1. **Java Modifiers**

**Modifiers in Java are keywords that specify the characteristics of classes, methods, variables, and constructors. They provide additional information about the accessibility, behavior, and lifecycle of these elements.**

**Access Modifiers:**

* **public: Accessible from anywhere.**
* **private: Accessible only within the same class.**
* **protected: Accessible within the same package or subclasses.**
* **default: Accessible within the same package.**

**Non-Access Modifiers:**

* **static: Belongs to the class rather than individual objects.**
* **final: Prevents modification after initialization.**
* **abstract: Declares a method without a body, requiring implementation in subclasses.**
* **synchronized: Ensures thread safety for methods and blocks of code.**
* **native: Implemented in a non-Java programming language.**
* **volatile: Indicates that a variable may be modified by multiple threads.**

**2. Java Buzzword:**

* **OOP:** Object-Oriented Programming is a fundamental paradigm in Java, emphasizing the creation of objects with properties and behaviors.
* **Platform Independence:** Java's ability to run on any platform with a JVM makes it highly portable.
* **Robustness:** Java's strong type checking and exception handling mechanisms contribute to its reliability.
* **Security:** Java's security features, such as bytecode verification and sandboxing, help protect against malicious code.
* **Performance:** Java's JIT compilation and optimization techniques have significantly improved its performance.
* **Multithreading:** Java's built-in support for multithreading allows concurrent execution of tasks, enhancing responsiveness and efficiency.

**3. Components of Java:**

* **Java SE (Standard Edition):** The core platform for general-purpose Java applications.
* **Java EE (Enterprise Edition):** A set of APIs for building large-scale enterprise applications.
* **Java ME (Micro Edition):** A platform for developing applications for embedded devices and mobile phones.

**4. Meaning of main method:**

* The main method is the entry point of a Java program.
* It is the first method executed when the program starts.
* It must be declared as public, static, and have a void return type.
* It accepts an array of strings as arguments (command-line arguments).

**5. Java Modifiers:**

* **Access Modifiers:** public, private, protected, default (package-private)
* **Non-Access Modifiers:** static, final, abstract, synchronized, native, volatile

**6. Access Modifiers:**

* **public:** Accessible from anywhere.
* **private:** Accessible only within the same class.
* **protected:** Accessible within the same package or subclasses.
* **default:** Accessible within the same package.

**7. Java Virtual Thread:**

* A lightweight thread implementation introduced in Java 19.
* Created and managed by the JVM, allowing for more efficient handling of concurrent tasks.
* Can be used to improve application performance and scalability.

**8. Entry Point Method:**

* The main method is the entry point of a Java program.
* It is the first method executed when the program starts.

**9. Meaning of System.in and System.out:**

* **System.in:** An input stream representing the standard input device (usually the keyboard).
* **System.out:** An output stream representing the standard output device (usually the console).

**10. Size of Data Types:**

|  |  |
| --- | --- |
| **Data Type** | **Size (bits)** |
| byte | 8 |
| short | 16 |
| int | 32 |
| long | 64 |
| float | 32 |
| double | 64 |
| char | 16 |
| boolean | 1 |

**11. Loader in Java:**

* A component of the JVM responsible for loading class files into memory.
* There are three types of loaders: bootstrap, extension, and system class loaders.

**12. Garbage Collector:**

* A component of the JVM responsible for automatically reclaiming memory that is no longer in use.
* Different garbage collection algorithms are used to optimize memory management.