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 Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if

necessary and update the balance.

CODE:

```
import java.util.Scanner;


class Account {
    protected String customerName;
    protected int accountNumber;
    protected String accountType;
    protected double balance;


    public Account(String customerName, int accountNumber, String
accountType, double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }


    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit Successful. New Balance: " + balance);
    }
```

```java
    public void displayBalance() {
        System.out.println("Current Balance: " + balance);
    }


    public void withdraw(double amount) {
        balance -= amount;
        System.out.println("Withdrawal Successful. New Balance: " +
balance);
    }
}


class CurAcct extends Account {
    private static final double MIN_BALANCE = 1000;
    private static final double PENALTY = 50;


    public CurAcct(String customerName, int accountNumber, double balance)
{
        super(customerName, accountNumber, "Current", balance);
    }

    @Override
    public void withdraw(double amount) {
        if (balance - amount < MIN_BALANCE) {
            System.out.println("Insufficient balance. Minimum balance of "
+ MIN_BALANCE + " must be maintained.");
        } else {
            super.withdraw(amount);
            if (balance < MIN_BALANCE) {
                balance -= PENALTY;
                System.out.println("Minimum balance breached. Penalty of "
+ PENALTY + " imposed. New Balance: " + balance);
            }
        }
    }


    public void issueChequeBook() {
        System.out.println("Cheque Book Issued.");
    }
}

class SavAcct extends Account {
    private static final double INTEREST_RATE = 0.05; // 5% compound
interest
    private static final int COMPOUND_FREQUENCY = 1; // yearly compound
interest

    // Constructor for Savings Account
    public SavAcct(String customerName, int accountNumber, double balance)
{
        super(customerName, accountNumber, "Savings", balance);
    }


    @Override
    public void withdraw(double amount) {
        if (amount <= balance) {
            super.withdraw(amount);
        } else {
            System.out.println("Insufficient funds for withdrawal.");
        }
    }

    public void computeAndDepositInterest() {
        double interest = balance * Math.pow((1 + INTEREST_RATE),
COMPOUND_FREQUENCY) - balance;
        balance += interest;
        System.out.println("Interest of " + interest + " computed and
added to your balance. New Balance: " + balance);
    }
}

public class amb {
    private static Scanner scanner = new Scanner(System.in);

    public static Account createAccount() {
        System.out.print("Enter customer name: ");
        String customerName = scanner.nextLine();
        System.out.print("Enter account number: ");
        int accountNumber = scanner.nextInt();
        System.out.print("Enter initial balance: ");
        double balance = scanner.nextDouble();
        scanner.nextLine(); // consume newline character

        System.out.println("Select Account Type (1 for Current, 2 for
Savings): ");
        int choice = scanner.nextInt();
        scanner.nextLine(); // consume newline character

        if (choice == 1) {
            return new CurAcct(customerName, accountNumber, balance);
        } else if (choice == 2) {
            return new SavAcct(customerName, accountNumber, balance);
        } else {
            System.out.println("Invalid choice. Defaulting to Savings
Account.");
            return new SavAcct(customerName, accountNumber, balance);
        }
    }

    public static void main(String[] args) {
        System.out.println("Welcome to the Bank!");

        Account account = createAccount();

        boolean exit = false;
        while (!exit) {
            System.out.println("\nChoose an operation:");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display Balance");
            System.out.println("4. Compute and Deposit Interest (only for
Savings Account)");
            System.out.println("5. Issue Cheque Book (only for Current
Account)");
            System.out.println("6. Exit");
            int option = scanner.nextInt();
            scanner.nextLine(); // consume newline character

            switch (option) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    account.deposit(depositAmount);
                    break;

                case 2:
                    System.out.print("Enter amount to withdraw: ");
                    double withdrawAmount = scanner.nextDouble();
                    account.withdraw(withdrawAmount);
                    break;

                case 3:
                    account.displayBalance();
                    break;

                case 4:
                    if (account instanceof SavAcct) {
                        ((SavAcct) account).computeAndDepositInterest();
                    } else {
                        System.out.println("Interest can only be
calculated for Savings Account.");
                    }
                    break;

                case 5:
                    if (account instanceof CurAcct) {
                        ((CurAcct) account).issueChequeBook();
                    } else {
                        System.out.println("Cheque Book can only be issued
for Current Account.");
                    }
                    break;

                case 6:
                    exit = true;
                    System.out.println("Thank you for using our bank 
services.");
                    break;

                default:
                    System.out.println("Invalid choice. Please try 
again.");
            }
        }

        scanner.close();
    }
}
```

