5) Develop a Java Pgm 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 & 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.

```
-> import java.util.Scanner;
   class Account {
       String customerName;
       long accountNumber;
       double balance;
       Public void acceptDeposit(double amount) {
           balance += amount;
       }
```

```
public void display() {
    System.out.println("Balance: "+balance);
}
public void getDetails() {
    System.out.println("Customer Name "+customer
        Name + " In Account Number: "+account
                                    Number);
}
}
class CurrentAccount extends Account {
    private static final double MIN_BALANCE
                = 2000;
    private static final double PENALTY=500;

    public void acceptDeposit(double amount) {
        super.acceptDeposit(amount);
        if (balance < MIN_BALANCE) {
            System.out.println("Penalty
        imposed for not maintaining minimum
                balance.");
        balance -= PENALTY;
    }
}
}
class SavingsAccount extends Account {
    private static final double Intrest_
                RATE = 0.05;
    Public void computeAndDepositInterest(int years)
    {
        double compoundIntrest = balance * Math.
            Pow(1+INTREST_RATE, years) -
                balance;
```

```
        balance += compoundInterest;
    }
    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal Successful.
                Amount withdrawn: " + amount);
        }
        else {
            System.out.println("Insufficient balance.
                Withdrawal unsuccessful.");
        }
    }
}

public class lab5 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        CurrentAccount currentAccount = new CurrentAccount();
        SavingsAccount savingsAccount = new SavingsAccount();
        System.out.println("Enter customer name: ");
        currentAccount.customerName = savingsAccount.customerName = scanner.nextLine();
        System.out.println("Enter account number: ");
        currentAccount.accountNumber = savingsAccount.accountNumber = scanner.nextLong();
        int choice;
        do {
```

```
System.out.println("\n Menu");
System.out.println(" 1. Deposit to current Account");
System.out.println(" 2. Deposit to Savings Account");
System.out.println(" 3. Display Balance for current Account");
System.out.println(" 4. Display Balance for Savings Account");
System.out.println(" 5. Compute Interest for Savings Account");
System.out.println(" 6. Withdraw from Savings Account");
System.out.println(" 7. Exit");
System.out.println(" Enter your choice:");
choice = Scanner.nextInt();

switch (choice) {
    case 1:
        S.o.p("Enter amount to deposit to current Account : ");
        double currentDeposit = Scanner.next
        currentAccount.acceptDeposit(current Double(); deposit);
        break;

    case 2:
        System.out.println("Enter amount to deposit to Savings Account : ");
        double SavingsDeposit = Scanner.nextDouble;
        SavingsAccount.acceptDeposit(SavingsDepos
                                     it);
        break;
```

```
case 3:
    S.o.p ("Current Account Balance : ");
    currentAccount.DisplayBalance();
    break;

case 4:
    S.o.p ("Savings Account Balance : ");
    SavingsAccount.displayBalance();
    break;

case 5:
    System.out.print("Enter the years for
                    Interest calculation : ");
    int years = Scanner.nextInt();
    SavingsAcount.computeAndDepositInterest(
                                    years);
    S.o.p ("Interest Computed & Deposited");
    break;

case 6:
    S.o.p ("Enter amount to withdraw from
                    SavingsAccount : ");
    double withdrawAmount = Scanner.next
                                    Double();
    SavingsAccount.withdraw (withdrawAmount);
    break;

case 7:
    S.o.p ("Exiting program ...");
    break
default:
    S.o.p ("Invalid choice. Try again");
}
} while (choice != 7);
Scanner.close();
}
```

o/p: Enter customer name : vday
Enter account number : 5 6 4 5 7 4 5 6

Menu:
1 - Deposit to current Account
2 - Deposit to Savings Account
3 Display Balance for current Account
4 Display Balance for Savings Account
5. Compute Interest for Savings Account
6. Withdraw from Savings Account
7 - Exit

Enter your choice : 1
Enter amount to deposit to current Accnt : 5000

Menu:
Enter your choice : 2.
Enter amount to deposit to Savings Accnt : 3000

Menu:
Enter your choice : 3.
current Accont Balance :
Balance : 5000.0

Menu:
Enter your choice : 4
Savings Account Balance :
Balance : 3000.0

Menu:
Enter your choice : 5
Enter the number of years for interest calculation : 5
Interest computed and deposited

Menu:
choice : 4

Savings Account Balance :

Balance : 3828.84468.

Menu :

choice : 6.

Enter Amount to withdraw : 3000

Successfully withdrawn : ! 300

Menu :

choice : 4.

Balance : 3528.84468