/\*Question 1: Bank Account Management

Create a class BankAccount with the following private fields:

accountNumber (String)

accountHolderName (String)

balance (double)

Requirements:

Provide public getter and setter methods.

Allow deposit and withdrawal using methods.

Ensure balance cannot be negative.

Prevent withdrawal if the balance is insufficient.\*/

class BankAccount {

//private fields

private String accountNumber;

private String accountHolderName;

private double balance;

//constructor

public BankAccount(String accountNumber,String accountHolderName, double balance){

this.accountNumber = accountNumber;

this.accountHolderName = accountHolderName;

if (balance >= 0){

this.balance = balance;

}else{

this.balance = 0;

System.out.println("Initial balance cannot be negative. Setting to 0.");

}

}

// Getters and Setters

public String getAccountNumber(){

return accountNumber;

}

public void setAccountNumber(String AccountNumber) {

this.accountNumber = accountNumber;

}

public String getAccountHolderName(String accountHolderName){

return accountHolderName;

}

public void setAccountHolderName(String accountHolderName){

this.accountHolderName = accountHolderName;

}

public double getBalance(){

return balance;

}

// Deposit method

public void deposit(double amount){

if(amount > 0){

balance += amount;

System.out.println("Deposited : " +amount);

}else{

System.out.println("Deposit amount must be positive.");

}

}

// Withdraw method

public void withdraw(double amount){

if(amount > 0){

if(balance >= amount){

balance -= amount;

System.out.println("Withdrawn:" +amount);

}else{

System.out.println("Insufficient balance");

}

}else{

System.out.println("Withdrawal amount must be positive.");

}

}

//Display method

public void displayAccountDetails(){

System.out.println("Account Number : "+accountNumber);

System.out.println("Account Holder : " +accountHolderName);

System.out.println("Current Balance : " +balance);

}

}

//Main class

public class BankTest {

public static void main(String[] args){

// create account

BankAccount acc = new BankAccount("1243446565","Sowbarnika",1000);

//Display details

acc.displayAccountDetails();

//Deposit details

acc.deposit(500);

//Try withdrawal

acc.withdraw(200);

acc.withdraw(1500);

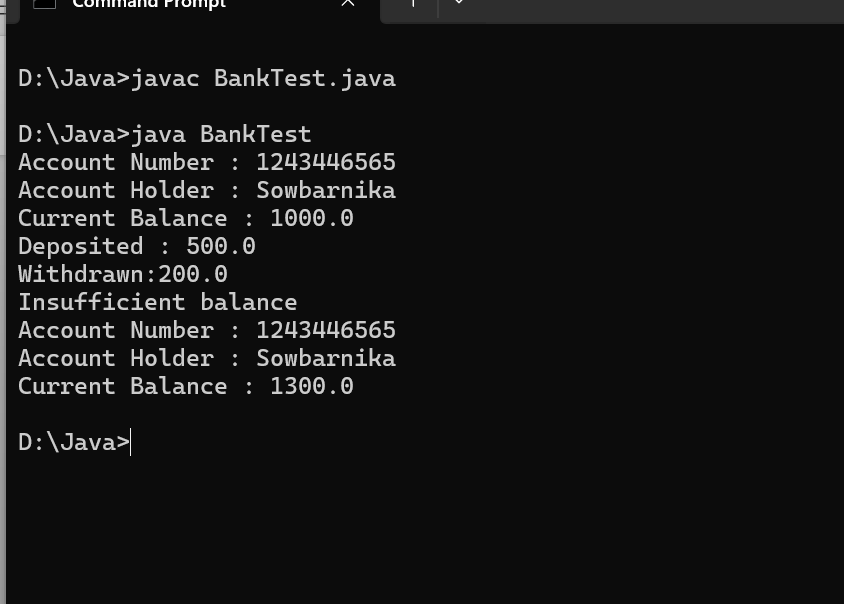
// Final details

acc.displayAccountDetails();

}

}

**OUTPUT:**



/\*Question 2: Student Report Card

Create a class Student with the following private fields:

name (String)

rollNumber (int)

marks (float)

Requirements:

Use getter and setter methods to access fields.

Restrict marks between 0 and 100.

Add a method to display student details.\*/

class Student {

//Private attributes

private String name;

private int rollNumber;

private float marks;

//constructor

public Student(String name, int rollNumber, float marks){

this.name = name;

this.rollNumber = rollNumber;

setMarks(marks);

}

//Getter and Setter for name

public String getname(){

return name;

}

public void setName(String name){

this.name = name;

}

//Getter and setter for rollNumber

public int getrollNumber(){

return rollNumber;

}

public void setrollNumber(){

this.rollNumber = rollNumber;

}

//Getter and setter for marks

public void setMarks(float marks){

if(marks >=0 && marks<=100){

this.marks = marks;

}else{

this.marks = 0;

System.out.println("Invalid marks. Setting marks to 0.");

}

}

// Display student details

public void displayStudentDetails(){

System.out.println("Name : " +name);

System.out.println("Roll Number : " +rollNumber);

System.out.println("Marks : " +marks);

}

}

public class StudentApp{

public static void main(String args[]){

// Create student object

Student student1 = new Student("Sowbarnika", 101, 95.5f);

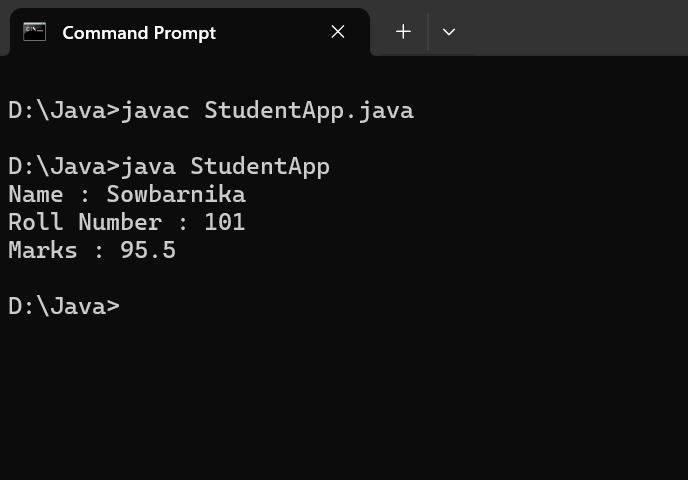
//Display details

student1.displayStudentDetails();

}

}

**OUTPUT :**



/\*Question 3: Product Catalog

Create a class Product with private fields:

productId (int)

productName (String)

price (double)

Requirements:

Implement getters and setters with validation.

Price must be greater than 0.

Add a method to display product details.\*/

class Product{

//Private attributes

private int productId;

private String productName;

private double price;

//constructor

public Product(int productId,String productName,double price){

this.productId = productId;

this.productName = productName;

setPrice(price);

}

// Getter and setter for productId

public int getProductId(){

return productId;

}

public void setProductId(int productId){

this.productId = productId;

}

//Getter and setter for productName

public String getProductName(){

return productName;

}

public void setProductName(String productName){

this.productName = productName;

}

//Getter and Setter for price

public double getPrice(){

return price;

}

public void setPrice(double price){

if(price > 0){

this.price = price;

}else{

this.price = 0;

System.out.println("Invalid price. Setting price to 0.");

}

}

//Method to display product details

public void displayProductDetails(){

System.out.println("Product ID : " +productId);

System.out.println("Product Name : " +productName);

System.out.println("Price : Rs." +price);

}

}

public class ProductApp{

public static void main(String args[]){

// create object

Product product = new Product(1001,"Laptop",49999.99);

product.displayProductDetails();

}

}

**OUTPUT :**

