



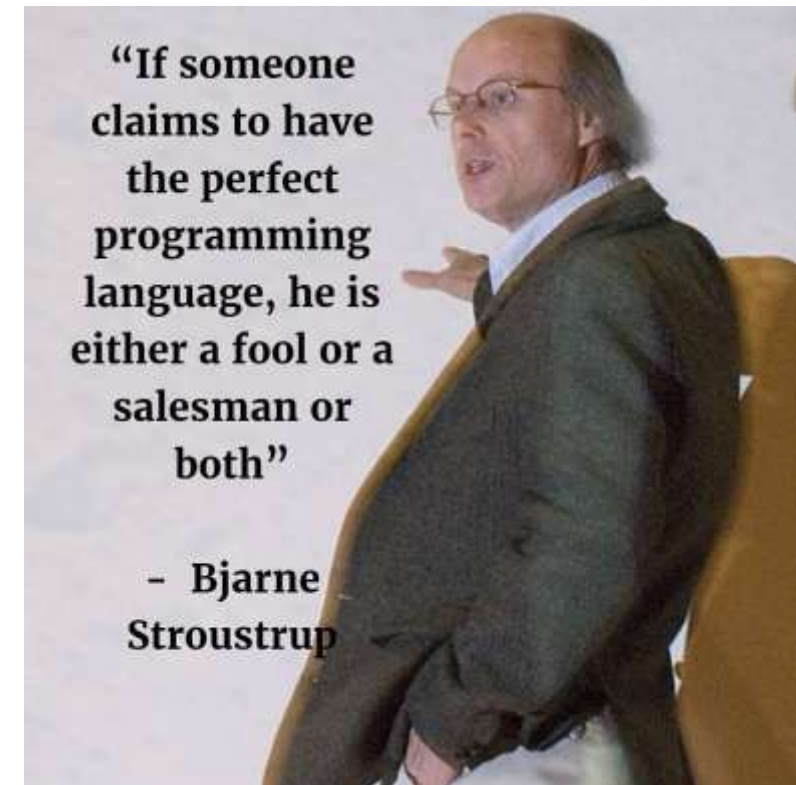
Introduction of Java

By Er. Suman Bhandari

What is programming Language?

Stroustrup (Danish computer scientist) **[1994]** says that **a programming languages is:**

- a tool for instructing machines
- a means of communicating between programmers
- a vehicle for expressing high-level designs
- a notation for algorithms
- a way of expressing relationships between concepts
- a tool for experimentation
- a means for controlling computerized devices
- Programming languages are notations, used for specifying, organizing, and reasoning about computations.



Different Programming Paradigm

- Imperative - C
- Functional - LISP
- Object Oriented - C++, smalltalk, Java, simula
- Logic - Prolog

Content

- What is Java ?
- Where is Java used ?
- History of Java
- Features of Java
- Disadvantage of Java
- Java Development Kit (JDK)
- Java Compiler
- Java Virtual Machine (JVM)
- Java Program-Development phase
- Portability of Java
- Simple Java Program
- Java Development Tools
- What is Object Oriented Programming

What is Java ?

- Java is relatively high-level language, class-based and Object-oriented Programming Language.
- Java was developed by **James Gosling** at Sun Microsystems.
- Java is a first programming language which provide the concept of writing programs that can be executed using the web.



James Arthur Gosling, often referred to as "Dr. Java", is a Canadian computer scientist, best known as the founder and lead designer behind the Java programming language.

Where Is Java Used ?

- 3 Billion devices run java.
- There are many devices where Java is currently used.
- Java is full-featured and general-purpose programming language that is capable for developing a robust mission-critical application for:
 - **Desktop Applications** - Media player, Antiviruses etc.
 - **Servers**
 - **Mobile Devices**
 - **Web Applications**
 - **Enterprise Application – Banking Application, Business Application**
 - **Embedded System**
 - **Games**
 - **Robotics**
- Java running on the desktop is called ***Application***
- Java running on the Internet is called ***Applets***
- Java running on the server-side is called ***Servlet***

James Gosling

- 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**.



