Experiment:5

1. **Inner Class:**
   Define two nested classes: Processor and RAM inside the outer class: CPU with following data members

```
class CPU {
    double price;
        class Processor{ // nested class
            double cores;
            double catch()
            String manufacturer;
            double getCache()
            void displayProcesorDetail()
        }
        protected class RAM{ // nested protected class
            // members of protected nested class
            double memory;
            String manufacturer;
            Double clockSpeed;
            double getClockSpeed()
            void displayRAMDetail()
        }
  }
```

   1. **Write appropriate Constructor and create instance of Outer and inner class and call the methods in main function Write a program to demonstrate usage of static inner class, local inner class and anonymous inner class**

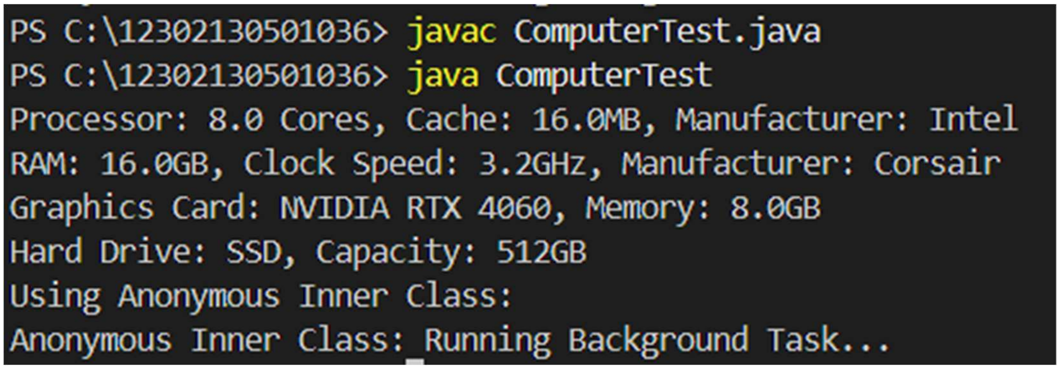
Solution :

```
class CPU {
  double price;

  public CPU(double price) {
    this.price = price;
  }

  class Processor {
    double cores;
    double cache;
    String manufacturer;

    public Processor(double cores, double cache, String manufacturer) {
        this.cores = cores;
        this.cache = cache;
        this.manufacturer = manufacturer;
    }

    double getCache() {
        return cache;
    }

    void displayProcessorDetail() {
        System.out.println("Processor: " + cores + " Cores, Cache: " + cache + "MB,
Manufacturer: " + manufacturer);
    }

    protected class RAM {
        double memory;
        String manufacturer;
        double clockSpeed;

        public RAM(double memory, String manufacturer, double clockSpeed) {
            this.memory = memory;
            this.manufacturer = manufacturer;
            this.clockSpeed = clockSpeed;
        }

        double getClockSpeed() {
            return clockSpeed;
        }

        void displayRAMDetail() {
            System.out.println("RAM: " + memory + "GB, Clock Speed: " + clockSpeed +
"GHz, Manufacturer: " + manufacturer);
```

```
            }
        }
    }

    static class GraphicsCard {
        String name;
        double memory;

        public GraphicsCard(String name, double memory) {
            this.name = name;
            this.memory = memory;
        }

        void displayGraphicsCardDetail() {
            System.out.println("Graphics Card: " + name + ", Memory: " + memory + "GB");
        }
    }

    void showLocalInnerClass() {
        class HardDrive {
            String type;
            int capacity;

            public HardDrive(String type, int capacity) {
                this.type = type;
                this.capacity = capacity;
            }

            void displayHardDriveDetail() {
                System.out.println("Hard Drive: " + type + ", Capacity: " + capacity + "GB");
            }
        }

        HardDrive hd = new HardDrive("SSD", 512);
        hd.displayHardDriveDetail();
    }
}
```

```
public class ComputerTest {
    public static void main(String[] args) {
        CPU myCPU = new CPU(350.50);

        CPU.Processor myProcessor = myCPU.new Processor(8, 16, "Intel");
        myProcessor.displayProcessorDetail();

        CPU.Processor.RAM myRAM = myProcessor.new RAM(16, "Corsair", 3.2);
        myRAM.displayRAMDetail();

        CPU.GraphicsCard myGPU = new CPU.GraphicsCard("NVIDIA RTX 4060", 8);
        myGPU.displayGraphicsCardDetail();

        myCPU.showLocalInnerClass();

        System.out.println("Using Anonymous Inner Class:");
        Runnable task = new Runnable() {
            @Override
            public void run() {
                System.out.println("Anonymous Inner Class: Running Background Task...");
            }
        };
        task.run();
    }
}
```

Output:



```
PS C:\12302130501036> javac ComputerTest.java
PS C:\12302130501036> java ComputerTest
Processor: 8.0 Cores, Cache: 16.0MB, Manufacturer: Intel
RAM: 16.0GB, Clock Speed: 3.2GHz, Manufacturer: Corsair
Graphics Card: NVIDIA RTX 4060, Memory: 8.0GB
Hard Drive: SSD, Capacity: 512GB
Using Anonymous Inner Class:
Anonymous Inner Class: Running Background Task...
```