DAKSH BANGORIA 21CE007

1 Intro. To Object Oriented Concepts,

2 Compilation of Java with other object oriented programming language

3 Introduction to JDK,JVM,JRE

Ans 1 :

Object-oriented programming is a model that provides different types of concepts, such as **inheritance, abstraction, polymorphism**, etc. These concepts aim to implement real-world entities in programs. They create working methods and variables to reuse them without compromising security

When completing an object-oriented design, there are five basic concepts to understand: **classes/objects, encapsulation/data hiding, inheritance, polymorphism, and interfaces/methods**

**List of OOPs Concepts in Java**

* Objects
* Classes
* Object
* Class
* Abstraction
* Inheritance
* Polymorphism
* Encapsulation

Ans 2 :-

Java is an Object Oriented programming language which is developed by [James Gosling](https://en.wikipedia.org/wiki/James_Gosling) and it was published by [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) in 1995.

**Java**is one of the most popular and widely used programming language and platform. A platform is an environment that helps to develop and run programs written in any programming language.

Java is fast, reliable and secure.

#### The main difference between Java and any other programming language is the unique method in which Java code is executed. **Java is compiled into bytecode which can run on any device with the Java Virtual Machine (JVM)**. **Java vs C** (Very Different)

#### **Java vs C** (Very Different)

* Java is platform-independent. C is platform-dependent.
* Java is an object-oriented programming (OOP) language. C is procedural.
* Java follows the bottom-up approach. C follows the top-down approach.
* Java has automatic memory management. C has manual memory management.
* Java has built-in thread support. C does not.

#### **Java vs C#** .

* Java is used for developing web applications and games.
* C# is used for game and mobile development.
* Java is platform-independent. C# is platform-dependent

#### **Java vs C++**

* Java is derived from C++.
* Java is compiled to bytecode (platform-independent). C++ is compiled to machine code.
* Java has automatic garbage collection. C++ does not.
* Java doesn’t support operator overloading. C++ does.
* Java has a powerful cross-platform library. C++ libraries are robust but simple.

#### **Java vs Javascript**

* Java is a programming language. Javascript is a scripting language.
* Java runs on the Java Virtual Machine (JVM). Javascript runs on web browsers.
* Java is compiled and interpreted. Javascript is interpreted.
* Java requires more memory than Javascript.

#### **Java vs Python** (Some Similarities)

* Java is essentially a compiled language. Python is an interpreted language.
* Java requires more lines of code than Python for an equivalent program.
* Java is faster than Python on execution.

#### **Java vs Ruby**

* Java is a static programming language. Ruby is dynamic.
* Java is faster than Ruby..

Ans 3 :-

JVM = JAVA VIRTUAL MACHINE

JRE = JAVA RUNTIME ENVIROMENT contains JVM and all the other libraries to run Java application.

JDK=JAVA DEVELOPING KIT is a superset which comprises of JVM, JRE, and the tools to develop Java Application.

JRE is an environment, in order to execute any Java program locally. JVM is where it's responsible for converting the Bytecode into machine specific code and makes java program write-once-run-anywhere.

1. **The JDK is a key platform component for building Java applications as its heart is the Java compiler**
2. **The JVM is a virtual machine, an abstract computer that has its own ISA, own memory, stack, heap, etc**. It runs on the host OS and places its demands for resources to it. The JVM is the Java platform component that executes programs.
3. The JRE is the on-disk part of Java that creates the JVM.
4. The JDK allows developers to create Java programs that can be executed and run by the JVM and JRE.

**A JRE is part of a Java development kit (JDK)**. A JRE is made up of a Java virtual machine (JVM), Java class libraries, and the Java class loader. JDKs are used to develop Java software; JREs provide programming tools and deployment technologies; and JVMs execute Java programs.