LAB PROGRAMS: Local, Instance, and Static Variables in Java

Experiment 1: Local Variable

Aim

To understand the scope and lifetime of a local variable.

Program

```
class LocalVariableDemo {

    void show() {
        int x = 10;   // local variable
        System.out.println("Value of x: " + x);
    }

    public static void main(String[] args) {
        LocalVariableDemo obj = new LocalVariableDemo();
        obj.show();
    }
}
```

Expected Output

Value of x: 10

Observation

- Variable x exists only inside the method show().
- x cannot be accessed in main().

Experiment 2: Instance Variable

Aim

To understand that each object has its own copy of an instance variable.

Program

```
class InstanceVariableDemo {

   int count = 0;   // instance variable

   void increment() {
      count++;
      System.out.println("Count: " + count);
   }

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

      obj1.increment();
      obj1.increment();

      obj2.increment();
   }
}
```

Expected Output

Count: 1

Count: 2

Count: 1

Observation

- obj1 and obj2 maintain separate copies of count.
- Instance variables reset for each new object.

Experiment 3: Static Variable

Aim

To understand that a static variable is shared among all objects of a class.

Program

```
class StaticVariableDemo {

    static int count = 0;   // static variable

    void increment() {
        count++;
        System.out.println("Count: " + count);
    }

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

        obj1.increment();
        obj1.increment();

        obj2.increment();
    }
}
```

Expected Output

Count: 1

Count: 2

Count: 3

Observation

- Static variable count is shared across all objects.
- Value does not reset when a new object is created.

4: Comparison of Local, Instance, and Static Variables

Program

```
class VariableComparison {

   int instanceVar = 0;        // instance variable
   static int staticVar = 0;   // static variable

   void increment() {
      int localVar = 0;       // local variable

      localVar++;
      instanceVar++;
      staticVar++;

      System.out.println("Local: " + localVar);
      System.out.println("Instance: " + instanceVar);
      System.out.println("Static: " + staticVar);
      System.out.println("-----");
   }
```

```
    public static void main(String[] args) {
        VariableComparison obj1 = new VariableComparison();
        VariableComparison obj2 = new VariableComparison();

        obj1.increment();
        obj1.increment();
        obj2.increment();
    }
}
```

Expected Output

Local: 1

Instance: 1

Static: 1

Local: 1

Instance: 2

Static: 2

Local: 1

Instance: 1

Static: 3

Summary

Variable Type Behavior

Local	Resets every method call
Instance	Separate copy for each object
Static	Single shared copy for all objects
