ULAB

UNIVERSITY OF LIBERAL ARTS
BANGLADESH

Course Code: CSE 2104
Course Title: Object-oriented Programming Lab
Lab Work 1

SUBMITTED TO :

Shakib Mahamud Dipto
CSE
lecturer
University of Liberal Arts

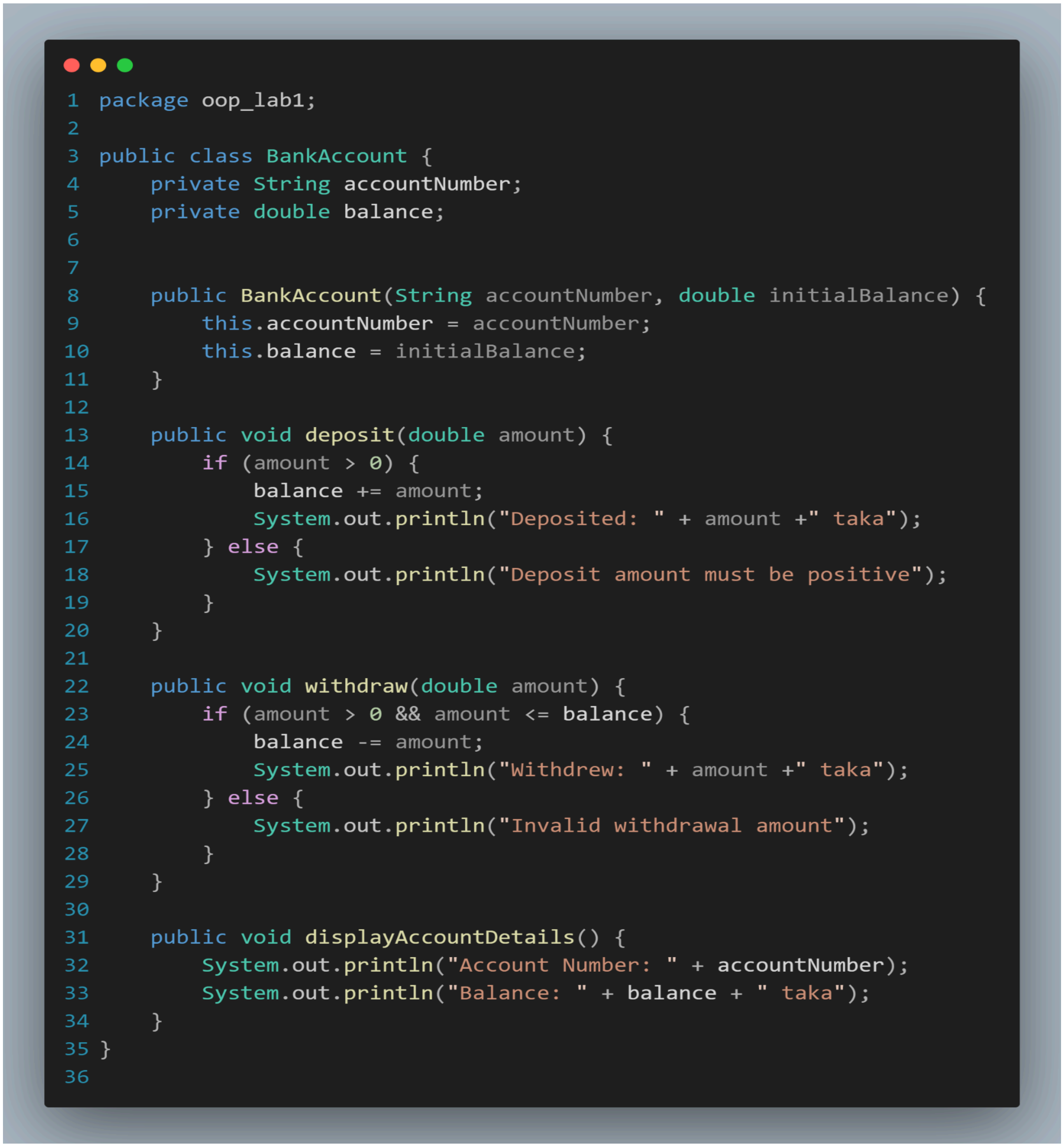
SUBMITTED BY :

