Lecture # 1

Imran Ahsan

Java Introduction

- Java is a programming language and a platform.
- Java is an object oriented, high level programming language. It was originally developed by Sun Microsystem and later acquired by Oracle. It is one of the most secured language.
- Java is a platform independent language. So, it can be run on multiple platforms like Windows, Linux, Sun Solaris, Mac/OS etc. once it is converted in bytecode. Java provides a multithreaded environment that makes its performance better.

Types of Java Applications

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. 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.

Java Platforms / Editions

1) **Java SE (Java Standard Edition)**

- It is a java programming platform. It includes Java programming APIs such as java.lang, java.io, java.net, java.util, java.sql, java.math etc. It includes core topics like OOPs, String, Regex, Exception, Inner classes, Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection etc.

2) **Java EE (Java Enterprise Edition)**

- It is an enterprise platform which is mainly used to develop web and enterprise applications. It is built on the top of Java SE platform. It includes topics like Servlet, JSP, Web Services, EJB, JPA etc.

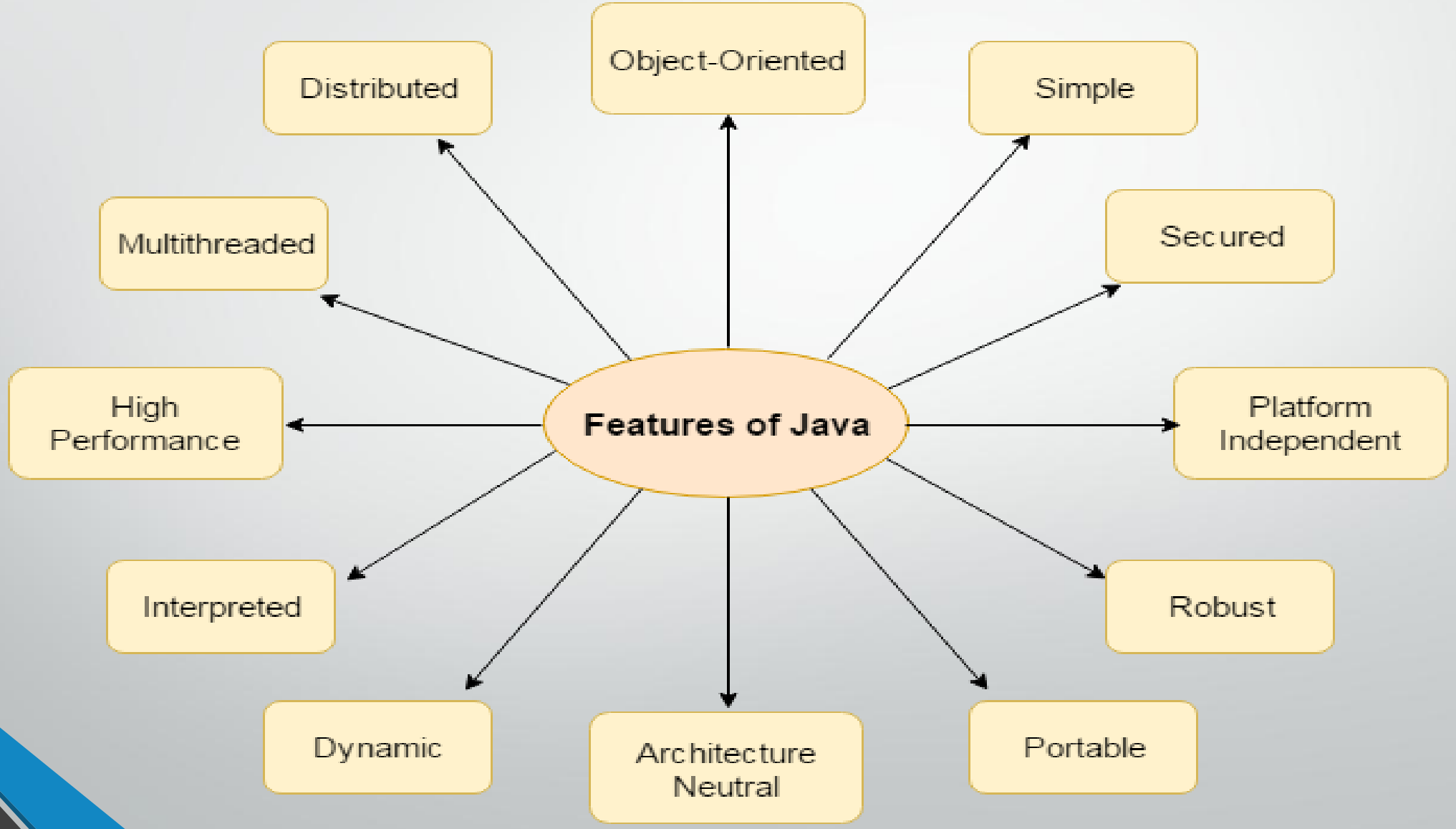
3) **Java ME (Java Micro Edition)**

- It is a micro platform which is mainly used to develop mobile applications.

4) **JavaFx**

- It is used to develop rich internet applications. It uses light-weight user interface API.

Features of Java



Features of Java

Platform Independent

- A platform is the hardware or software environment in which a program runs.
- There are two types of platforms software-based and hardware-based. Java provides software-based platform.
- The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on the top of other hardware-based platforms. It has two components:

  Runtime Environment

  API(Application Programming Interface)

- Java code can be run on multiple platforms e.g. Windows, Linux, Sun Solaris, Mac/OS etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere(WORA).

Features of Java

Secured

- Java is secured because: No explicit pointer. Java Programs run inside virtual machine sandbox
- **Classloader**: adds security by separating the package for the classes of the local file system from those that are imported from network sources.
- **Bytecode Verifier**: checks the code fragments for illegal code that can violate access right to objects.
- **Security Manager**: determines what resources a class can access such as reading and writing to the local disk.
- These security are provided by java language. Some security can also be provided by application developer through SSL, JAAS, Cryptography etc.

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.

Features of Java

Architecture-neutral

- 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 architecture 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.

High-performance

