OOPS PRACTICAL - 4

Name : Prathamesh Rajbhoj

Branch : CSE

Roll No : A - 53

Aim :

Write a program to implement multiple inheritance.

Consider a class BankAccount with data members as account number, aadhar number, owner name, ROI and balance with member functions openAccount(), deposit(amount),closeAccount() and updateInterest(). Create an interface Debitable which has method withdraw().

Derive a class FixedDepositAccount from BankAccount having data member lockInPeriod. Override methods updateInterest() to update Simple Interest, and method closeAccount() to charge 5 % for closure of FD Account before lockInPeriod.

Derive a class SavingAccount from class BankAccount and interface Debitable.

[ROI for Saving Account is 4% and for FD – 1-2yrs-6% ; 2-5yrs-6.5% ; >5yrs- 7%]

**BankAccount .java :**

public class BankAccount {

String account\_number;

String aadhar\_number;

String owner\_name;

double ROI;

double balance;

boolean account\_status = false;

BankAccount() {

}

BankAccount(String account\_number, String aadhar\_number, String owner\_name) {

this.account\_number = account\_number;

this.aadhar\_number = aadhar\_number;

this.owner\_name = owner\_name;

this.account\_status=true;

System.out.println("\n\nCongartulations !!! \nYour New Account is ready for Transactions\n\n");

}

void openAccount(String owner\_name, String aadhar\_number) {

String acc\_num = "";

for (int i = 0; i < 8; i++) {

int xnum = (int) ((Math.random() \* 10));

acc\_num = acc\_num + Integer.toString(xnum);

}

this.balance = 0;

this.account\_status = true;

}

void deposit(double amount) {

if(this.account\_status==false){

System.out.println("Account was Closed.....");

System.out.println("Please Open New Account for Transactions !!\n\n");

return;

}

System.out.println("\nAdding Rs " + amount + " to your Account\n");

this.balance += amount;

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

}

void display\_account() {

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

System.out.println();

System.out.println("Account Owner : " + this.owner\_name);

System.out.println("Available Balance : " + this.balance + " Rs");

System.out.println("Account Number : " + this.account\_number);

System.out.println("Account Status : " + (this.account\_status==true?" Active":" Closed"));

System.out.println("Account ROI : " + this.ROI);

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

System.out.println();

}

}

**FixedDepositAccount .txt :**

public class FixedDepositAccount extends BankAccount {

int lockInPeriod;

FixedDepositAccount() {

}

FixedDepositAccount(String account\_number, String aadhar\_number, String owner\_name, int amount, int lockInPeriod) {

super(account\_number, aadhar\_number, owner\_name);

this.balance=amount;

this.ROI = 0;

this.lockInPeriod = lockInPeriod;

}

void updateInterest(int year) {

if(this.account\_status==false){

System.out.println("Account was Closed.....");

System.out.println("Please Open New Account for Transactions !!\n\n");

return;

}

if (year > 0 && year <= 2) {

this.ROI = 6;

}

else if (year > 2 && year <= 5) {

this.ROI = 6.5;

}

else if (year > 5) {

this.ROI = 7;

}

else {

System.out.println("Invalid Year");

return;

}

this.balance += (this.ROI / 100) \* this.balance;

System.out.println("Interest Updated");

}

void closeAccount(int year) {

if(year < this.lockInPeriod){

this.balance = this.balance-((5/100)\*(this.balance));

System.out.println("Account Closed Before Lockin Period");

System.out.println("Deducting 5% from Balance...\n\n");

}

System.out.println("Account Closed Succesfully !!!\n");

this.account\_status = false;

}

void display\_account() {

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

System.out.println();

System.out.println("Account Owner : " + this.owner\_name);

System.out.println("Available Balance : " + this.balance + " Rs");

System.out.println("Account Number : " + this.account\_number);

System.out.println("Account Status : " + (this.account\_status==true?" Active":" Closed"));

System.out.println("Account ROI : " + this.ROI);

System.out.println("Account LockinPeroid : " + this.lockInPeriod);

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

System.out.println();

}

}

**SavingAccount .txt :**

public class SavingAccount extends BankAccount {

SavingAccount() {

}

SavingAccount(String account\_number, String aadhar\_number, String owner\_name) {

super(account\_number, aadhar\_number, owner\_name);

this.ROI = 4;

