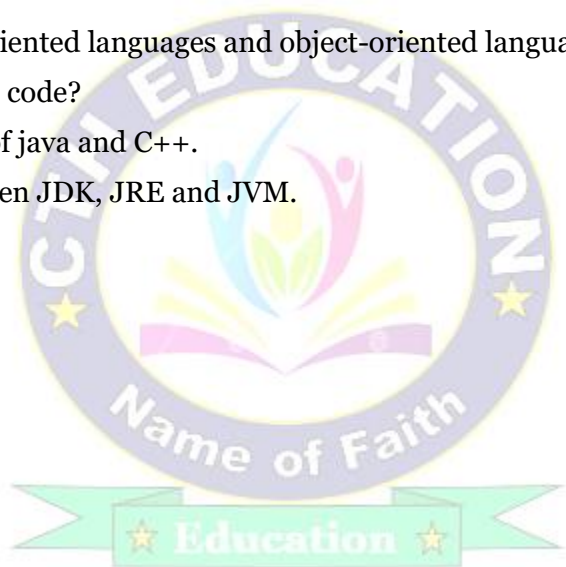


UNIT-01: Principles of Object-Oriented Programming with Introduction to JAVA

- The Traditional approach, drawback of procedure-oriented languages,
- The three basic constructs of OOPS including abstraction and encapsulation.
- Comparison of various object-oriented languages, Need of java,
- The creation of java, Basic differences of java and C++,
- byte code, difference between JDK, JRE, JVM,
- java applets and applications, java buzzword,
- three basic constructs of oops applicable to java.

Questions to be discussed:

1. Explain the basic concept of object oriented programming.
2. State four features of java.
3. Differentiate procedure-oriented languages and object-oriented languages.
4. What do you mean by byte code?
5. Explain basic differences of java and C++.
6. Write the difference between JDK, JRE and JVM.



Introduction to Java:

- Java is an object-oriented, class-based, secured and general-purpose computer programming language.
- It was developed by Sun Microsystems in the year 1995.
- James Gosling is known as the father of Java.
- Before Java, its name was Oak.
- Java runs on a variety of platforms, such as Windows, Mac OS, and UNIX OS.

An Overview/History of Java:

- The history of Java starts with the Green Team. Java team members (also known as Green Team), initiated this project to develop a language for digital devices such as set-top boxes, televisions, etc.
- It was suited for internet programming.
- Currently, Java is used in internet programming, mobile devices, games, e-business solutions, etc.
- JDK 1.0 released in (January 23, 1996).
- Now Java is being used in Windows applications, Web applications, enterprise applications, mobile applications etc.

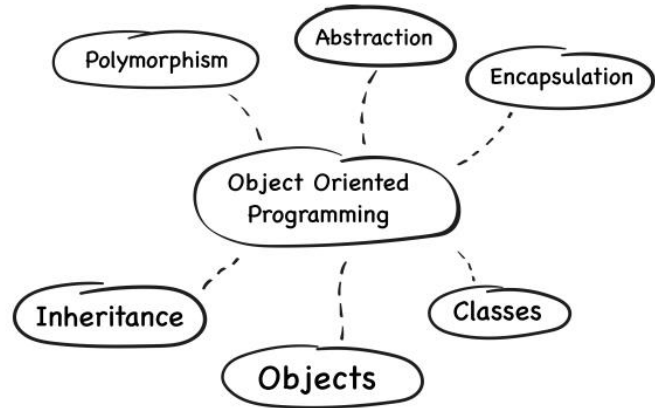
Difference between C and JAVA:

C	JAVA
C is a Procedural Programming Language.	Java is Object-Oriented language.
C was developed by Dennis M. Ritchie between 1969 and 1973.	Java was developed by James Gosling in 1995.
It is a compiled language.	It is an interpreted language.
It does not follow OOPs concepts.	It follows OOPs concepts.
The file is saved with the extension .c .	The file is saved with the extension .java .
It is not secure.	It is fully secured language.
It translates the code into machine language so that the machine can understand the code.	It translates the code into a bytecode that is executed by the JVM.
It does not support inheritance that is useful for code reusability.	It supports inheritance that provides code reusability.
It generates .exe file.	It generates .class file.
It directly executes the code.	It executes code with the help of JVM.

Basic Concepts of Object-Oriented Programming:

- OOP is a methodology or paradigm to design a program using classes and objects.
- Alan Kay coined the term “object oriented programming” at grad school in 1966 or 1967.
- It simplifies software development and maintenance by providing some concepts:

1. Object
2. Class
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation



Objects

- Any entity that has state and behavior is known as an object.
- An object is an instance of class.
- For example a chair, pen, table, keyboard, bike, etc.

