Java features
Buzzwords

1. Simple and familiar

- it is simple because of its coding style is very clean & easy to understand
- familiar because it has base of familiar languages like C & C++ & contain many features of these
- it removes the drawbacks, complexites and confusing elements of C/C++

2. Compiled & Interpreted

- Java combines both these approaches thus making it 2 stage system.
- it integrates the power of compiled languages with the flexibility of interprted languages
- Java compiler (Java C) compiles the Java source code itito the bytecode
- Java virtual machine then executes this bytecode which is executable on many OS & is portable

3. Platform independent

- Being platform independent means a program compiled on one machine can be executed on any machine in the world without any change
- Java achieves platform independence by using the concept of the BYTE Code

4. Portable

- portability of Java comes from artitecture - neurability
- Java bytecode can run on any hardware that has a complaint JVM which can convert the bytecode according to machine
- In Java the size of primitive data types is machine independent which make Java programs portable among different platforms
- Any changes & updates made in OS Processors & system resources will not enforce any changes in Java programs

5. Architectural Neutral

- the program written on one platform or OS is independent of other platform or environments & can run on any other OS without recompiling them.
- It is based on write Once run anywhere (WORA) or write once run everywhere (WORE) approach.
- This feature is very useful when we develop applets or download applications from the internet

6. Object Oriented

- it strongly supports the concept of Object Oriented programming due to which it is called a pure object oriented language
- it supports major Object oriented programming features like Encapsulation, Abstraction, & inheritance.
- Almost everything in Java is an object
- All programs & data live within objects & classes

7 Robust

- Java is capable of handling runtime errors, supports automatic garbage collections & exception handling & avoids explict pointer concepts
- it has strong memory management systems
- it helps in eleminating errors as it checks the code during both compile & runtime
- Exceptional handling identifies & removes runtime errors
- any runtime error when encounterd the program gets terminated.

8. Secure

- it not only verify all the memory access but also ensured that no thread of virus communicated through an applet
- the absence of pointers in Java ensures that programs cannot give access to memory locations without proper authorization.
- Java is a more secure language as compared to C/C++, as it does not allow a programmer to explicitily create pointers
- programs run in a virtual machine sandbox - A separate environment that allows users to execute their applications without affecting the underlying system.
- It has a bytecode verifier that checks the code fragments for any illegeal code that violates the access right.

9. Distributed

- Java is distributed because it encourages user to create distributed application
- In Java a program can be split into many parts & store these parts on different computers
- a Java programmer sitting on a machine can access another program running on the other machine
- This feature is very helpful in developing large projects

10 multi-threadad & interactive

- multithreaded means handling multiple tasks simultaneously or executing multiple functions of the same program in parallel

11. High Performance

- it provides high performance with the use of JIT Just in time compiler. this saves time and makes it mor efficiont
- the illusion of multithreading enhances the overall exeution speed of Java

12 Dynamic & Extensible

- highly dynamic as it can adapt to its evolving environment
- even supports the functions written on other languages such as C / C++ to be written in java programs.

These functions are called native methods

These methods are dynamically linked at runtime.

ACTIVITY-01

Q1. Explore the various IDE's available for implementing java source code.

1. Eclipse:

* Eclipse is a widely used open-source IDE for Java Development.

* It has a modular architecture, allowing developers to install plugins for additional functionality.

* Supports various languages through plugins.

* offers a rich set of features like code Completion, refactoring ;and debugging.

2. Visual Studio Code:

* While not a traditional java IDE, Vscode has gained popularity for Java development with the help of extensions.

* Extensions like "Red Hat Java" or "Language Support for java" Can turn VS Code into a Powerful java development environment.

3. Spring Tool suite :

* STS is an IDE based on Eclipse and tailored for developing spring applications.

* It includes features for working with the spring framework, such as integration with the spring boot.

4. Dr Java:

* Dr Java is a lightweight and fast IDE for java development.

* It is simple and easy to use, making it a good choice for beginners.

5. Jcreator:

* JCreator is a lightweight and fast IDE for Java development.

* It focuses on simplicity and ease to use making it suitable for developers who prefer a streamlined environment.

6. Blue J:

* Blue J is designed for beginners and is often used in edutional settings.

* It provides a simplified interface and Visualization tools to help understand object-oriented programming concepts.

7. IntelliJ IDEA:

* Developed by Jetbrains, IntelliJIDE with a focus on productivity.

* Known for its intelligent code completion, Code analysis and excellent support for various frameworks.

* offers a community edition(free) and an ultimate edition(paid) with additional features.

8. Net beans:

* Netbeans is an open-source IDE that Supports Java, as well as other languages.

* It provides features like smart code Completion, integrated debugger, & a visual GUI builder.

Activity-01

Q3. Compare and understand the java programming lanuage with Python.

Syntax:

Java:

* Strictly typed language with explict type declarations.

* Requires the use of semicolons to terminate statements.

* Relies on curly braces '{}' to define blocks of code.

Python:

* Dynamically typed language with implicit type declarations.

* Uses indentation (whitespace) to denote blocks of code instead of Curly braces.

* Requires no semicolons at the end of statements.

Memory Management:

Java :

* Manages Memory automatically through garbage collector.

* Provides strong memory safety.

Python:

* Also utlizes automatic memory mangement through a garbage collector.

* Generally, Python's memory management is considered more simplified Compared to Java.

Perfomance:

Java:

* Typically considered faster than python in terms of raw execution speed.

* often chosen for performance-critical applications, such as large - scale enterprise systems.

Python:

* Generally slower than Java due to its interpreted nature.

* Suitable for a wide range of applications but may not be the best choice for performance sensitive tasks.

Use Cases:

Java:

* Widely used in enterprise-level applications, web development (especially on the server side), & Android app development.

* Known for its robustness and scalability.

Python:

* Popular for web developer, scripting, automation, data science, machine learning, & Artifical intelligence.

* Not as comman in enterprise-level Systems but gaining popularity in various domains.

Community and Ecosytem:

8/10

Java:

* Has a mature and extensive ecosytem with a wide range of libraries and frameworks.

Python:

* Boasts a rich ecosytem with a vast collection of libraries and frameworks (e.g., Django, NumPy, etc.)