

7. Writing a program in Java to verify the implementation of inner classes

ALGORITHM

Step 1: Start

Step 2: Define outer class and write display()

Step 3: Define inner class and write display()

Step 4: In main(), Create object for outer class and call outer display()

Step 5: Create object for inner class and call inner display()

Step 6: Stop

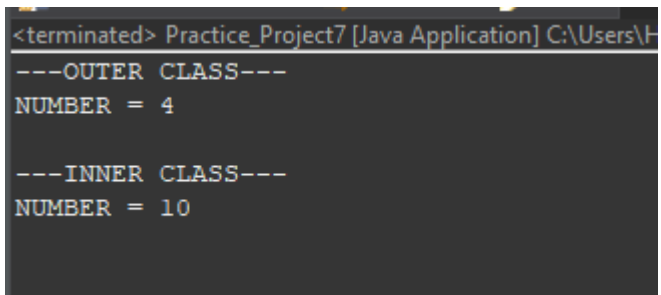
SOURCE CODE

```
package assistedPracticeProject;

class Outer_class //outer class
{
    private int m=4;
    private int n=10;
    public int display()
    {
        System.out.println("---OUTER CLASS---");
        return m;
    }
    public class Inner_class // inner class
    {
        public int display() {
            System.out.println("\n---INNER CLASS---");
            return n;
        }
    }
}
```

```
public class Practice_Project7
{
    public static void main(String args[]) {
        // Instantiating outer class
        Outer_class outer = new Outer_class();
        System.out.println("NUMBER = "+outer.display());
        // Instantiating inner class
        Outer_class.Inner_class inner = outer.new Inner_class();
        System.out.println("NUMBER = "+inner.display());
    }
}
```

OUTPUT



The screenshot shows a Java application window titled "<terminated> Practice_Project7 [Java Application] C:\Users\H...". The output is displayed in a dark-themed console window. It shows two sections: the first section is labeled "---OUTER CLASS---" and displays "NUMBER = 4"; the second section is labeled "---INNER CLASS---" and displays "NUMBER = 10".

```
<terminated> Practice_Project7 [Java Application] C:\Users\H...
---OUTER CLASS---
NUMBER = 4

---INNER CLASS---
NUMBER = 10
```