Class

- Collection of objects is called class.
- Class is a user-defined data-type
- It is the basic building block of OOP.

Example:

Class: Human

Object: Man, Woman

Class: Fruit

Object: Apple, Banana, Mango, Guava etc.

Class: Mobile phone

Object: iPhone, Samsung, Moto, Nokia etc.

Class: Food

Object: Pizza, Burger, Samosa

Inheritance

- When one object acquires all the properties and behaviors of a parent object, it is known as inheritance.

Polymorphism

- Existing in multiple forms.
- If one task is performed by different ways, it is known as polymorphism.

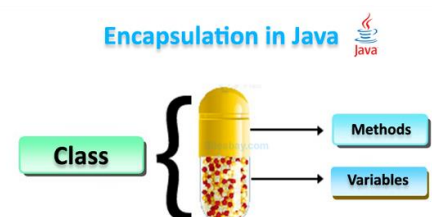
Abstraction

- Hiding internal details and showing functionality is known as abstraction.
- For example phone call, we don't know the internal processing.



Encapsulation

- Binding (or wrapping) code and data together into a single unit are known as encapsulation.
- For example capsule, it is wrapped with different medicines.



Features of Java:

- The primary objective was to make it portable, simple and secure programming language.
- The features of Java are also known as Java buzzwords.
- Java supports dynamic compilation and automatic memory management (garbage collection).
- A list of the most important features of the Java language is given below.
 1. Simple
 2. Object-Oriented
 3. Portable
 4. Platform independent
 5. Secured
 6. Dynamic etc.

Simple:

- It is very easy to learn, and its syntax is simple, clean and easy to understand.
- According to Sun Microsystem, Java language is a simple programming language because:
- Java syntax is based on C++.

Object-oriented:

- Java is an object-oriented programming language. Everything in Java is an object.
- OOPs is a methodology that simplifies software development and maintenance by providing some rules.

Platform Independent:

- Java code can be executed on multiple platforms, for example, Windows, Linux, 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).

Portable

- Java is portable because it facilitates you to carry the Java bytecode to any platform.
- It doesn't require any implementation.

Secured:

- Java is best known for its security.
- With Java, we can develop virus-free systems.

Dynamic:

- Java is a dynamic language & it supports the dynamic loading of classes.
- It means classes are loaded on demand.

Explain the difference between POP & OOP.

POP	OOP
POP Stands for Procedural Oriented Programming.	OOP Stands for Object Oriented Programming.
Entire program is divided into functions.	Entire program is divided into objects.
It follows Top-down approach.	It follows Bottom-up approach.
No access specifiers are supported.	Access specifiers are supported.
Here, no concept of overloading.	It overloads functions, constructors & operators.
Inheritance is not supported.	Inheritance is supported.
No data hiding is present, so data is insecure	Encapsulation is used to hide in data.
C, VB, FORTRAN, Pascal etc.	C++, JAVA, VB.NET, Python etc.

Write the difference between JDK, JRE and JVM.

- JDK is the development platform, while JRE is for execution.
- JVM is the foundation, or the heart of the Java programming language.
- JVM is included in both JDK and JRE—Java programs won't run without it.

JDK	JRE	JVM
JDK stands for Java Development Kit.	JRE stands for Java Runtime Environment.	JVM stands for Java Virtual Machine.
The JDK is a software development kit that develops applications in Java.	The JRE is an implementation of JVM. It is a type of software package that provides class libraries of Java.	JVM is a platform-independent abstract machine that has three notions in the form of specifications.
The JDK is platform-dependent.	JRE, just like JDK, is also platform-dependent.	The JVM is platform-independent.
JDK = Development Tools + JRE (Java Runtime Environment)	JRE = Libraries for running the application + JVM (Java Virtual Machine)	JVM = Only the runtime environment that helps in executing the Java bytecode.
It consists of various tools required for writing Java programs.	If a user wants to run the Java applets, then they must install JRE on their system.	It provides its users with a platform-independent way for executing the Java source code.

What do you mean by Byte Code?

- Byte Code can be defined as an intermediate code generated by the compiler after the compilation of source code(JAVA Program).
- This intermediate code makes Java a platform-independent language.

How is Byte Code generated?

- Compiler converts the source code or the Java program into the Byte Code(machine code), and secondly, the Interpreter executes the byte code on the system.
- The Interpreter can also be called JVM(Java Virtual Machine).
- The byte code is the common piece between the compiler(which creates it) and the Interpreter (which runs it).

