Q) 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:

5. Develope 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.

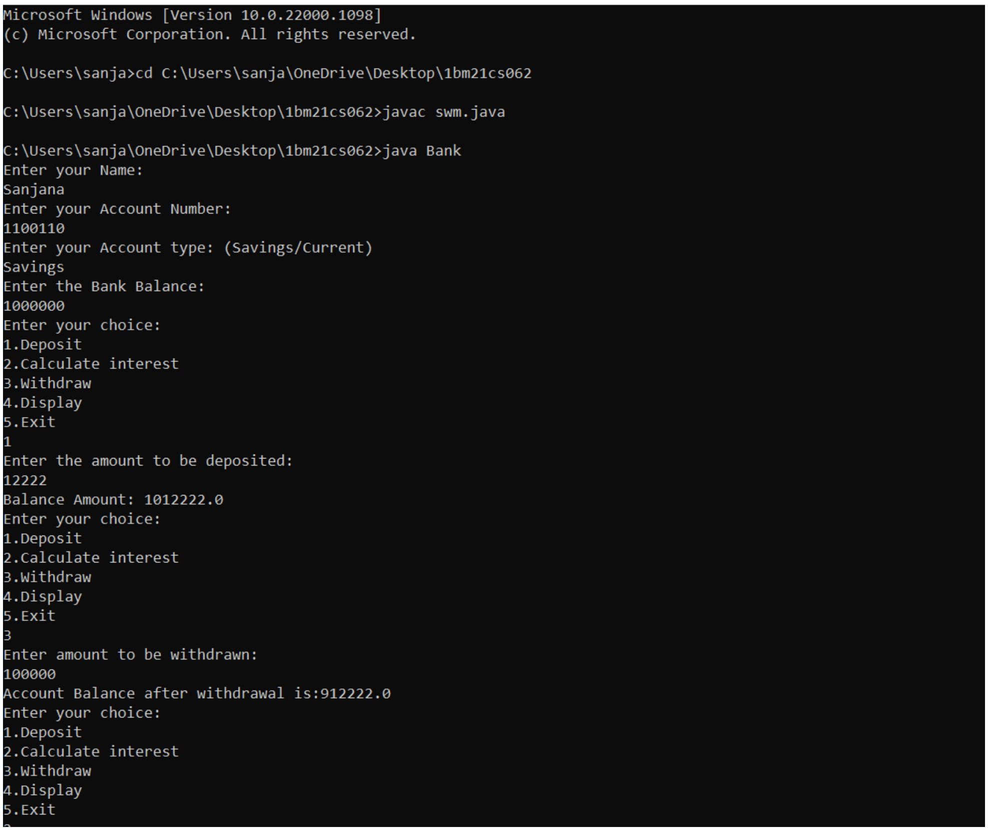
```
import java.util.Scanner.
import java.lang.Math;
class Account
{
    String name, acc_type;
    int acc_no;
     double bal, dep;
    Scanner ss = new Scanner(System.in)
void set()
{
 System.out.println("Enter your name");
    name = ss.next()
  System.out.println(" Enter your account number");
    acc_no = ss.nextInt();
 System.out.println(" Enter your account type=
    (savings, current)");
    acc_type = ss.next();
  ~~S.P.~~ S.O.P("Enter the Bank balance");
       bal = ss.nextDouble();

  }
   void disp()
   {
    S.O.P("Name:"+name);
     S.O.P("Account no:"+acc_no);
     S.O.P("Account type:"+acc_type);
```

```
S.o.p ("Current Balance is" + bal);
}
void deposit()
  {
   S.O.P ("Enter the amount to be deposited");
    dep = ss.nextDouble();
     bal+=dep;
   S.O.P ("Balance amount :"+bal);
  }
 boolean acc( String acc_type)
     {
        if(acc_type = ="Savings")
        return true
      else if (acc_type = ="current"){
        return false;
     else
        return true;
     }
 }
class cur_acct extends Account
     {
       int penal()
        {
           double min ,pen;
        S.O.P ("Enter min amount
   & penalty amount if not followed");
min = SS.nextDouble();
   pen = SS.nextDouble();

  if (bal < min)
   {
     bal -= pen;
   S.O.P ("Penalty imposed for having unsufficient
            balance"); }

      else
       return 1;
   }
  void withdrawal()
    {
      double amt;
    S.O.P ("Enter amt to be withdrawn");
   amt = SS.nextDouble();
      int a = penal();
   if (a == 1)
    {
      if (bal >= amt)
       bal -= amt;
       System.o.p ("Account balance after
        withdrawn is :" + bal; }
     }
   else
   S.O.P("The amount can't be withdrawn");
  }
 }
```