}

void updateInterest(int year) {

if(this.account\_status==false){

System.out.println("Account was Closed.....");

System.out.println("Please Open New Account for Transactions !!\n\n");

return;

}

if (year > 0) {

this.ROI = 4;

this.balance = (this.ROI / 100) \* this.balance;

System.out.println("Interest Updated");

}

else {

System.out.println("Invalid Year");

}

}

void closeAccount() {

this.account\_status = false;

}

void withdraw(double amount) {

if(this.account\_status==false){

System.out.println("Account was Closed.....");

System.out.println("Please Open New Account for Transactions !!\n\n");

return;

}

if(amount>this.balance){

System.out.println("Not Sufficient Amount Present in Account");

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

}

else{

this.balance-=amount;

System.out.println("Debited Rs " + amount + " from your Account\n");

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

}

}

}

**Debitable.java :**

interface Debitable {

void withdraw();

}

**Test.java :**

public class Test {

public static void main(String[] args) {

System.out.println();

System.out.println();

System.out.println();

FixedDepositAccount fd1 = new FixedDepositAccount("Acc-101","Adhar-101","Owner-101",1000,5);

fd1.display\_account();

fd1.updateInterest(1);

fd1.updateInterest(2);

fd1.updateInterest(3);

fd1.deposit(1000);

fd1.updateInterest(4);

fd1.display\_account();

fd1.closeAccount(2);

fd1.display\_account();

fd1.deposit(600);

SavingAccount sa1=new SavingAccount("Acc-201","Adhar-201","Owner-201");

sa1.display\_account();

sa1.deposit(1000);

sa1.withdraw(400);

sa1.display\_account();

sa1.closeAccount();

