🛡️ Access Control in Java

Access to class members (fields and methods) is determined by the **access modifier** used.
Java provides **four access levels**:

Modifier	**Class**	**Package**	**Subclass (same pkg)**	**Subclass (diff pkg)**	**Other (World)**
public	✅	✅	✅	✅	✅
protected	✅	✅	✅	✅	❌
no modifier	✅	✅	✅	❌	❌
private	✅	❌	❌	❌	❌

- ✅ = accessible
- ❌ = not accessible

🔓 Public

- Accessible **from anywhere**, including outside the package.

🔒 Private

- Accessible **only within the same class**.

🟡 Default

- When no modifier is specified.
- Accessible **only within the same package**.
- Not accessible from outside the package, even in subclasses.

🟢 Protected

- Accessible:
    - Within the **same package**.
    - In **subclasses**, even if they are in **different packages**.

- But: When accessed from a subclass **outside the package**, the reference type must be the subclass itself (not the superclass!).
- In **different packages**, protected allows access **only in subclasses**, and **only through subclass-type references**.

Example:

// packageOne/Base.java

```
package packageOne;
public class Base {
   protected void display() {
      System.out.println("in Base");
   }
}
```

// packageTwo/Derived.java

```
package packageTwo;
import packageOne.Base;
public class Derived extends Base {
   public void show() {
      display();           // ✅ Allowed - inherited method
      new Derived().display(); // ✅ Allowed - Derived is subclass
      new Base().display();   // ❌ Error - Not allowed
   }
}
```

📦 In-Built Packages

Java includes essential packages:

- java.lang (automatically imported; basic types, Object, String, etc.)
- java.util (collections, utilities)
- java.io (file and stream I/O)
- Other standard packages for networking, concurrency, XML, etc.

▨ The Object Class

Every Java class inherits from java.lang.Object, providing core methods:

- **toString()**: prints a human-readable representation (usually overridden for clarity) .
- **hashCode()**: returns the integer representation of an object to identify the uniqueness of object.
- **equals(Object obj)**: checks logical equality; should be overridden alongside hashCode().
- **getClass()**: returns class information.
- **clone()**: creates object copies (must implement Cloneable).
- **finalize()**: called before garbage collection (deprecated—better not rely on it).

⚖ Comparison: == vs .equals()

- **==**: compares **reference identity**—whether two references point to the *exact same object*.
- **.equals()**: compares **logical equality**, based on object content.

instanceof Operator

Checks if an object is an instance of a specific class or interface (e.g., if (obj instanceof A)).