Brief History of Java

- In 1990, Sun Microsystems began an internal project known as the ***Green Project*** to work on a new technology.
- In 1992, the Green Project was spun off and its interest directed toward building highly interactive devices for the cable TV industry.
- In 1994, the focus of the original team was re-targeted, this time to the use of Internet technology. A small web browser called ***HotJava*** was written.
- ***Oak*** was renamed to ***Java*** after learning that Oak had already been trademarked.

- In 1995, Java was first publicly released.
- In 1996, Java Development Kit (**JDK**) 1.0 was released.
- In 2002, JDK 1.4 (codename *Merlin*) was released, the most widely used version.
- In 2004, JDK 5.0 (codename *Tiger*) was released, the latest version.
- There are the JDK version list and its release date.

Version Name	Code Name	Release Date
JDK 1.0	Oak	January 1996
JDK 1.1	(none)	February 1997
J2SE 1.2	Playground	December 1998
J2SE 1.3	Kestrel	May 2000
J2SE 1.4	Merlin	February 2002
J2SE 5.0	Tiger	September 2004
JDK 6	Mustang	December 2006
JDK 7	Dolphin	July 2011
JDK 8		March 2014
JDK 9		September, 21st 2017
JDK 10		March, 20th 2018
JDK 11		September, 25th 2018
JDK 12		March, 19th 2019
JDK 13		September, 10th 2019
JDK 14		March, 17th 2020
JDK 15		September, 15th 2020
JDK 16		March, 16th 2021

Features of Java

- Java is Simple.
- Java is Object Oriented
- Java is Distributed
- Java is Architecture Neutral
- Java is Robust
- Java is Interpreted and Compiled
- Java is Secure
- Java is Portable
- Java is Multi-Threaded
- Java Is Dynamic

Features of Java

- Java is Simple
- Java is Object Oriented
- Java is Distributed
- Java is Architecture Neutral
- Java is Robust
- Java is Interpreted
- Java is Secure
- Java is Portable
- Java is Multi-Threaded
- Java Is Dynamic

Java is partially modeled on C++, but greatly simplified and improved. Some people refer to Java as "C++--" because it is like C++ but with more functionality and fewer negative aspects.

Features of Java

- Java is Simple
- Java is Object Oriented
- Java is Distributed
- Java is Architecture Neutral
- Java is Robust
- Java is Interpreted
- Java is Secure
- Java is Portable
- Java is Multi-Threaded
- Java Is Dynamic

Java was designed from the start to be object-oriented. Object-oriented programming (OOP) is a popular programming approach that is replacing traditional procedural programming techniques.

Object-oriented programming provides great flexibility, modularity, clarity, and reusability through encapsulation, inheritance, and polymorphism.

Features of Java

- Java is Simple
- Java is Object Oriented
- Java is Distributed
- Java is Architecture Neutral
- Java is Robust
- Java is Interpreted
- Java is Secure
- Java is Portable
- Java is Multi-Threaded
- Java Is Dynamic

Distributed computing involves several computers working together on a network.

Java is designed to make distributed computing easy. Since networking capability is inherently integrated into Java, writing network programs is like sending and receiving data to and from a file.

Features of Java

- Java is Simple
- Java is Object Oriented
- Java is Distributed
- **Java is Architecture Neutral**
- Java is Robust
- Java is Interpreted
- Java is Secure
- Java is Portable
- Java is Multi-Threaded
- Java Is Dynamic

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

Features of Java

- Java is Simple
- Java is Object Oriented
- Java is Distributed
- Java is Architecture Neutral
- **Java is Robust**
- Java is Interpreted
- Java is Secure
- Java is Portable
- Java is Multi-Threaded
- Java Is Dynamic

Java compilers can detect many problems that would first show up at execution time in other languages.

Java has a runtime exception-handling feature to provide programming support for robustness.

Features of Java

- Java is Simple
- Java is Object Oriented
- Java is Distributed
- Java is Architecture Neutral
- Java is Robust
- **Java is Interpreted**
- Java is Secure
- Java is Portable
- Java is Multi-Threaded
- Java Is Dynamic

You need an interpreter to run Java programs. 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).

Features of Java

- Java is Simple
- Java is Object Oriented
- Java is Distributed
- Java is Architecture Neutral
- Java is Robust
- Java is Interpreted
- **Java is Secure**
- Java is Portable
- Java is Multi-Threaded
- Java Is Dynamic

Java implements several security mechanisms to protect your system against harm caused by stray programs.

