Java Programming Tutorial

Java Installation + First OOP Program (Student Class & Object Creation)

PART A: JAVA INSTALLATION (FROM SCRATCH)

1. What is Java?

Java is a programming language and a platform.

To write and run Java programs, we need:

- **JDK (Java Development Kit)** – to write and compile programs
- **JRE (Java Runtime Environment)** – to run programs

Installing **JDK automatically includes JRE**, so we install only JDK.

2. System Requirements

- Operating System: Windows 10 / 11 (recommended)
- RAM: Minimum 4 GB
- Disk Space: At least 2 GB free
- Administrator access

3. Java Version to Install

Java 17 (LTS – Long Term Support)

- Stable
- Industry standard
- Best for learning OOP

4. Downloading Java (JDK)

1. Open a browser
2. Go to: https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html
3. Download **Windows x64 Installer (.exe)**

5. Installing Java on Windows

1. Double-click the downloaded .exe file
2. Click **Yes** if permission is asked

3. Click **Next**
4. Keep default settings
5. Note the installation path (usually):
   C:\Program Files\Java\jdk-17
6. Click **Install → Finish**

Java is now installed, but not yet configured.

6. Setting Environment Variables (IMPORTANT)

Step 1: Open Environment Variables

1. Press **Windows + R**
2. Type sysdm.cpl and press Enter
3. Go to **Advanced → Environment Variables**

Step 2: Set JAVA_HOME

Under **System Variables**:

1. Click **New**
2. Variable Name: JAVA_HOME
3. Variable Value:
   C:\Program Files\Java\jdk-17
4. Click **OK**

Step 3: Update PATH Variable

1. Under **System Variables**, select Path
2. Click **Edit → New**
3. Add:
   %JAVA_HOME%\bin
4. Click **OK → OK → OK**

Step 4: Restart Command Prompt

Close all terminals and reopen them.

---

7. Verifying Java Installation

Open **Command Prompt** and type:

java -version

Expected output:

java version "17.x.x"

Then check compiler:

javac -version

If both work, Java is correctly installed.

PART B: FIRST JAVA OOP PROGRAM

---

8. Understanding Class and Object

- **Class** → Blueprint or template
- **Object** → Instance created from a class

Example:

- Student → Class
- Rahul, Anita → Objects

---

9. Creating the Student Class

File Name: Student.java

```
class Student {

    int id;
    String name;
    int age;

    void displayDetails() {
        System.out.println("Student ID: " + id);
        System.out.println("Student Name: " + name);
        System.out.println("Student Age: " + age);
    }
}
```

---

10. Creating Object of Student Class

File Name: StudentDemo.java

```
class StudentDemo {

    public static void main(String[] args) {

        Student s1 = new Student();

        s1.id = 101;
        s1.name = "Rahul";
        s1.age = 19;

        s1.displayDetails();
    }
}
```

---

11. Explanation of Object Creation

Student s1 = new Student();

- Student → Class (data type)
- s1 → Object reference
- new → Allocates memory
- Student() → Creates object

---

12. Compiling and Running the Program

Step 1: Compile

javac Student.java StudentDemo.java

Step 2: Run

java StudentDemo

Output:

Student ID: 101

Student Name: Rahul

Student Age: 19

---

13. Creating Multiple Objects

Student s2 = new Student();

s2.id = 102;

s2.name = "Anita";

s2.age = 18;

s2.displayDetails();

Same class → multiple objects → different data

---

14. Improving Code Using Constructor

Updated Student.java

```
class Student {

    int id;
    String name;
    int age;

    Student(int i, String n, int a) {
        id = i;
        name = n;
        age = a;
    }

    void displayDetails() {
        System.out.println("Student ID: " + id);
        System.out.println("Student Name: " + name);
        System.out.println("Student Age: " + age);
    }
}
```

Updated StudentDemo.java

```
class StudentDemo {

    public static void main(String[] args) {

        Student s1 = new Student(101, "Rahul", 19);
        Student s2 = new Student(102, "Anita", 18);

        s1.displayDetails();
        s2.displayDetails();
    }
}
```

---

15. Common Errors and Fixes

Error	Reason	Fix
java not recognized	PATH not set	Check environment variables
javac not found	JDK not installed	Install JDK
Compilation error	File name mismatch	Match class & file names

---

16. Student Practice Task

1. Add variables: branch, semester
2. Create 3 student objects
3. Display details of all students