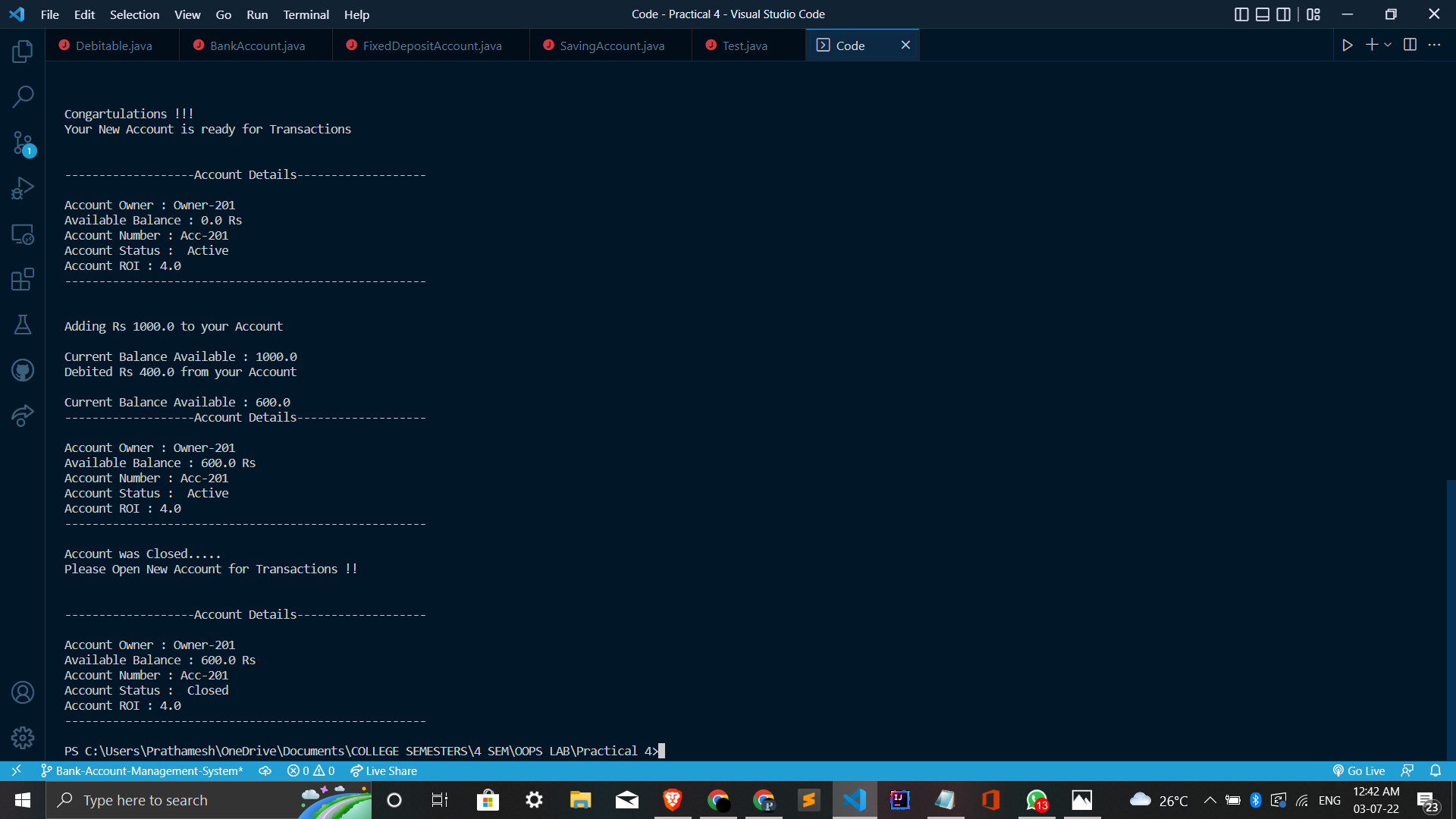
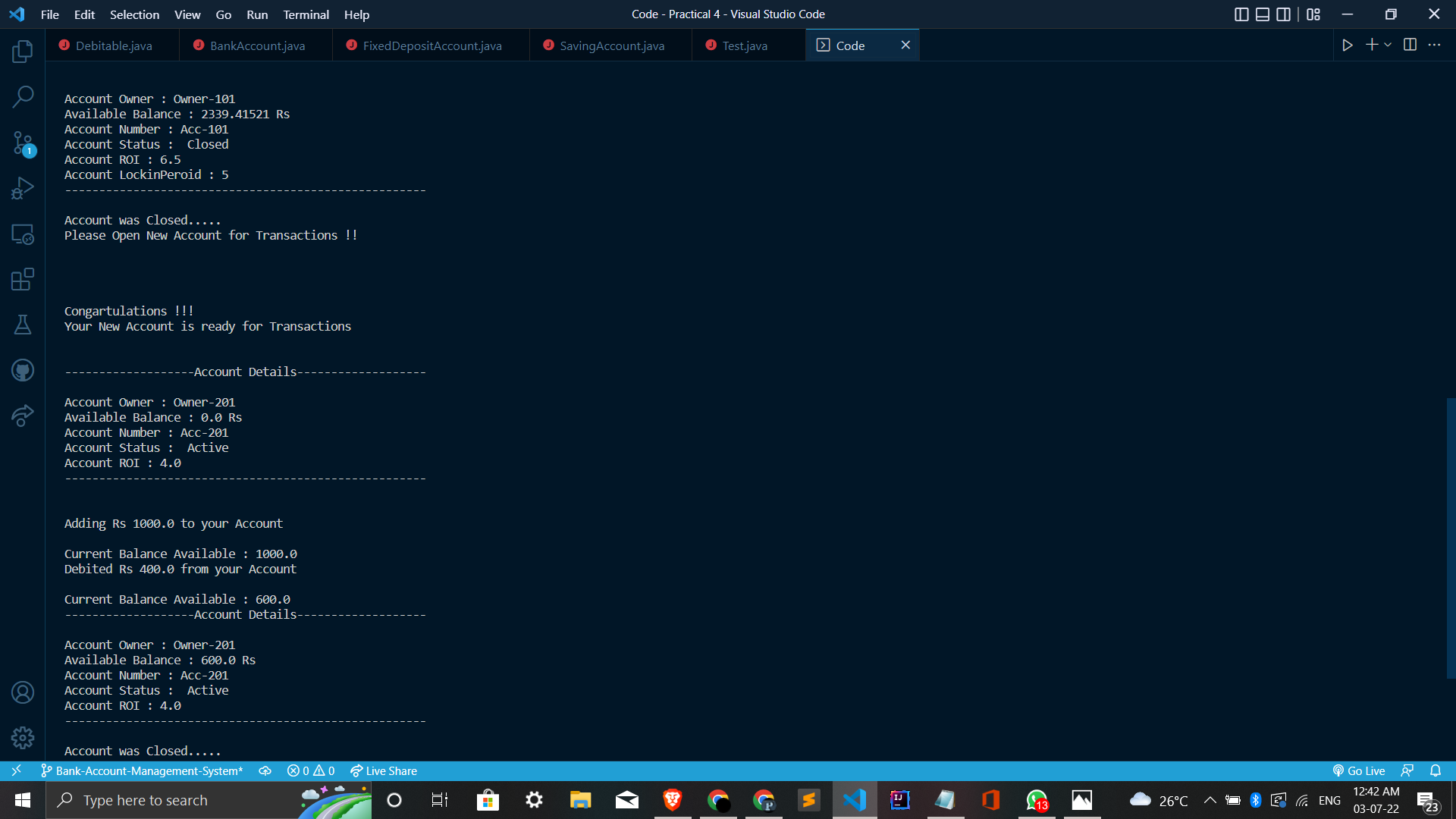
