Task-2

4.Create a class to represent a student with attributes like name, age, and Grade

```
public class Student {
    private String name;
    private int age;
    private String grade;

    public Student(String name, int age, String grade) {
        this.name = name;
        this.age = age;
        this.grade = grade;
    }

    @Override
    public String toString() {
        return "Name: " + name + ", Age: " + age + ", Grade: " + grade;
    }

    public static void main(String[] args) {
        Student student1 = new Student("Alice", 18, "A");
        Student student2 = new Student("Bob", 17, "B");

        System.out.println(student1);
        System.out.println(student2);
    }
}
```

4. Implement inheritance by creating subclasses for different types of students (e.g., undergraduate, postgraduate).

```
class Student {
    private String name;
    private int age;
    private String grade;

    public Student(String name, int age, String grade) {
        this.name = name;
        this.age = age;
        this.grade = grade;
    }

    @Override
    public String toString() {
        return "Name: " + name + ", Age: " + age + ", Grade: " + grade;
    }
}

class UndergraduateStudent extends Student {
    private int year;

    public UndergraduateStudent(String name, int age, String grade, int year) {
        super(name, age, grade);
        this.year = year;
    }

    @Override
    public String toString() {
        return super.toString() + ", Year: " + year;
    }
}

class PostgraduateStudent extends Student {
    private String program;

    public PostgraduateStudent(String name, int age, String grade, String program) {
        super(name, age, grade);
        this.program = program;
    }

    @Override
    public String toString() {
        return super.toString() + ", Program: " + program;
    }
}

public class Main {
    public static void main(String[] args) {
        UndergraduateStudent undergrad = new UndergraduateStudent("Alice", 20, "A", 2);
        PostgraduateStudent postgrad = new PostgraduateStudent("Bob", 25, "B", "Computer Science");

        System.out.println(undergrad);
        System.out.println(postgrad);
    }
}
```

6.Utilize encapsulation to set and get student information securely.

```
public class Student {
    private String name;
    private int age;
    private String grade;

    public Student(String name, int age, String grade) {
        this.name = name;
        this.age = age;
        this.grade = grade;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        if (age >= 0 && age <= 150) { // Assuming a reasonable age range
            this.age = age;
        } else {
            System.out.println("Invalid age.");
        }
    }

    public String getGrade() {
        return grade;
    }

    public void setGrade(String grade) {
        this.grade = grade;
    }

    @Override
    public String toString() {
        return "Name: " + name + ", Age: " + age + ", Grade: " + grade;
    }
}
```