- Java is faster than traditional interpretation since byte code is "close" to native code still somewhat slower than a compiled language (e.g., C++)

Features of Java

Distributed

- We can create distributed applications in java. RMI and EJB are used for creating distributed applications. We may access files by calling the methods from any machine on the internet.

Multi-threaded

- 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. Threads are important for multi-media, Web applications etc.

Basics of Java

- Java Variables
- Java If-else
- Java switch
- Java For Loop
- Java while Loop
- Java Do While Loop
- Comments
- Java Functions
- Java Arrays

Hello Java Program

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

Output: Hello Java

Understanding first java program

- Let's see what is the meaning of class, public, static, void, main, String[], System.out.println().
- **class** keyword is used to declare a class in java.
- **public** keyword is an access modifier which represents visibility, it means it is visible to all.
- **static** is a keyword, if we declare any method as static, it is known as static method. The core advantage of static method is that there is no need to create object to invoke the static method. The main method is executed by the JVM, so it doesn't require to create object to invoke the main method. So it saves memory.
- **void** is the return type of the method, it means it doesn't return any value.
- **main** represents startup of the program.
- **String[] args** is used for command line argument.

  System.out.println() is used to print statement.

Java OOP Concepts

- **Object** means a real word entity such as pen, chair, table etc. **Object-Oriented Programming** is a methodology or paradigm to design a program using classes and objects. It simplifies the software development and maintenance by providing some concepts:
  - Object
  - Class
  - Inheritance
  - Polymorphism
  - Abstraction
  - Encapsulation

Java OOP Concepts

Object

An object has three characteristics:

- **state:** represents data (value) of an object.
- **behavior:** represents the behavior (functionality) of an object such as deposit, withdraw etc.
- **identity:** Object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. But, it is used internally by the JVM to identify each object uniquely.
- For Example: Pen is an object. Its color is white, known as its state. It is used to write, so writing is its behavior.

Object is an instance of a class.

Java OOP Concepts

Class

- A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.
- A class in Java can contain:
  - **fields**
  - **methods**
  - **constructors**
  - **blocks**
  - **nested class and interface**

Java OOP Concepts

```
class Student
{
    int id;//field or data member or instance variable
    String name;
    public static void main(String args[])
    {
        Student s1=new Student();//creating an object of Student
        System.out.println(s1.id);//accessing member
        System.out.println(s1.name);
    }
}
```

End.