Object Oriented Programming with Java (Subject Code: BCS-403)

Unit 1

Lecture 1

Lecture 1

- Why Java
- History of Java
- OOPs Concept
- Features of Java

Why Java

1. Java is Easy to Learn

Java is beginner-friendly and one of the most popular programming languages among new developers. It has a syntax similar to English and enables you to write, debug, compile, and learn java programming fast.

2. Java is Versatile

Java follows the ‘write once and run anywhere’ principle and can be used for programming applications using different platforms.

3. Java is Object-Oriented

Java is an object-oriented programming language and this makes it scalable and flexible. Since it uses the syntax of an object-oriented programming language, the developers can create modular programs.

4. Java is Scalable

Java is used everywhere, including desktops, mobile, applications, and so on. It can effectively run on any operating system and is ideal for building applications. This scalability and versatility have made Java a game-changing language across multiple sectors and devices.

5. Java is Platform-Independent

Java has the ability to easily move across platforms and can be run similarly on different systems. This critical nature of being platform-independent at the source and binary levels makes Java an essential language to learn for developers.

6. Java Has a Rich API

Java has a rich Application Programming Interface (API) system that includes packages, interfaces, and classes, along with their methods and fields. This enables developers to integrate various websites and applications.

7. Java is Open Source

Most of Java’s features are open-source; this makes building applications cheap and easy. Java has the support of libraries like Google Guava, Maven, JHipster, and Apache Commons, allowing developers a wide choice to work with.

8. Java is Free of Cost

Java is a free-to-download software on Oracle Binary Code License (BCL), enabling beginners to develop applications easily and learn Java programming effectively.

History of Java

- Java is an Object-Oriented programming language developed by **James Gosling** in the early 1990s.
- The team initiated this project to develop a language for digital devices such as set-top boxes, television, etc.
- Originally C++ was considered to be used in the project but the idea was rejected for several reasons.
- James Gosling and his team called their project “**Greentalk**” and its file extension was **.gt** and later became to known as “**OAK**”.

Why “Oak”?

- The name **Oak** was used by **Gosling** after an **oak tree** that remained outside his office.
- But they had to later rename it as “**JAVA**” as it was already a trademark by **Oak Technologies**.
- **Java** name was decided after much discussion since it was so unique.
- Gosling came up with this name while having a coffee near his office.

- Java was created on the principles like **Robust, Portable, Platform Independent, High Performance, Multithread, etc.** and was called one of the **Ten Best Products of 1995** by the **TIME MAGAZINE**.

History of various Java versions

VERSION	RELEASE DATE
JDK Beta	1995
JDK 1.0	January 1996
JDK 1.1	February 1997
J2SE 1.2	December 1998
J2SE 1.3	May 2000
J2SE 1.4	February 2002
J2SE 5.0	September 2004
JAVA SE 6	December 2006
JAVA SE 7	July 2011
JAVA SE 8	March 2014
JAVA SE 9	September 2017
JAVA SE 10	March 2018
JAVA SE 11	September 2018
JAVA SE 12	March 2019

Latest Versions of Java

- Java SE 21 (LTS) September, 19th 2023
- Java 22 was released on March 19, 2024.

OOPs (Object Oriented Programming System)

- **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

Object - Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behaviours -wagging, barking, eating. An object is an instance of a class.

Class - A class can be defined as a template/blue print that describes the behaviors/states that object of its type support

Inheritance

When one object acquires all the properties and behaviours of parent object i.e. known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

Polymorphism

- When **one task is performed by different ways** i.e. known as polymorphism. For example. shape or rectangle etc.
- In java, we use method overloading and method overriding to achieve polymorphism.

Abstraction

- **Hiding internal details and showing functionality** is known as abstraction. For example: phone call, we don't know the internal processing.
- In java, we use abstract class and interface to achieve abstraction.

Encapsulation

- **Binding (or wrapping) code and data together into a single unit is known as encapsulation**. For example: capsule, it is wrapped with different medicines.
- A java class is the example of encapsulation. Java bean is the fully encapsulated class because all the data members are private here.

Features of Java

- **Object Oriented :** In java everything is an Object.
- **Platform independent:** Unlike many other programming languages including C and C++ when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code.
- **Simple :**Java is designed to be easy to learn. If you understand the basic concept of OOP java would be easy to master.
- **Secure :** With Java's secure feature it enables to develop virus-free, tamper-free systems.

- **Architectural- neutral :**Java compiler generates an architecture-neutral object file format which makes the compiled code to be executable on many processors.
- **Portable :**being architectural neutral and having no implementation dependent aspects of the specification makes Java portable.
- **Robust :**Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
- **Multi-threaded :** With Java's multi-threaded feature it is possible to write programs that can do many tasks simultaneously.

- **Interpreted :**Java byte code is translated on the fly to native machine instructions and is not stored anywhere.
- **Distributed :**Java is designed for the distributed environment of the internet.