1. Install Java (JDK)

- Download & install JDK (OpenJDK or Oracle JDK).
- Set environment variables:
  - `JAVA_HOME` → path to JDK.
  - Add `JAVA_HOME/bin` to `PATH`.
- Verify installation:
- ```
  java -version
  javac -version
  ```

2. Choose Editor/IDE

- Text editor (Notepad, VS Code, Sublime) + terminal.
- IDEs (IntelliJ IDEA, Eclipse, NetBeans).
- For beginners, **IntelliJ IDEA** or **VS Code with Java extensions** is easiest.

3. Create Java File

1. Make a folder (e.g., `FirstJava`).
2. Create file: **HelloWorld.java**.

Code:

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Explanation:

- `public class HelloWorld` → class name = file name.
- `main` method → entry point of program.
- `System.out.println` → prints text to console.
- Each statement ends with `;`.

4. Compile & Run (Command Line)

- Compile:
- `javac HelloWorld.java`

  → creates `HelloWorld.class` file.

- Run:
- `java HelloWorld`

  → Output:

  ```
  Hello, World!
  ```
  ```

5. Common Errors

- `javac: command not found` → JDK not set in PATH.
- `class HelloWorld is public, should be declared in a file named HelloWorld.java` → filename and class mismatch.
- Wrong `main` method signature → program won't run.

6. Add More Examples

- Variables:
- `int a = 5;`
- `System.out.println("Value of a: " + a);`
- Comments:
- `// single-line`
- `/* multi-line */`
- Arithmetic:
- `int x = 2 + 3;`
- `System.out.println("x = " + x); // x = 5`
- Data types: `int`, `double`, `boolean`, `char`, `String`.

7. Compile vs Run

- **Compile (javac)** → converts `.java` → `.class` (bytecode).
- **Run (java)** → JVM executes `.class` file.

8. Using an IDE

- IntelliJ IDEA:
  - New Project → New Java Class → write code → Run ▶️.
- VS Code:
  - Open folder → create file → run with Java extension pack.