler translates source code into **bytecodes**

Represents the task in execution phase (phase 5)

**Virtual Machine (VM)**

Software application that simulates a computer (hides the underlying OS)

Largely utilized VMs: **Java Virtual Machine (JVM) Microsoft’s .NET**

Bytecode instructions are platform independent

Makes Java bytecodes portable on any platform containing a JVM

**Phase 3: Loading a Program into Memory**

JVM places the program in memory to execute it

Known as **loading**

**Class loader**, grabs .class to transfer to primary memory

.class can be loaded locally or through cloud

**Phase 4: Bytecode Verification**

Loaded classes get verified by **bytecode verifier**

Checks if they are valid

Java enforces strong security to protect from harmful files (viruses)

**Phase 5: Execution**

JVM **executes** the bytecodes for specified action

JVM was previously just an interpreter

Execute one bytecode at a time

Today using a combination of interpretation and **just-in-time (JIT)** compilation

Analyses and interpretates simultaneously, searching for hot spots

Oracle’s **Java Hotspot** compiler translates into machine language

JVM encounters compiled files, faster machine language code executes

Going through two compilation phases

* Translated into bytecodes for portability
* During execution translated into machine language