Piyas sarkar
223014024
Section 4

Problem-1:

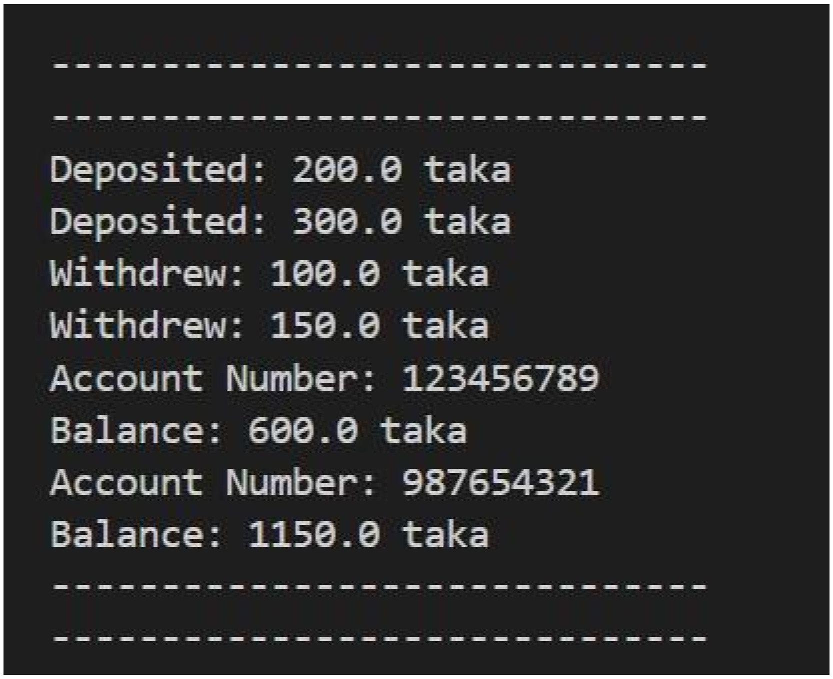
Create a class called BankAccount with the instance variables accountNumber and balance. Add methods to deposit and withdraw money from the account. Create objects in BankAccount and perform deposit and withdrawal operations.

Solution Code:



```java
package oop_lab1;

public class BankAccount {
    private String accountNumber;
    private double balance;


    public BankAccount(String accountNumber, double initialBalance) {
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount +" taka");
        } else {
            System.out.println("Deposit amount must be positive");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: " + amount +" taka");
        } else {
            System.out.println("Invalid withdrawal amount");
        }
    }

    public void displayAccountDetails() {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Balance: " + balance + " taka");
    }
}

```

Code Output:



```
------------------------------
------------------------------
Deposited: 200.0 taka
Deposited: 300.0 taka
Withdrew: 100.0 taka
Withdrew: 150.0 taka
Account Number: 123456789
Balance: 600.0 taka
Account Number: 987654321
Balance: 1150.0 taka
------------------------------
------------------------------
```

Code explanation:

