

## LAB PROGRAM 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- Accept deposit from customer and update the balance.
- Display the balance.
- Compute and deposit interest
- Permit withdrawal and update the balance
- Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
```

```
class Account{  
    String cName,accNum,accType;  
    public static final String ANSI_RED = "\u001B[31m";  
    public static final String ANSI_GREEN = "\u001B[32m";  
    public static final String ANSI_RESET = "\u001B[0m";  
    Scanner sc = new Scanner(System.in);  
  
    Account(String name,String accNo,String accType){  
        this.cName = name;  
        this.accNum = accNo;  
        this.accType = accType;  
    }  
    Account(){};
```

```
}
```

```
class CurrentAcc extends Account{
```

```
    double balance = 5000,rate = 0.06;
```

```
    int time = 5;
```

```
    private boolean canWithdraw = false;
```

```
    CurrentAcc(String name,String accNo,String accType){
```

```
        super(name,accNo,accType);
```

```
        System.out.println("New Customer: " + cName);
```

```
    }
```

```
    void getBalance() {
```

```
        System.out.format("Your balance: %f\n",balance);
```

```
    }
```

```
    void deposit(double amount){
```

```
        char choice;
```

```
        System.out.println("Deposit. Account holder: " + cName + " Amount: " +  
amount);
```

```
        System.out.println("Approve Deposit?(Y/S): ");
```

```
        choice = sc.next().charAt(0);
```

```
        if(choice == 'Y' || choice == 'y'){
```

```
            balance+=amount;
```

```
            System.out.println(ANSI_GREEN + "Deposit approved. Updated balance:  
" + balance + ANSI_RESET);
```

```
        }else{
```

```
        System.out.println(ANSI_RED + "Deposit not approved" + ANSI_RESET);
    }
}
```

```
void withdraw(double amount){
    System.out.println(ANSI_RED + "This account cannot withdraw any funds"
+ ANSI_RESET);
}
```

```
void calcInterest() {};
void checkMinAmount(){
    if(balance < 3000){
        balance-=500;

        System.out.println(ANSI_RED + "Balance under minimum amount to be
maintained." + ANSI_RESET);

        System.out.println(ANSI_RED + "Penalty imposed. Updated balance: " +
balance + ANSI_RESET);
    }
}
}
```

```
class SavingsAcc extends Account{
    double balance = 5000,rate = 0.06;
    private boolean canWithdraw = true;

    SavingsAcc(String name,String accNo,String accType){
        super(name,accNo,accType);
    }
}
```

```
        System.out.println("New Customer: " + cName);  
    }
```

```
void getBalance() {  
    System.out.format("Your balance: %f\n",balance);  
}
```

```
void deposit(double amount){  
    char choice;  
  
    System.out.println("Deposit. Account holder: " + cName + " Amount: " +  
amount);  
  
    System.out.println("Approve Deposit?(Y/S): ");  
  
    choice = sc.next().charAt(0);  
  
    if(choice == 'Y' || choice == 'y'){  
        balance+=amount;  
  
        System.out.println(ANSI_GREEN + "Deposit approved. Updated balance:  
" + balance + ANSI_RESET);  
    }else{  
        System.out.println(ANSI_RED + "Deposit not approved" + ANSI_RESET);  
    }  
}
```

```
void calcInterest(){  
    double CI;  
  
    CI = (balance * (Math.pow((1+((rate/12)/100)),12))) - balance;  
  
    balance+=CI;
```

```
        System.out.println(ANSI_GREEN + "Interest added. Updated balace: " +  
balance + ANSI_RESET);  
    }
```

```
void withdraw(double amount){  
    char choice;  
  
    if(this.canWithdraw){  
        if(balance < amount){  
            System.out.println("Account balance is lower than amount to be  
withdrawn");  
            return;  
        }  
        System.out.println("Approve " + cName + "'s request for withdrawal?  
(Y/N): ");  
        choice = sc.next().charAt(0);  
        if(choice == 'Y' || choice == 'y'){  
            balance-=amount;  
            System.out.println(ANSI_GREEN + "Withdrawal approved. Updated  
balance: " + balance + ANSI_RESET);  
        }else{  
            System.out.println(ANSI_RED + "Withdrawal not approved" +  
ANSI_RESET);  
        }  
    }else{  
        System.out.println(ANSI_RED + "Your Account type doesn't allow  
withdrawals" + ANSI_RESET);  
    }  
}
```