Features of Java

- Java is Simple
- Java is Object Oriented
- Java is Distributed
- Java is Architecture Neutral
- Java is Robust
- Java is Interpreted
- Java is Secure
- **Java is Portable**
- Java is Multi-Threaded
- Java Is Dynamic

Because Java is architecture neutral, Java programs are portable.

They can be run on any platform without being recompiled.

Features of Java

- Java is Simple
- Java is Object Oriented
- Java is Distributed
- Java is Architecture Neutral
- Java is Robust
- Java is Interpreted
- Java is Secure
- Java is Portable
- Java is Multi-Threaded
- Java Is Dynamic

Multithread programming is smoothly integrated in Java, where as in other languages you have to call procedures specific to the operating system to enable multithreading.

Features of Java

- Java is Simple
- Java is Object Oriented
- Java is Distributed
- Java is Architecture Neutral
- Java is Robust
- Java is Interpreted
- Java is Secure
- Java is Portable
- Java is Multi-Threaded
- Java Is 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.

Disadvantages of Java

- Slower than compiled language such as **C**
- an experiment in 1999 showed that Java was 3 or 4 times slower than **C** or **C++**
- Since Java Programs run on top of **JVM**, it consumes more memory.

Java Platform Edition

- A Java Platform is the set of APIs, class libraries, and other programs used in developing Java programs for specific applications.

There are 3 Java Platform Editions

1. Java 2 Platform, Standard Edition (J2SE)

- Core Java Platform targeting applications running on workstations

2. Java 2 Platform, Enterprise Edition (J2EE)

- Component-based approach to developing distributed, multi-tier enterprise applications

3. Java 2 Platform, Micro Edition (J2ME)

- Targeted at small, stand-alone or connectable consumer and embedded devices

Java Development Kit (JDK)

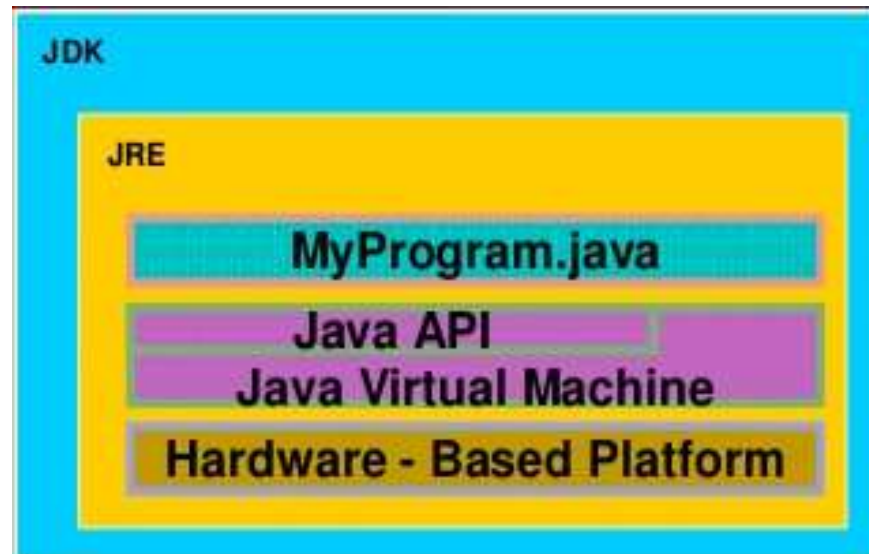
Java Development Kit (JDK)

- Is a set of Java tools for developing Java programs
- Consists of Java API, Java Compiler, and JVM

Java Application Programming Interface (API)

- Is prewritten code, organized into packages of similar topics

Java Virtual Machine (JVM) is an execution engine that runs compiled Java byte code.



Java Compiler (Byte Code)

- The Java compiler translates Java programs into byte-code.
- Byte-code creates and store on disk as *.class* file
- Byte-code verifier verify the validation and check whether it violate java's security or not.
- Once compiled to byte-code, a Java program can be used on any computer, making it very portable machine.

