

## Java Inheritance Practice - Class Level Questions

### Question 8: Employee and Manager (Single-Level Inheritance)

Q: Write a Java program where:

- Class Employee has method work()
- Class Manager inherits Employee and has method attendMeeting()
- Create an object of Manager and call both methods.

```
class Employee {  
    void work() {  
        System.out.println("Employee is working");  
    }  
}  
  
class Manager extends Employee {  
    void attendMeeting() {  
        System.out.println("Manager is attending a meeting");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Manager m = new Manager();  
        m.work();  
        m.attendMeeting();  
    }  
}
```

### Question 9: Person Student Monitor (Multi-Level Inheritance)

Q: Write a Java program where:

- Class Person has method displayName()
- Class Student inherits Person and adds method displayClass()
- Class Monitor inherits Student and adds method checkDiscipline()
- Create an object of Monitor and call all three methods.

```
class Person {  
    void displayName() {  
        System.out.println("Name: Aryan");  
    }  
}  
  
class Student extends Person {  
    void displayClass() {  
        System.out.println("Class: 12th");  
    }  
}  
  
class Monitor extends Student {  
    void checkDiscipline() {  
        System.out.println("Monitor is checking discipline");  
    }  
}
```

```

}

public class Main {
    public static void main(String[] args) {
        Monitor mon = new Monitor();
        mon.displayName();
        mon.displayClass();
        mon.checkDiscipline();
    }
}

```

### **Question 10: Shape and Circle (Single-Level Inheritance)**

Q: Write a Java program where:

- Class Shape has method draw()
- Class Circle inherits Shape and has method calculateArea()
- Create an object of Circle and call both methods.

```

class Shape {
    void draw() {
        System.out.println("Drawing a shape");
    }
}

class Circle extends Shape {
    void calculateArea() {
        System.out.println("Area of circle = " + r * r);
    }
}

public class Main {
    public static void main(String[] args) {
        Circle c = new Circle();
        c.draw();
        c.calculateArea();
    }
}

```

### **Question 11: Device Laptop GamingLaptop (Multi-Level Inheritance)**

Q: Write a Java program where:

- Class Device has method powerOn()
- Class Laptop inherits Device and has method boot()
- Class GamingLaptop inherits Laptop and has method startGame()
- Create an object of GamingLaptop and call all methods.

```

class Device {
    void powerOn() {
        System.out.println("Device is powered on");
    }
}

class Laptop extends Device {
    void boot() {
        System.out.println("Laptop is booting");
    }
}

class GamingLaptop extends Laptop {
    void startGame() {
        System.out.println("GamingLaptop is starting game");
    }
}

```

```

        }
    }

class GamingLaptop extends Laptop {
    void startGame() {
        System.out.println("Game is starting on Gaming Laptop");
    }
}

public class Main {
    public static void main(String[] args) {
        GamingLaptop g = new GamingLaptop();
        g.powerOn();
        g.boot();
        g.startGame();
    }
}

```

### **Question 12: Book and Novel (Single-Level Inheritance)**

Q: Write a Java program where:

- Class Book has method read()
- Class Novel inherits Book and has method storyLine()
- Create an object of Novel and call both methods.

```

class Book {
    void read() {
        System.out.println("Reading a book");
    }
}

class Novel extends Book {
    void storyLine() {
        System.out.println("The novel has an interesting storyline");
    }
}

public class Main {
    public static void main(String[] args) {
        Novel n = new Novel();
        n.read();
        n.storyLine();
    }
}

```