```

    }

    void checkMinAmount(){
        if(balance < 3000){
            balance-=500;

            System.out.println(ANSI_RED + "Balance under minimum amount to be
maintained." + ANSI_RESET);

            System.out.println(ANSI_RED + "Penalty imposed. Updated balance: " +
balance + ANSI_RESET);
        }
    }
}

```

```

public class Lab11 {

    public static void main(String[] args) {

        int c;

        double temp;

        String name,accNo,accType;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Name: ");
        name = sc.nextLine();

        System.out.println("Enter Account number: ");
    }
}

```

```
accNo = sc.nextLine();
System.out.println("Enter Account Type: ");
accType = sc.nextLine();

if(accType.charAt(0) == 'c'){
    CurrentAcc a = new CurrentAcc(name, accNo, accType);
    while(true){
        System.out.println("1. Deposit money\n2. Withdraw money\n3.
Display money\n4. Exit");
        c = sc.nextInt();
        switch(c){
            case 1: {
                System.out.println("Enter amount to be deposited: ");
                temp = sc.nextDouble();
                a.deposit(temp);
                a.checkMinAmount();
                break;
            }
            case 2: {
                System.out.println("Enter amount to be withdrawn: ");
                temp = sc.nextDouble();
                a.withdraw(temp);
                a.checkMinAmount();
                break;
            }
            case 3: {
                a.getBalance();
            }
        }
    }
}
```

```

        break;
    }
    case 4: {
        System.exit(0);
        break;
    }
    default: System.out.println("Enter the correct options");
}

}

}else if(accType.charAt(0) == 's'){
    SavingsAcc a = new SavingsAcc(name, accNo, accType);
    while(true){
        System.out.println("1. Deposit money\n2. Withdraw money\n3.
Display money\n4. Exit");
        c = sc.nextInt();
        switch(c){
            case 1: {
                System.out.println("Enter amount to be deposited: ");
                temp = sc.nextDouble();
                a.deposit(temp);
                a.calcInterest();
                a.checkMinAmount();
                break;
            }
            case 2: {

