Types of Variables in Java – Complete Notes

---

1. Introduction

- A **variable** is a name given to a memory location that stores data.
- In Java, every variable is associated with a **data type** and a **scope**.
- Variables must be **declared before use**.

---

2. Types of Variables in Java

Java variables can be classified into the following categories:

🔹 a) Local Variables

- Declared **inside a method, constructor, or block**.
- Created when the method is invoked and destroyed after method exits.
- No default values → must be initialized before use.

```
class Example {
    void display() {
        int num = 10;  // Local variable
        System.out.println(num);
    }
}
```

🔹 b) Instance Variables

- Declared **inside the class but outside methods**.
- Each object has its own copy.
- Have **default values** (0, false, null, etc.).

```
class Student {
    String name;     // Instance variable
    int age;         // Instance variable
}
```

🔹 c) Static Variables (Class Variables)

- Declared using the ``** keyword**.
- Shared among all objects of a class.
- Have **default values**.
- Stored in the **method area memory**.

```
class Counter {
    static int count = 0;  // Static variable
    Counter() { count++; }
}
```

---

3. Difference Between Types of Variables

Feature	Local Variable	Instance Variable	Static Variable
Declared In	Method/block	Inside class	Inside class (with `static`)
Memory Allocation	Stack	Heap	Method Area
Lifetime	During method	Object lifetime	Class lifetime
Default Value	None	Yes	Yes
Shared Across Obj.?	❌No	❌No	✅Yes

---

4. Special Types

- **Parameters**: Variables passed to methods/constructors.
- **Constants**: Declared with `final` keyword (value cannot change).
- **Reference Variables**: Used to refer to objects.

```
final double PI = 3.14159;   // Constant
Student s1 = new Student();  // Reference variable
```

---

5. Best Practices

- Always **initialize local variables** before use.
- Use meaningful variable names (camelCase).
- Use `static final` for constants.
- Keep variable scope as **small as possible**.