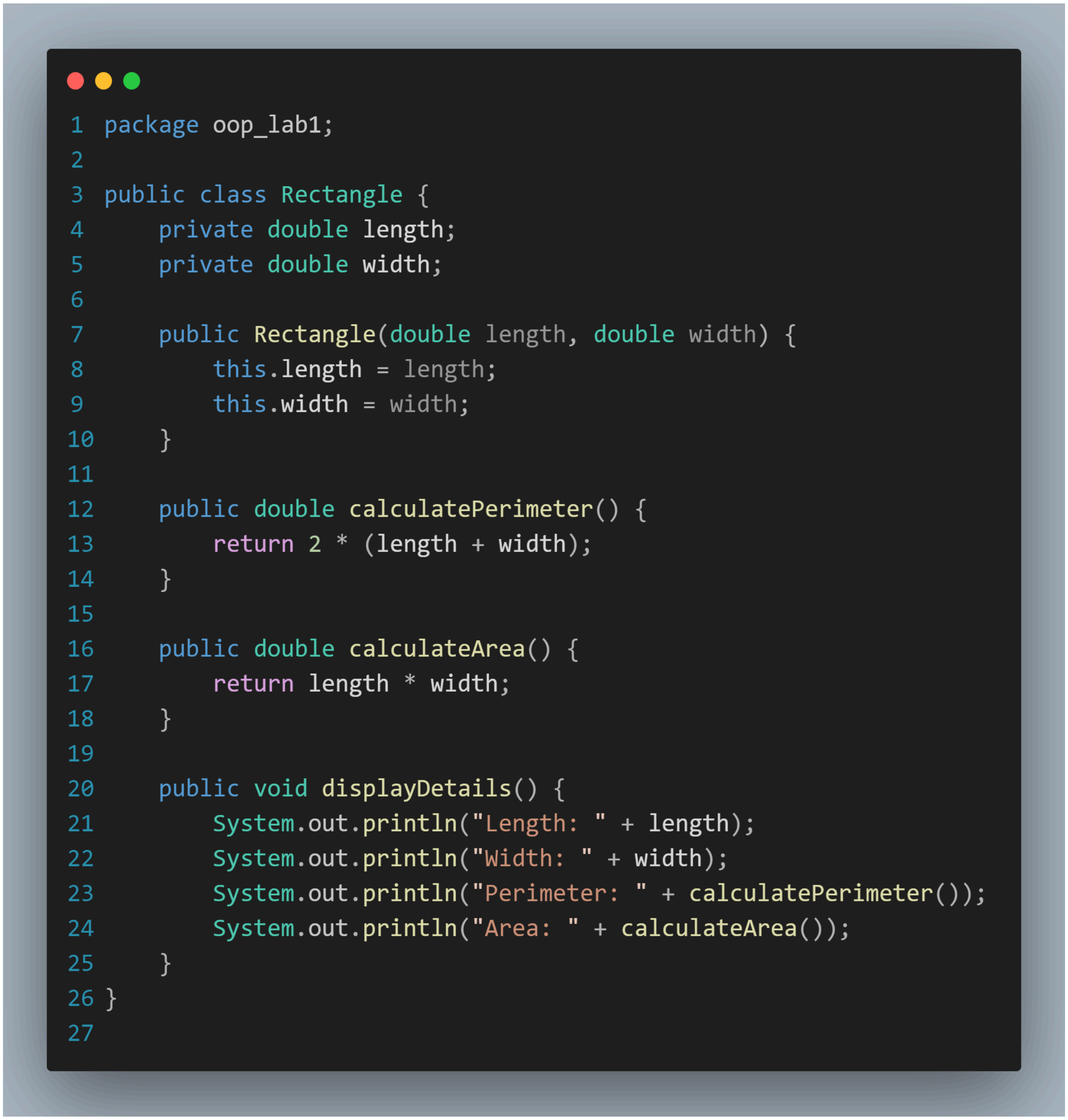
BankAccount:

With crucial instance variables like the account number and amount, the BankAccount class mimics the operations of a standard bank account. Its techniques make money transactions easier, such as making deposits and withdrawals. The class needs the account number and balance to be initialized at instantiation. By adding money to the account, the deposit method guarantees that the amount deposited is legitimate. On the other hand, if the account has enough balance and the withdrawal amount is legitimate, the withdrawal method allows for cash extraction. The getBalance function lets clients view the current balance, guaranteeing account management and transparency.

Problem-2:

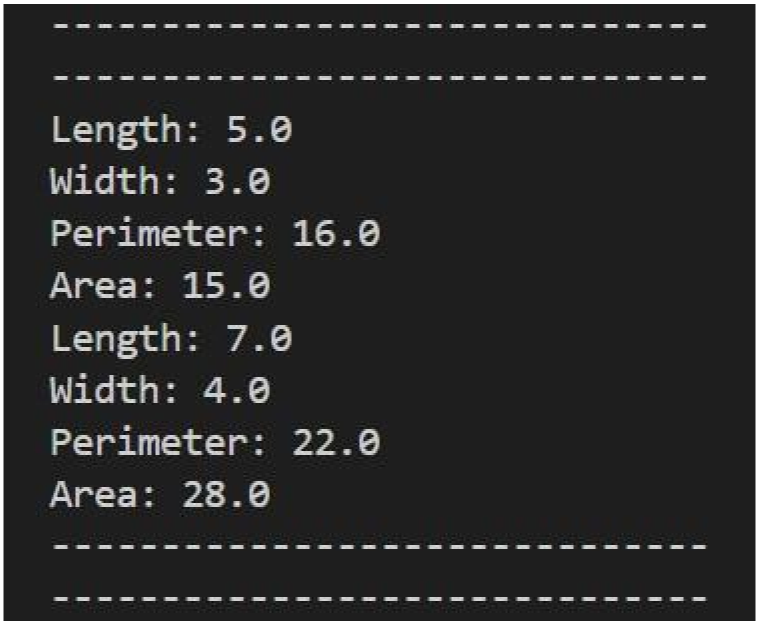
Create a class rectangle with properties such as length and width. Add methods to calculate the perimeter and area of the rectangle. Create objects and display their corresponding perimeter and area.

Solution Code:



```java
package oop_lab1;

public class Rectangle {
    private double length;
    private double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    public double calculatePerimeter() {
        return 2 * (length + width);
    }

    public double calculateArea() {
        return length * width;
    }

    public void displayDetails() {
        System.out.println("Length: " + length);
        System.out.println("Width: " + width);
        System.out.println("Perimeter: " + calculatePerimeter());
        System.out.println("Area: " + calculateArea());
    }
}
```

Code Output:



```
------------------------------
------------------------------
Length: 5.0
Width: 3.0
Perimeter: 16.0
Area: 15.0
Length: 7.0
Width: 4.0
Perimeter: 22.0
Area: 28.0
------------------------------
------------------------------
```

Code explanation:

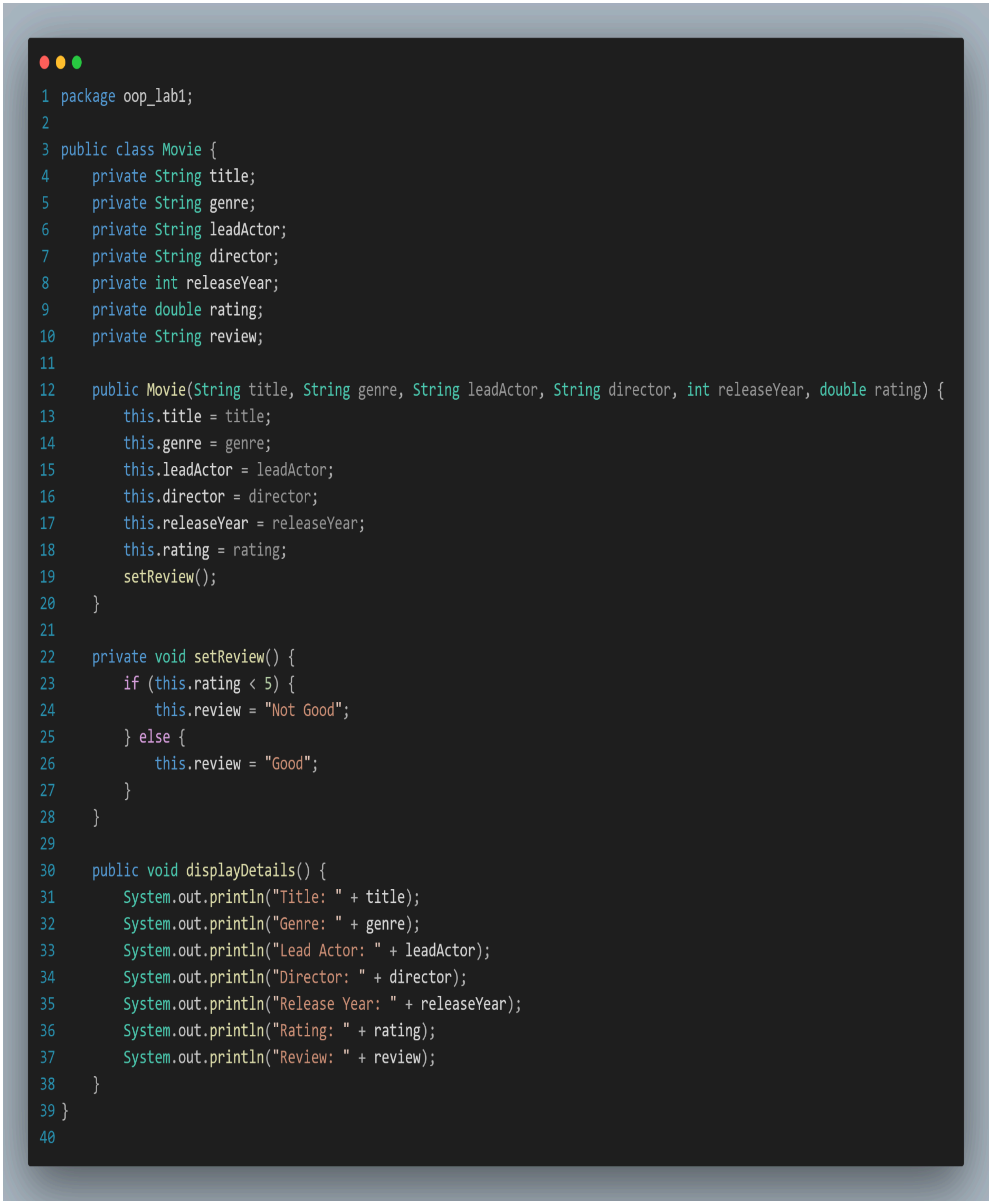
Rectangle:

Geometric concepts are shown in the Rectangle class, utilizing a rectangular shape with attributes such as width and length. A rectangle object must have its dimensions specified when it is created. This class has methods that allow you to calculate the area and perimeter of the rectangle. While the calculateArea technique uses (length*width) to compute the area, the calculatePerimeter method uses the formula 2×(length+width) to calculate the perimeter. The class encourages modularity and abstraction by encapsulating these computations, making geometric calculations easier to use and understand.

Problem-3:

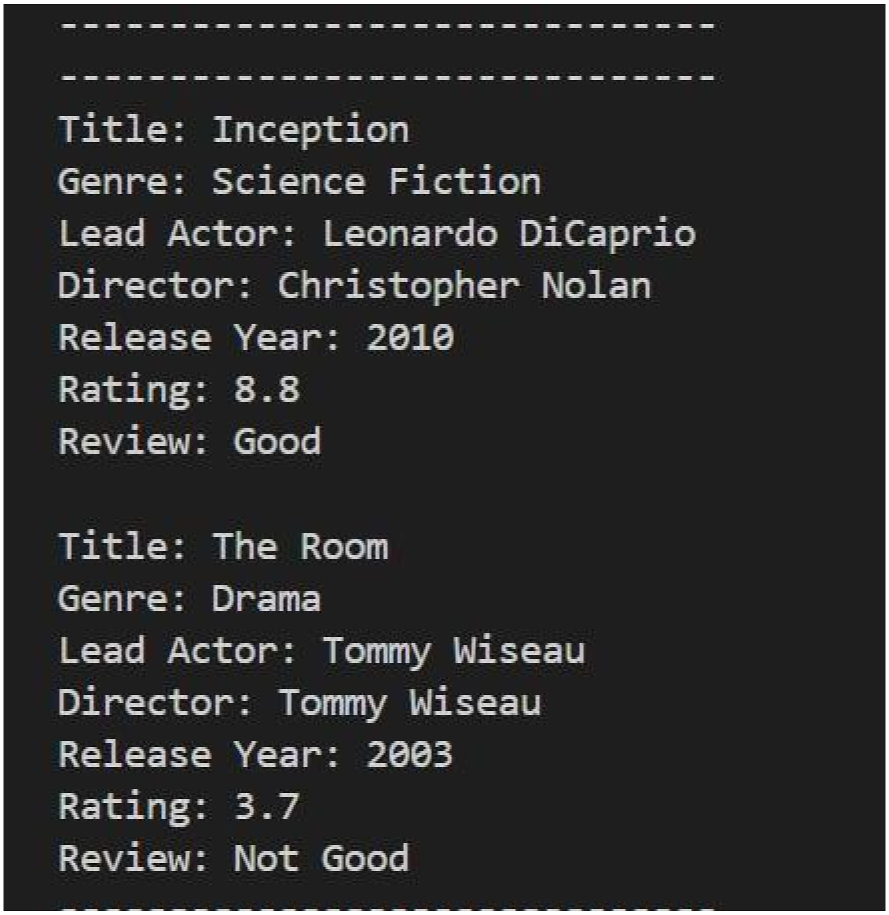
Create a class called Movie that has properties such as title, genre, lead actor, director, release year, rating, and review. Create two movie objects and display their properties. If the rating is <5, the review should be “Not Good”. Otherwise, the review would be “Good”.