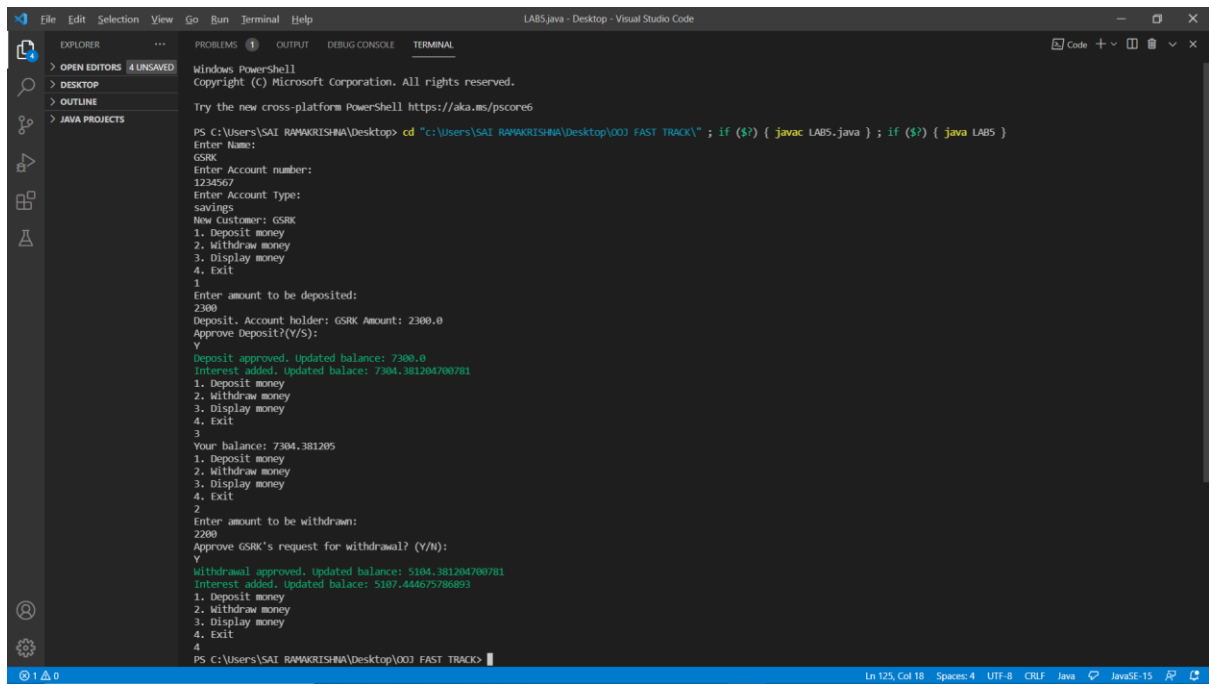
```



```
        System.out.println("Enter amount to be withdrawn: ");
        temp = sc.nextDouble();
        a.withdraw(temp);
        a.calcInterest();
        a.checkMinAmount();
        break;
    }
    case 3: {
        a.getBalance();
        break;
    }
    case 4: {
        System.exit(0);
        break;
    }
    default: System.out.println("Enter the correct options");
}

}

}else{
    System.out.println("Enter valid type... Exiting");
}
}
}
```



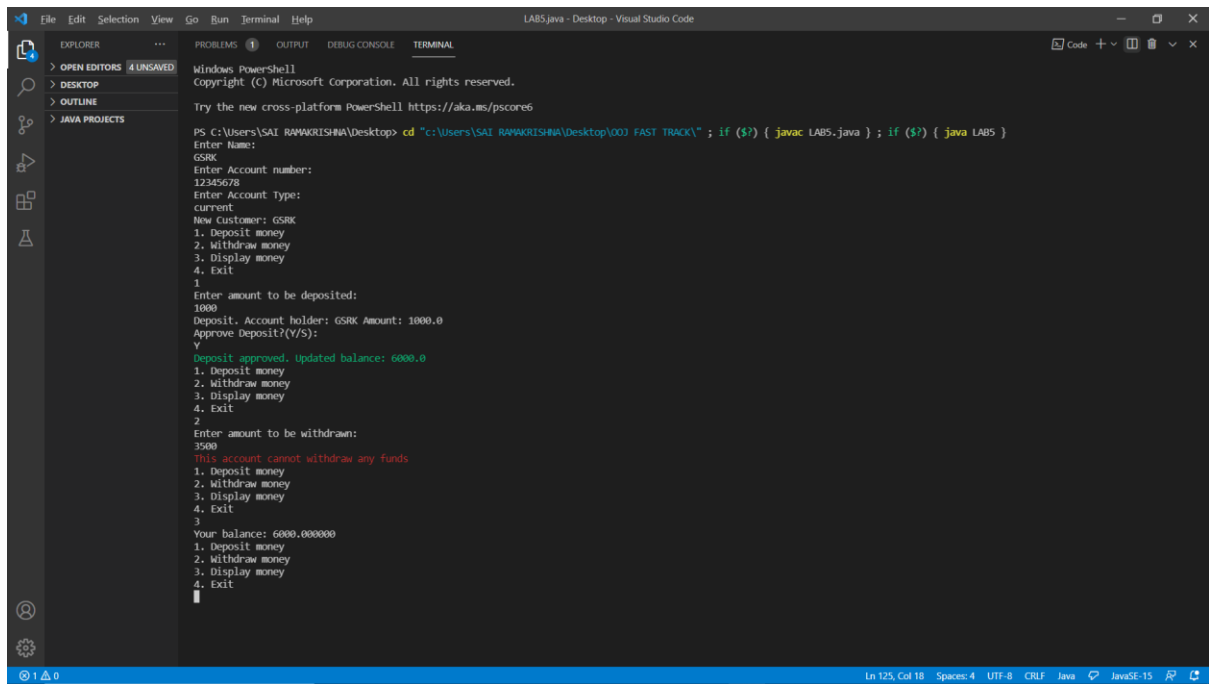
The screenshot shows a Visual Studio Code window with the title 'LAB5.java - Desktop - Visual Studio Code'. The interface includes a sidebar on the left with icons for Explorer, Search, Run and Debug, and Extensions. The main area is divided into three panes: Explorer, Output, and Terminal. The Explorer pane shows a file tree with 'LAB5.java' selected. The Output pane is empty. The Terminal pane shows a Windows PowerShell session. The user has run the command `cd "c:\Users\SAI RAMAKRISHNA\Desktop\001 FAST TRACK\" ; if ($?) { javac LAB5.java } ; if ($?) { java LAB5 }`. The program output is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\SAI RAMAKRISHNA\Desktop\001 FAST TRACK> cd "c:\Users\SAI RAMAKRISHNA\Desktop\001 FAST TRACK\" ; if ($?) { javac LAB5.java } ; if ($?) { java LAB5 }
Enter Name:
GSRK
Enter Account number:
1234567
Enter Account Type:
savings
New Customer: GSRK
1. Deposit money
2. Withdraw money
3. Display money
4. Exit
1
Enter amount to be deposited:
2300
Deposit. Account holder: GSRK Amount: 2300.0
Approve Deposit?(Y/S):
Y
Deposit approved. Updated balance: 7300.0
Interest added. Updated balance: 7304.381204700781
1. Deposit money
2. Withdraw money
3. Display money
4. Exit
3
Your balance: 7304.381205
1. Deposit money
2. Withdraw money
3. Display money
4. Exit
2
Enter amount to be withdrawn:
2200
Approve GSRK's request for withdrawal? (Y/N):
Y
Withdrawal approved. Updated balance: 5104.381204700781
Interest added. Updated balance: 5107.444675706093
1. Deposit money
2. Withdraw money
3. Display money
4. Exit
4
PS C:\Users\SAI RAMAKRISHNA\Desktop\001 FAST TRACK>
```

