Java Day 1

What is Java?

Java is an **object-oriented programming language** and a **platform** that allows programs to run on different systems using the JVM.

Why JVM is needed?

JVM (Java Virtual Machine) acts as a middle layer that executes Java bytecode so Java programs can run on any system without changing the code.

What is a variable?

A variable is a named memory location used to store data.

Why data types exist?

Data types tell Java **what kind of data** is stored, **how much memory** to allocate, and **what operations** can be performed on that data.

Important correction:

- Computer *does* understand data
- Data types help **Java manage memory and safety**, not variable names

What happens after we write Java code? (steps)

We write **.java** file

- **Compiler (javac)** converts it to **bytecode (.class)**
- Compiler checks **syntax errors**
- **JVM** executes bytecode line by line

Important precision:

- JDK **contains** the compiler
- Compiler does error checking, not JVM

Why Java is called *platform independent*?

Java is platform independent because compiled bytecode can run on any system that has a JVM, without changing the code.

Give one real-life example of a variable.

A labeled water bottle named “age” storing value 20.

Give one wrong example where data type matters (conceptual)

> If we store `10.5` in an `int` variable, Java will give an error because `int` cannot store decimal values.

OR

> If we store `"10"` in an `int`, Java will not allow it because it is a String, not a number.

Key idea:

- Java does **not guess**
- You **must** tell the data type clearly

Why does Java force you to declare data type before using a variable?

Java forces us to declare data types so it knows **how much memory to allocate**, **what operations are allowed**, and to **prevent type-related errors at compile time**.

In simple words:

- Java wants to be **safe**
- Java doesn’t want to **guess**
- Java wants errors **before runtime**

Mini Task: Input-based Calculator

Concepts:

- variables

- data types
- input
- basic operations

What data type will you use for numbers and why?

We can use `int` for simplicity, but `double` is safer because division can produce decimal results.

Example:

- `5 / 2`
  - `int` → `2` ❌ (data loss)
  - `double` → `2.5` ✅

What happens if the user enters `0` for division?

> If the user enters 0 for division, it causes a runtime error or invalid operation, so we must handle it using conditions.

In Java:

- `int / 0` → runtime exception
- `double / 0` → Infinity (still logically wrong)

So yes, **we must handle it manually**

What ONE thing felt easiest today, and what ONE thing felt hardest?