Solution Code:



```java
package oop_lab1;

public class Movie {
    private String title;
    private String genre;
    private String leadActor;
    private String director;
    private int releaseYear;
    private double rating;
    private String review;

    public Movie(String title, String genre, String leadActor, String director, int releaseYear, double rating) {
        this.title = title;
        this.genre = genre;
        this.leadActor = leadActor;
        this.director = director;
        this.releaseYear = releaseYear;
        this.rating = rating;
        setReview();
    }

    private void setReview() {
        if (this.rating < 5) {
            this.review = "Not Good";
        } else {
            this.review = "Good";
        }
    }

    public void displayDetails() {
        System.out.println("Title: " + title);
        System.out.println("Genre: " + genre);
        System.out.println("Lead Actor: " + leadActor);
        System.out.println("Director: " + director);
        System.out.println("Release Year: " + releaseYear);
        System.out.println("Rating: " + rating);
        System.out.println("Review: " + review);
    }
}

```

Code Output:



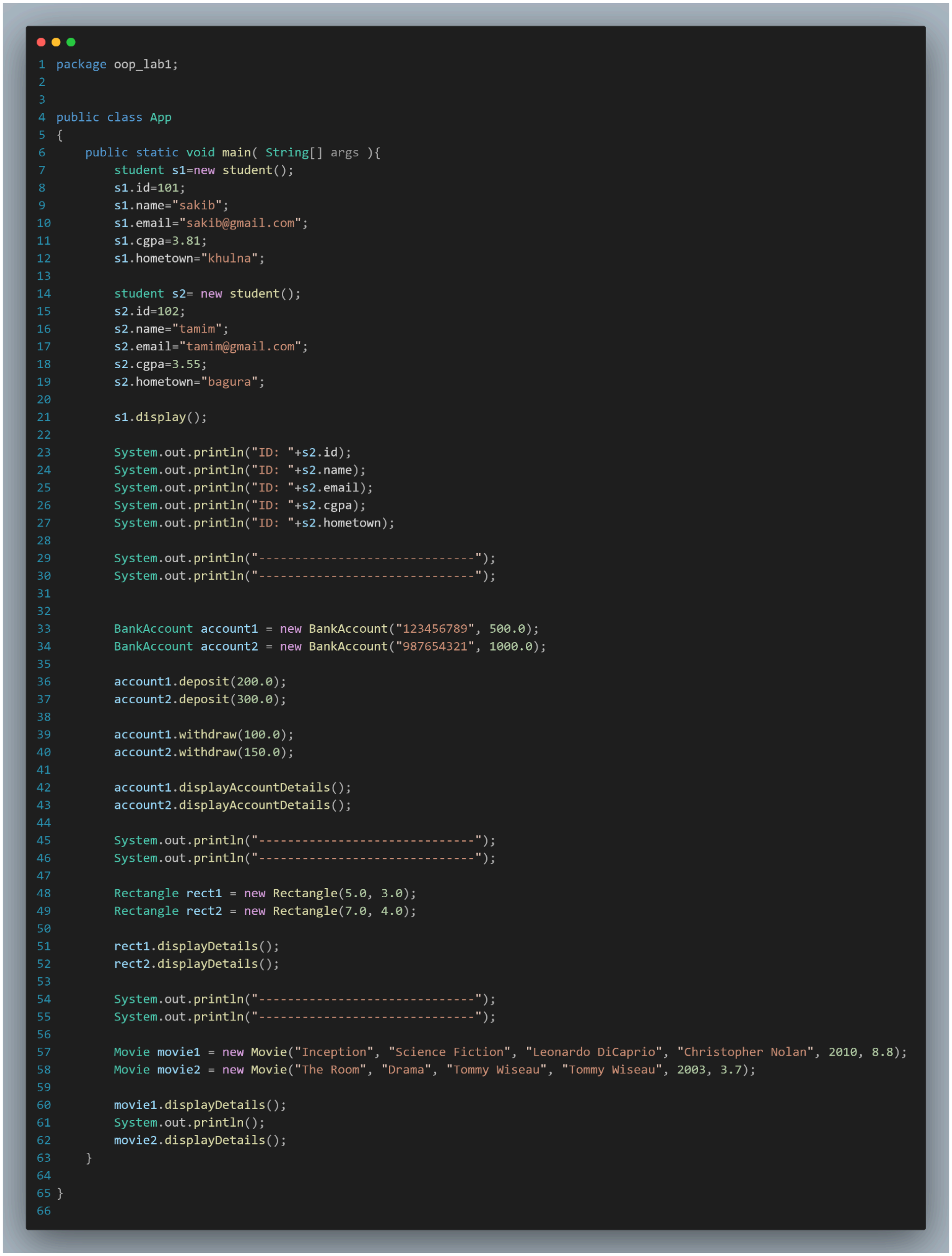
```
------------------------------
------------------------------
Title: Inception
Genre: Science Fiction
Lead Actor: Leonardo DiCaprio
Director: Christopher Nolan
Release Year: 2010
Rating: 8.8
Review: Good

Title: The Room
Genre: Drama
Lead Actor: Tommy Wiseau
Director: Tommy Wiseau
Release Year: 2003
Rating: 3.7
Review: Not Good
------------------------------
```