The status bar at the bottom indicates the current line and column: 'Ln 125, Col 18'. It also shows the file encoding as 'UTF-8' and the line ending as 'CRLF'. The active language is 'Java', and the Java version is 'JavaSE-15'.



The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab active. The terminal window displays the execution of a Java program named 'LAB5.java'. The program prompts the user for various inputs: Name (GSRK), Account number (12345678), and Account type (current). It then displays a menu with options: 1. Deposit money, 2. Withdraw money, 3. Display money, and 4. Exit. The user selects option 1 and enters a deposit amount of 1000. The program confirms the deposit and shows the updated balance as 6000.0. The user then selects option 2 and enters a withdrawal amount of 3500. The program displays an error message: 'this account cannot withdraw any funds'. Finally, the user selects option 3, and the program displays the current balance: 'Your balance: 6000.000000'. The terminal window also shows the command prompt and the execution of the 'javac' and 'java' commands.

```
LAB5.java - Desktop - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER 4 UNSAVED
> OPEN EDITORS
> DESKTOP
> OUTLINE
> JAVA PROJECTS

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\SAI RAMAKRISHNA\Desktop> cd "c:\Users\SAI RAMAKRISHNA\Desktop\001 FAST TRACK\" ; if ($?) { javac LAB5.java } ; if ($?) { java LAB5 }

Enter Name:
GSRK
Enter Account number:
12345678
Enter Account Type:
current
New Customer: GSRK
1. Deposit money
2. Withdraw money
3. Display money
4. Exit
1
Enter amount to be deposited:
1000
Deposit. Account holder: GSRK Amount: 1000.0
Approve Deposit?(Y/N):
Y
Deposit approved. Updated balance: 6000.0
1. Deposit money
2. Withdraw money
3. Display money
4. Exit
2
Enter amount to be withdrawn:
3500
this account cannot withdraw any funds
1. Deposit money
2. Withdraw money
3. Display money
4. Exit
3
Your balance: 6000.000000
1. Deposit money
2. Withdraw money
3. Display money
4. Exit

```

Ln 125, Col 18 Spaces: 4 UTF-8 CRLF Java JavaSE-15