Output:

```


Welcome to the Bank!
Enter customer name: sinchana
Enter account number: 19839803
Enter initial balance: 700
Select Account Type (1 for Current, 2 for Savings):
1

Choose an operation:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest (only for Savings Account)
5. Issue Cheque Book (only for Current Account)
6. Exit
1
Enter amount to deposit: 800
Deposit Successful. New Balance: 1500.0
```

```

Choose an operation:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest (only for Savings Account)
5. Issue Cheque Book (only for Current Account)
6. Exit
2
Enter amount to withdraw: 450
Withdrawal Successful. New Balance: 1050.0

Choose an operation:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest (only for Savings Account)
5. Issue Cheque Book (only for Current Account)
6. Exit
5
Cheque Book Issued.

Choose an operation:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest (only for Savings Account)
5. Issue Cheque Book (only for Current Account)
6. Exit
6
Thank you for using our bank services.
```

NOTES:

PAGE NO :
DATE :

5. Develop a java program to create a class Bank that maintains two kinds of account for its customers, one called saving 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 acount. From derive the classes cur-acct and sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks,
a) Accept deposite from customer and update the balance
b) Display the balance
c) compute and deposit interest
d) permit withdrawal and update the balance.
check for the minimum balance, impose penality if necessary and update the balance.

PAGE NO :
DATE :

```
→ Import java.util.Scanner;

class Account {
  ~~protected~~ String customername;
  ~~protected~~ int accountnumber;
        String accounttype;
        double balance;
 Account (String name, int accnum, String
type, double initialbalance) {

 Customername = name;
  accountname = accname;
  accounttype = type;
   balance = initialbalance

void deposit (double amount) {
   balance += amount
   System.out.println ("Amount deposited" +
amount);
  displaybalance ();
}

 void displaybalane () {
  System.out.println ("balane " + balance
}
}

class savacct extend Account {
```

PAGE NO :
DATE :

```
Savacct (String name, int accnum, double
Initial balance ) {

    Super (name, accnum, "Saving s", initialbala
-nce );

void computeanddepositeinterest (int years)
 {
    double interest = balance =* Math.pow(1+
interestrate) . years ) = balance;
    balance + = interest ;
    System.out.println ("Interest added" +
interest );
displaybalance ( );
}

void withdraw (double amount) {
   if (balance >= amount ) {
        balance - = amount;
   System.out.println ("Amount withdraw"
+ amount );
}
 else {
   System.out.println ("Insufficient
 balance for withdrawal ");
 }

displaybalance ( );
}
```

PAGE NO :
DATE :

```
class curAcct extends Account {
 double minimumbalane = 500.0;

 double servicecharge = 50.0;

 CurAcct (String name, int accNum, double
 initialbalane) {
super (name, accNum, "current", initial
balane);
}

 void withdraw (double amount) {
  if (balance >= amount) {
   balance -= amount:
  System.out.println("Amount withdrawn"+amount
  if (balance <minimumbalance) {
        balance -= service charge;
            system.out.println (servicecharge);
    }
   else {
   System.out.println("Insufficient balance
  for withdraw");
   }
    displaybalance();
  }
  }
  public class bank {
       public static void main (String[] args)
{
  Scanner sc = new
```

PAGE NO :
DATE :

```
double intbalance = sc.nextDouble();
Savacct savaccount = new savacct(name,
acc, intibalance);
      savacct.deposit(1000);
      savacct.withdraw(500);
       sc.nextLine();
System.out.println("Enter details of current
account");
String n2 = sc.nextLine();
int acc2 = sc.nextInt();
double intibalance2 = sc.nextDouble();

curracct(n2, acc2, intibalance2);

   curaccount.deposit(2000);
   curaccount.withdraw(1800);
       sc.close();
}
}.
```

output

```
Enter details saving account
customer name: Alice
   acc no: 12345
  initial deposit: 5000
amount deposit: 1000.0
balance =6000.0
Interest added: 240.0
   balance: 6240.0
    Amount withdrawn: 500.0
```

PAGE NO :

DATE :

```
Enter details for current account
customer name! Bob
acc =6780
initial deposit :2000
Amount deposit :2000.0
balance : 4000.0
Amount withdrawn:1800.0
balance below minimum:service charge
imposed :50.0
balance : 2150.0
```

[illegible] 19/11/2

6 create a package CIE which has a classes stud
& internals the class student has members US
usn, name, sem. the class Internals has an
array that stores the internal marks Stores
which has the class External which is a
derived class of student that declares the
final marks of n student in all 5 cours,