Java Virtual Machine (JVM)

- Java Virtual Machine is the like usual computer which translate high level language into machine language.
- Just like that Java virtual machine also translate Bytecode into machine language.
- JVM are available for many hardware and software Platform.

Java Program Development Phase

Java programs normally undergo four phases

1. Edit

- Programmer writes program (and stores program on disk)

2. Compile

- Compiler creates byte-codes from program (.class)

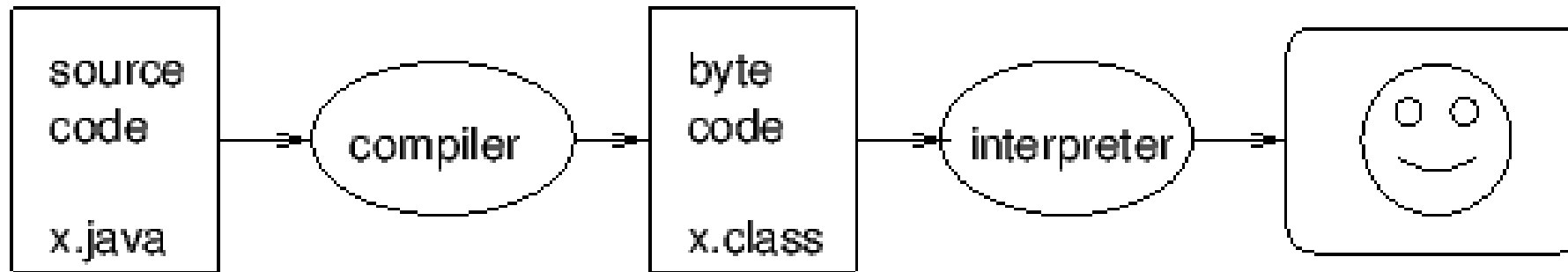
3. Load

- Class loader stores byte-codes in memory

4. Execute

- **Interpreter:** translates byte-codes into machine language

Cont...



The compiler
reads the
source code...

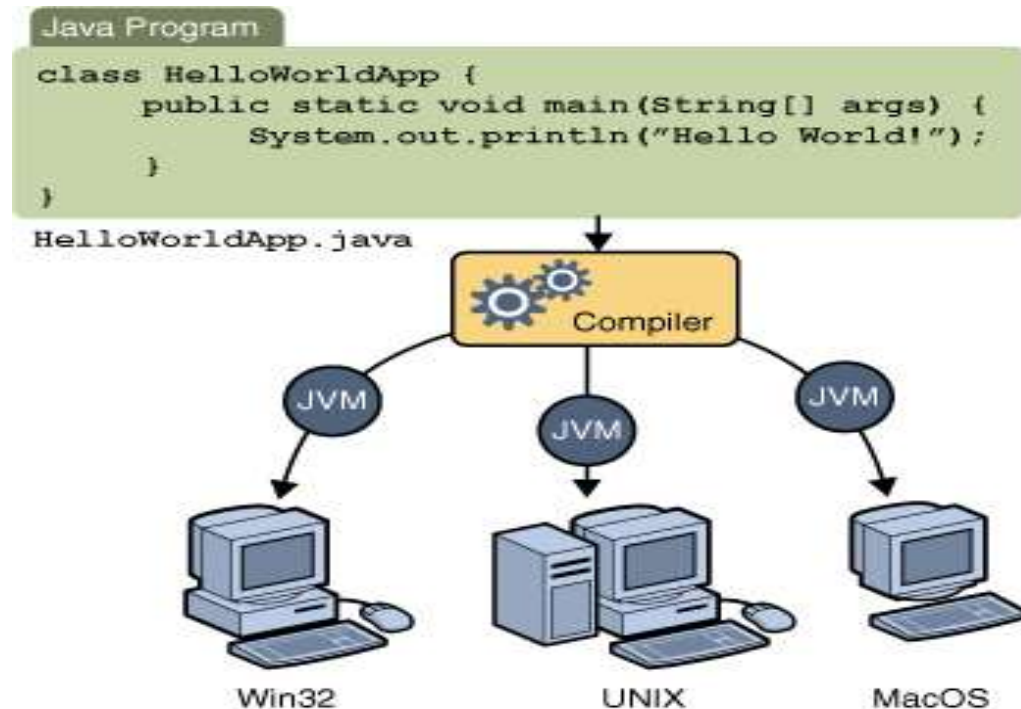
... and generates
Java byte code.

