

♦ 1. Normal Class

A normal class allows you to create **multiple objects**.

```
class NormalClass {
    public void showMessage() {
        System.out.println("This is a normal class.");
    }
}

public class Main {
    public static void main(String[] args) {
        NormalClass obj1 = new NormalClass();
        NormalClass obj2 = new NormalClass();

        obj1.showMessage();
        obj2.showMessage();

        System.out.println(obj1 == obj2); // false, different objects
    }
}
```

♦ 2. Singleton Class

Private constructor make object creation out side of class impossible

Object should be static because we have static method and static method is there because in single ton object creation happen in the same class

First we call for instance creation in main method by using get instance then it will check if it is null or not

Null == new object creation

Not null == no object creation

Final class because we dont want inheritance and if single ton have more methods by inhertnace child class can have multiple instance which is opposite of what we want to make so make class final

If not by using super key word child can access singleton methods

A singleton class ensures **only one object is ever created**.

```
Final public class Singleton {
    private static Singleton instance;

    // private constructor prevents external instantiation
    private Singleton() {}

    // static method to return the same instance
    public static Singleton getInstance() {
        if (instance == null) {
            instance = new Singleton();           // lazy initialization
        }
        return instance;
    }

    public void showMessage() {
        System.out.println("This is a singleton class.");
    }
}

public class Main {
    public static void main(String[] args) {
        Singleton obj1 = Singleton.getInstance();
        Singleton obj2 = Singleton.getInstance();

        obj1.showMessage();
        obj2.showMessage();

        System.out.println(obj1 == obj2); // true, same object
    }
}
```

✓ Key Differences

Feature	Normal Class	Singleton Class
Object Creation	Multiple objects allowed	Only one object (single instance)

Constructor	<code>public</code>	<code>private</code>
Instance Access	Use <code>new</code> keyword	Use <code>getInstance()</code> method
Use Case	General-purpose classes	Configs, Logging, DB connection pools, etc.

Would you like me to show **thread-safe** Singleton or real-world use cases next?