**1. What are the basic steps to write and run a Java program using Notepad?**

**Answer:**

1. **Open Notepad** and write a Java program (e.g., HelloWorld.java).
2. Save the file with a **.java extension** matching the class name (e.g., HelloWorld.java).
3. Open **Command Prompt**, navigate to the directory using cd.
4. Compile using: javac HelloWorld.java.
5. Run using: java HelloWorld.  
   This compiles the .java file into a .class file and executes it using the JVM.

**2. Why is the file name same as the class name in Java?**

**Answer:**  
In Java, the file name **must match the public class name** because the JVM looks for that specific class as the entry point. If they don’t match, the compiler throws an error. For example, public class Hello must be saved as Hello.java.

**3. What is public static void main(String[] args)? Explain each keyword.**

**Answer:**

* public: Access modifier, makes the method accessible from anywhere.
* static: Belongs to the class, not objects; JVM can call it without object creation.
* void: No return value.
* main: The entry point for Java programs.
* String[] args: Command-line arguments array.  
  This signature is **mandatory** for Java execution.

**4. Why do we write static in the main method?**

**Answer:**  
We use static so that the **JVM can execute the main() method without creating an object** of the class. Since the program starts before any object exists, static allows the method to run as part of the class itself.

**5. What does the System.out.println() statement do?**

**Answer:**

* System: A built-in Java class.
* out: A static object of PrintStream within System, connected to console.
* println(): Prints the string and moves to a new line.  
  This method is used to **display output on the console**.

**6. What is JVM, JDK, and JRE?**

**Answer:**

* **JVM (Java Virtual Machine)**: Executes compiled .class files.
* **JDK (Java Development Kit)**: Includes tools like javac, debugger, and JRE.
* **JRE (Java Runtime Environment)**: Provides libraries and JVM needed to run Java.  
  JDK = JRE + development tools.

**7. Why do we need to compile Java code before running it?**

**Answer:**  
Java is a **compiled language**. The compiler (javac) translates the .java file to **bytecode (.class file)**. This is a platform-independent format run by the JVM, ensuring portability.

**8. Can we have multiple classes in one Java file?**

**Answer:**  
Yes, but **only one class can be public**, and the file name must match that class. Other non-public classes can exist, but they cannot be accessed from outside the file directly.

**9. What happens if we remove static from the main method?**

**Answer:**  
If static is removed, JVM cannot call main() directly, as it requires an object. This results in a **NoSuchMethodError** at runtime because JVM looks for public static void main(String[] args).

**10. What is the role of the String[] args parameter?**

**Answer:**  
It allows the program to **accept command-line arguments**. For example, running java Hello abc will store "abc" in args[0]. It's useful for dynamic input without GUI or Scanner.

**11. How does the javac and java command work?**

**Answer:**

* javac Hello.java: Compiles source code → generates Hello.class.
* java Hello: Invokes JVM to run the main() method of Hello.class.

**12. What if we write class name in lowercase while running java hello?**

**Answer:**  
Java is **case-sensitive**, so java hello will result in an error like:

Error: Could not find or load main class hello

Always use **exact case** as defined in your class name.

**13. Can a Java file have no main method?**

**Answer:**  
Yes, but it won't run directly. It can be used for libraries or helper classes. To run a class, JVM **must find a main() method** as the entry point.

**14. What is the significance of ; in Java?**

**Answer:**  
Semicolons terminate **statements** in Java. Missing ; causes **syntax errors** during compilation. It separates each complete instruction.

**15. How do comments work in Java?**

**Answer:**

* Single-line: // comment
* Multi-line: /\* comment \*/  
  Comments are ignored by the compiler and used to **improve readability**.

**16. What is the extension of a Java source file and class file?**

**Answer:**

* Source file: .java (e.g., Hello.java)
* Compiled file: .class (e.g., Hello.class)  
  Java compiler generates .class file from .java.

**17. What happens if we misspell main()?**

**Answer:**  
The program compiles, but throws a **runtime error**:

Main method not found in class...

The method name must be exactly main.

**18. Is Java fully object-oriented if main is static?**

**Answer:**  
Java is **not 100% object-oriented** because it supports **static context** and **primitive types**. static is a necessity to start execution, even before object creation.

**19. What are the key rules for naming a Java class?**

**Answer:**

* Must begin with a letter or \_
* Can include letters, digits, \_, $
* No spaces or symbols like @, #
* Should follow **CamelCase**, e.g., MyFirstProgram
* Should **match the file name** if public

**20. How does Java ensure platform independence?**

**Answer:**  
Java source code is compiled into **bytecode**, which is the same on all systems. The **JVM on each OS** interprets the bytecode, ensuring **Write Once, Run Anywhere** behavior.