A Java interpreter
reads the byte
code...

... and the result
appears on
the screen.

Portability of Java

- Portable means that a program may be written on one type of computer and then run on a wide variety of computers, with little or no modification.
- Java byte code runs on the JVM and not on any particular CPU; therefore, compiled programs are highly portable.
- JVMs exist on many platforms:
 - Windows
 - Mac
 - Linux



A Simple Java Program

//This program prints Welcome to Java!

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

- **Class definition:** This line uses the keyword **class** to declare that a new class is being defined.

class Welcome

- **Main Method:** In Java programming language, every application must contain a main method whose signature is:

public static void main(String[] args)

- **public:** So that JVM can execute the method from anywhere.
 - **static:** Main method is to be called without object. The modifiers public and static can be written in either order.
 - **void:** The main method doesn't return anything.
 - **main():** Name configured in the JVM.
 - **String[]:** The main method accepts a single argument: an array of elements of type String.
- The next line of code is shown here. Notice that it occurs inside **main()**

System.out.println("Hello, World");

Output is actually accomplished by the built-in `println()` method. **System** is a predefined class that provides access to the system, and **out** is the variable of type output stream that is connected to the console.

- **Comments:** They can either be multi-line or single line comments.
- */* This is multiline comment */*
- *// This single line comment*

Java Development Tools

- A software that provides an **integrated development environment** (IDE) for rapidly developing Java programs.
- Java development tools on the market:
 - NetBeans by Sun (open source)
 - IntelliJ IDEA by JetBrains
 - Eclipse by IBM (open source)
- Other useful tools:
 - TextPad Editor
 - Visual Studio Code
 - Jedit

What is Object Oriented Programming?

- A language in which everything represent in the form of Object is called Object Oriented Programming Language.
- It not only makes the program less complex but also makes the software reuse feasible and possible.
- All object-oriented programming languages provide mechanisms that help you implement the object- oriented model.
- Principles of OOP:
 - Encapsulation
 - Inheritance
 - Polymorphism
 - Abstraction

Procedure Oriented Vs Object Oriented

S.N	Procedural Oriented Programming	Object Oriented Programming
1	In procedural programming, program is divided into small parts called functions .	In object oriented programming, program is divided into small parts called objects .
2	Procedural programming follows top down approach .	Object oriented programming follows bottom up approach .
3	There is no access specifier in procedural programming.	Object oriented programming have access specifiers like private, public, protected etc.
4	Adding new data and function is not easy.	Adding new data and function is easy.
5	Procedural programming does not have any proper way for hiding data so it is less secure .	Object oriented programming provides data hiding so it is more secure .
6	In procedural programming, overloading is not possible.	Overloading is possible in object oriented programming.
7	In procedural programming, function is more important than data.	In object oriented programming, data is more important than function.
8	Procedural programming is based on unreal world .	Object oriented programming is based on real world .
9	Examples: C, FORTRAN, Pascal, Basic etc.	Examples: C++, Java, Python, C# etc.

Handling Common Errors:

- When executing Java code, different errors can occur: coding errors made by the programmer, errors due to wrong input, or other unforeseeable things.
- When an error occurs, Java will normally stop and generate an error message. The technical term for this is: Java will throw an **exception** (throw an error).

Java try and catch

- The **try** statement allows you to define a block of code to be tested for errors while it is being executed.
- The **catch** Statement allows you to define a block of code to be executed, if an error occurs in the try block.
- The **try** and **catch** keywords come in pairs.

Syntax

```
try {  
    // Block of code to try  
}  
catch(Exception e) {  
    // Block of code to handle errors  
}
```

Example of Simple error Handling:

```
public class Main {  
    public static void main(String[] args) {  
        try {  
            int[] myNumbers = {1, 2, 3};  
            System.out.println(myNumbers[10]);  
        } catch (Exception e) {  
            System.out.println("Something went wrong.");  
        }  
    }  
}
```

Assignment

What is object oriented programming? Differentiate between procedural vs object oriented programming. What is error handling?

Next Class

What are the Data Types in Java?