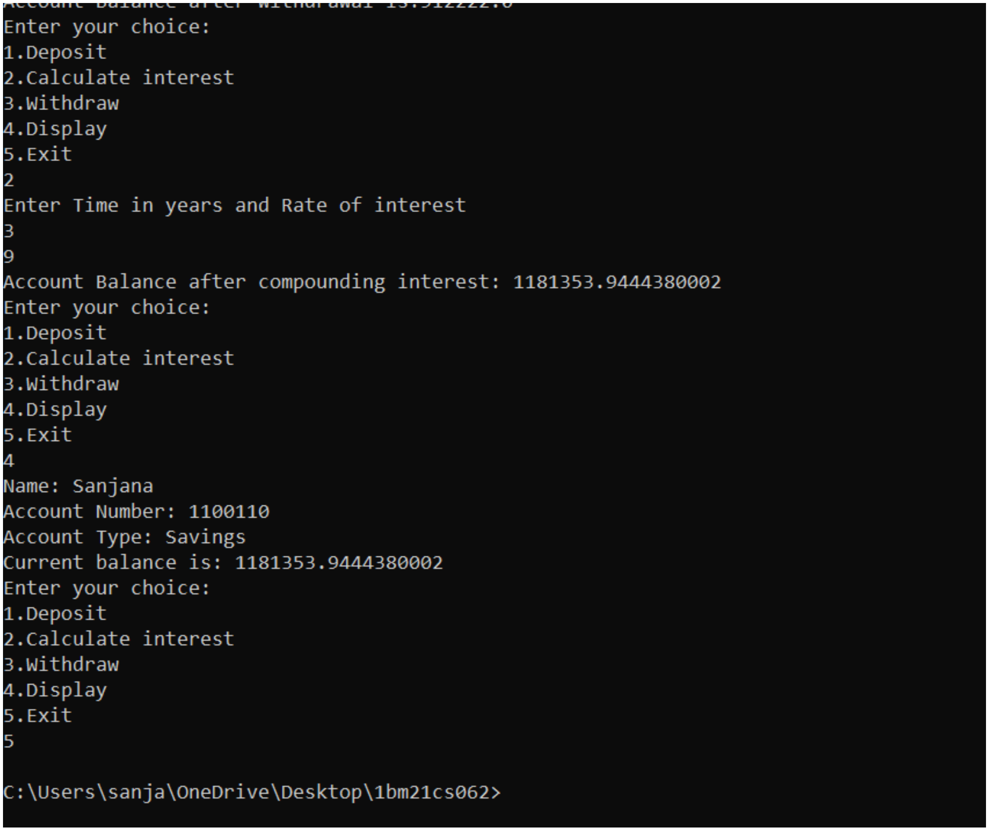
```
class sav_acct extends Account
 {
   void calc_interest ()
     {
       S.O.P ("Enter time & rate of interest");
   double t = ss.nextDouble();
   double r = ss.nextDouble();
   double CI = bal * Math.pow((1+r/100), t);
  S.O.P ("Compound interest is" +CF );
   bal += CI;
  S.O.P ("Balance amount:" +bal);
 }
void  withdrawal()
      {
        double amt;
       S.O.P ("Enter amount to be withdrawn");
        amt = ss.nextDouble();
          if (bal >=amt)
            {
              bal-= amt;
   S.O.P ("Acount balance after withdrawal"
              +bal); }

      }.
```

```
else
  S.o.p ("the amount can't be withdrawn ");
  }
}
class Bank
 {
   p s v m ( String args[]) ;
  {
   Scanner ss = new Scanner (System.in );
    Account a1 = new Account ();
    a1. add ();
  if (a1.acc (a1.acc_type) = = true)
 {
 sav_ acf  s1 = new sav_ acc ();
  s1. name = a1. name;
   s 1.acno = a1. ac_no ;
  s1 .ac_type = a1 ac_type ;
   s 1.bal = a1. bal;
  S. o. p ("Enter your choice : /n 1. Deposit  , 2. Calculate
       3. withdrawn  4. Display 5. Exit  ");
      int ch = ss. nextInt ();
      switch (ch)
  {
     case 1: s1.deposit(); break;
   case 2: s 1 . cal_ interest (). break;
   case 3 : S.1. withdrawal (); break;
   case 4 : s1.disp () break;
   case 5 : exit (0); break;
```

OUTPUT:



```
Microsoft Windows [Version 10.0.22000.1098]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sanja>cd C:\Users\sanja\OneDrive\Desktop\1bm21cs062

C:\Users\sanja\OneDrive\Desktop\1bm21cs062>javac swm.java

C:\Users\sanja\OneDrive\Desktop\1bm21cs062>java Bank
Enter your Name:
Sanjana
Enter your Account Number:
1100110
Enter your Account type: (Savings/Current)
Savings
Enter the Bank Balance:
1000000
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
1
Enter the amount to be deposited:
12222
Balance Amount: 1012222.0
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
3
Enter amount to be withdrawn:
100000
Account Balance after withdrawal is:912222.0
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
```



```
Account Balance after withdrawal is:912222.0
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
2
Enter Time in years and Rate of interest
3
9
Account Balance after compounding interest: 1181353.9444380002
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
4
Name: Sanjana
Account Number: 1100110
Account Type: Savings
Current balance is: 1181353.9444380002
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
5

C:\Users\sanja\OneDrive\Desktop\1bm21cs062>
```