Code explanation:

Movie:

The Movie class is used to construct a cohesive object containing all the important information about a movie. The title, genre, director, star, year of release and rating are only a few of its attributes. Upon instantiation, these properties are filled in, enabling a full movie representation. The class method displayProperties presents these attributes logically and gives a thorough synopsis of the main plot of the film. The class rates the rating as well, designating it as "good" if it is greater than a 5 and "Not Good" otherwise to provide a succinct review. The class is more beneficial for programmatic movie critique and analysis because of this sort of capacity.

Main Class (App) Code:



```java
package oop_lab1;


public class App
{
    public static void main( String[] args ){
        student s1=new student();
        s1.id=101;
        s1.name="sakib";
        s1.email="sakib@gmail.com";
        s1.cgpa=3.81;
        s1.hometown="khulna";

        student s2= new student();
        s2.id=102;
        s2.name="tamim";
        s2.email="tamim@gmail.com";
        s2.cgpa=3.55;
        s2.hometown="bagura";

        s1.display();

        System.out.println("ID: "+s2.id);
        System.out.println("ID: "+s2.name);
        System.out.println("ID: "+s2.email);
        System.out.println("ID: "+s2.cgpa);
        System.out.println("ID: "+s2.hometown);

        System.out.println("------------------------------");
        System.out.println("------------------------------");


        BankAccount account1 = new BankAccount("123456789", 500.0);
        BankAccount account2 = new BankAccount("987654321", 1000.0);

        account1.deposit(200.0);
        account2.deposit(300.0);

        account1.withdraw(100.0);
        account2.withdraw(150.0);

        account1.displayAccountDetails();
        account2.displayAccountDetails();

        System.out.println("------------------------------");
        System.out.println("------------------------------");

        Rectangle rect1 = new Rectangle(5.0, 3.0);
        Rectangle rect2 = new Rectangle(7.0, 4.0);

        rect1.displayDetails();
        rect2.displayDetails();

        System.out.println("------------------------------");
        System.out.println("------------------------------");

        Movie movie1 = new Movie("Inception", "Science Fiction", "Leonardo DiCaprio", "Christopher Nolan", 2010, 8.8);
        Movie movie2 = new Movie("The Room", "Drama", "Tommy Wiseau", "Tommy Wiseau", 2003, 3.7);

        movie1.displayDetails();
        System.out.println();
        movie2.displayDetails();
    }

}

```