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, complexities 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 interpreted languages
- Java compiler (Java c) compiles the Java source code into 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 architecture - neurability
- Java bytecode can run on any hardware that has a compliant 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 eliminating errors as it checks the code during both compile & runtime
- Exceptional handling identifies & removes runtime errors
- any runtime error when encountered 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 explicitly 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-threaded & interactive

- multi-threaded 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 more efficient
- the illusion of multithreading enhances the overall execution 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.

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

There are several Integrated Development Environment (IDEs) available for java development, each offering unique features and capabilities. Here are some of the popular IDEs for implementing Java Source code.

1. Eclipse

Eclipse is a widely used opensource IDE with extensive support for the java development. It offers features like code completion, debugging refactoring and a wide range of plugins. Eclipse is known for its flexibility and is often used in various domains, including web development and enterprise applications.

2. IntelliJ IDEA

IntelliJ IDEA, developed by JetBrains, smart code is a powerful and feature-rich commercial IDE It provides advanced code analysis, smart code completion, integrated version control and the support for various frameworks.

IntelliJ IDEA is known for its user-friendly interface and productivity-enhancing features

3. NetBeans

NetBeans is an open-source IDE that supports Java Development along with the other languages It offers features like code generators, version control and a variety of plugins.
NetBeans is known for its ease of use and is particularly suitable for Java desktop and web applications.

4. JDeveloper

Oracle JDeveloper is an IDE provided by Oracle for Java and other languages
It is well integrated with Oracle's development tools and technologies
JDeveloper is commonly used for developing applications on the Oracle platform.

5. Jupyter Notebook with Java kernel:

Jupyter Notebook is a popular open source web application for interactive computing
With the use of the Java kernel, Jupyter Notebook can be employed for Java development allowing for interactive code execution and visualization.

Compare and understand the java programming language with python

Java and python are both popular and powerful programming languages, but they have distinct characteristics, use cases and the syntax. Here's a comparision.

1. Syntax

Java - requires explict declaration of data types strict syntax, and the use of semicolons to terminate statements

Python - uses a more concise and readable syntax with dynamic typing and indentation to denote code blocks.

2. Typing

Java - statically typed languages, making the variable types must be declared at compile time

Python - Dynamically typed language, allowing variables to be assigned without explict type declarations.

3. Execution

Java - compiled language that translates the source code to bytecode, which runs on the Java Virtual Machine (JVM)

Python - Interpreted language where source code is executed line by line by the python interpreter

4. Platform Independence

Java - known for its 'write once Run Anywhere (WORA) phylosophy, allowing Java programs to run on any device with a JVM

Python - platform independent in theory but Python programs may require specific interpreters or dependencies.

5. Concurrency:

Java - Strong support for multithreading and concurrency with features like synchronized methods & the java.util.concurrent Package

Python - Global Interpreter Lock (GIL) can limit the effectiness of multi-threading in CPython (the default Python interpreter) but multiprocessing can be used for the parallelism.

8/10