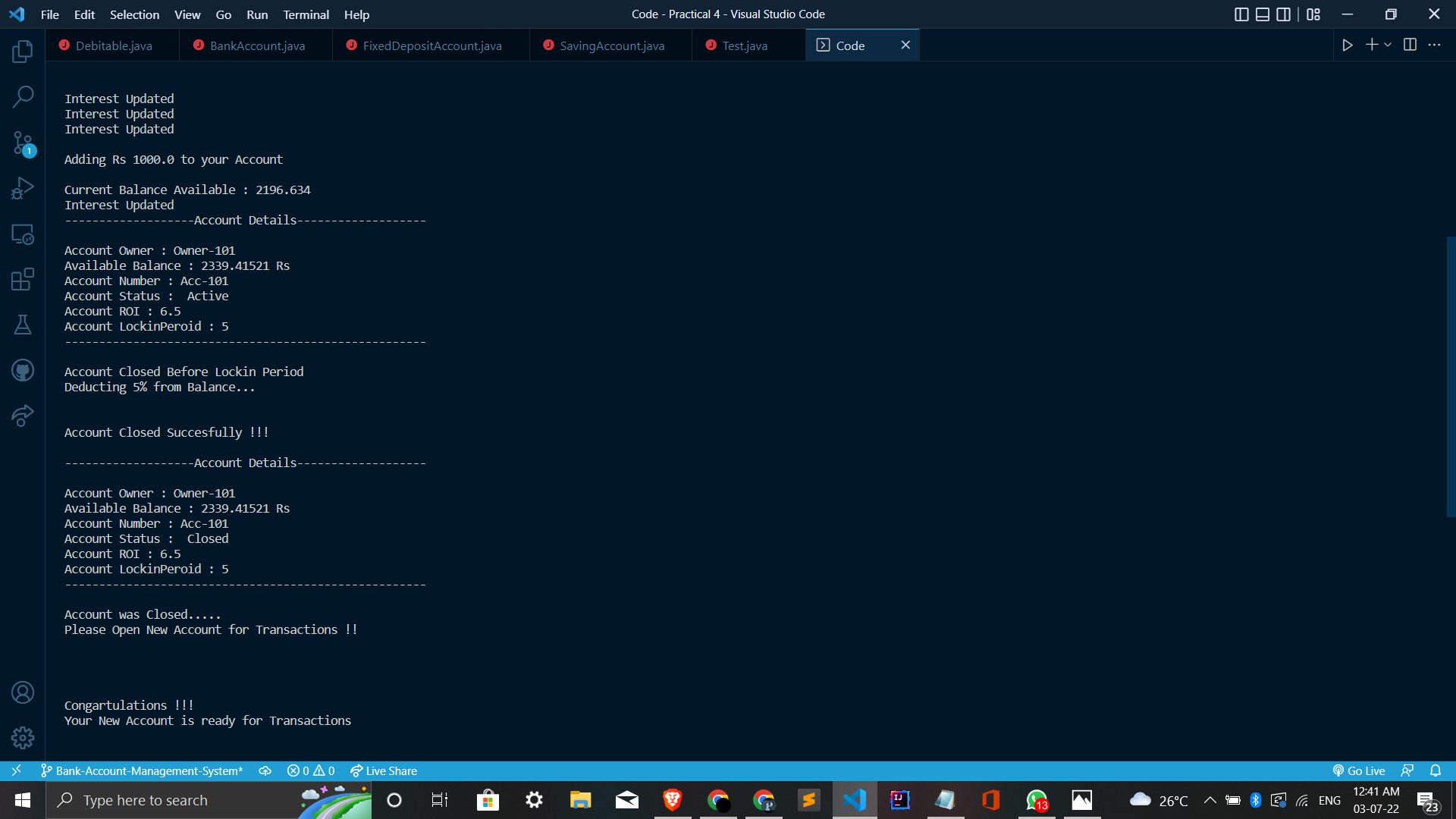
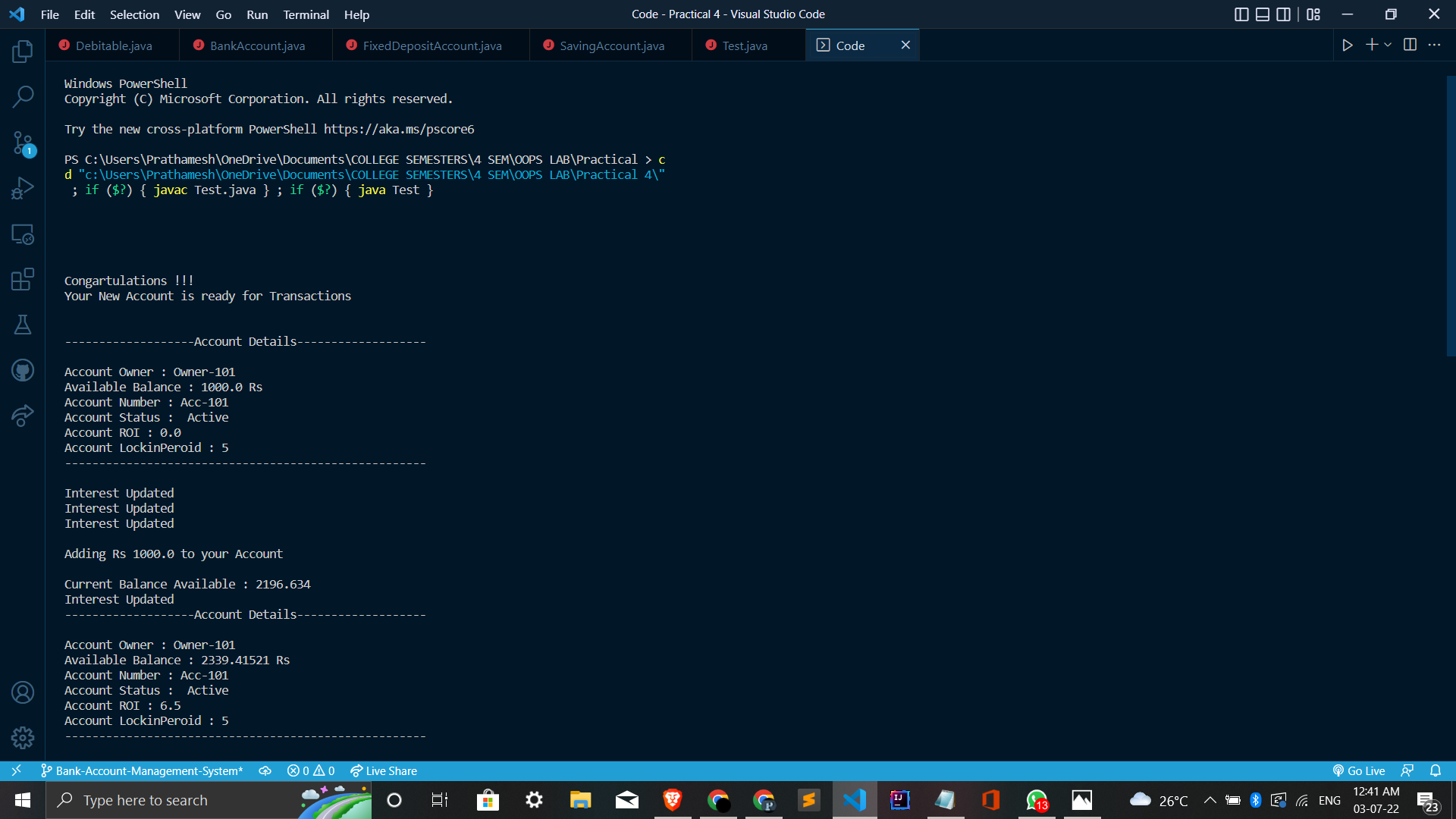
sa1.withdraw(200);

sa1.display\_account();

}

}

**Output :**

****