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 CI 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 min balance and if the balance falls below this 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 min balance, impose penalty if necessary and update the balance

```
- import java.util.Scanner;
  class Account {
        String customerName;
        long accno;
        String accountType;
        double balance;
        public Account (String CustomerName,
long accno, String accountType)
        { this.accno = accno;
             this.accountType = accountType;
             this.balance = 0.0;
        }
     public void displayBalance()
    {    System.out.println(" Account Number :"+accno);
         System.out.println ("Customer Name:"+ customer
                                                  Name);
         System.out.println (" Account Type :"
                                + accountType );

         System.out.println (" Balance : $"+ balance);
    }
 }
class Cur_Acct extends Account {
     double minBalance;
     double serviceCharge;
     public CurAcct (String customerName, long accno)
     {    super (customerName, accno, "Current");
          this.minBalance = 500;
          this.serviceCharge = 50;
     }
```

```
public void withdraw (double amount)
{     if (balance- amount > = minBalance)
        { balance- = amount;
      System.out.println ("Withdrawal Successful.
Current Balance: $ " + balance); }
      else
      { System.out.println ("Insufficient funds.
Withdrawal not allowed.");
      }
}
public void impose Service Charge ()
{       if (balance < min Balance)
        {  balance- = Service Charge;
              System.out.println (" Service charge
imposed. Current Balance: RS " + balance);
        }
}
}
class  Sav_Acct extends Account
{   double interestRate;
     public Sav Acct (String customerName, long accno
 { Super (customerName, accno, "Savings")
   this. interestRate = 0.05;
  }
public void ~~deposit~~ compound Interest (double initialAmount, int term)
 { // doube interest = balance * interest Rate;
    // balance + = interest;
double compoundInterest = initialAmount * Math.pow ((1+
     interest Rate), term) - initial Amount;
   balance + = compoundInterest;
```

```
System.out.println ("compound Interest deposited.
  Current Balance : Rs." + balance);
 }
}

public class Bank {
  public static void main (String [] args)
{   Scanner scanner = new Scanner (System.in);
    System.out.println ("Choose account type:");
  System.out.println ("1. Current");
  System.out.println ("2. Savings");
  System.out.print ("Enter choice (1 or 2):");
  int choice = scanner.nextInt ();
   System.out.print ("Enter customer name:");
   String customerName = scanner.next();
   System.out.print ("Enter accno:");
    long accno = Scanner.nextLong ();
      if (choice == 1)
  {     CurAcct cur_Account = new CurAcct
                          (customerName, accno);
   System.out.print (" Enter initial balance : $");
   double initialBalance = Scanner.nextDouble ();
   curAccount.balance = InitialBalance;
   System.out.print (" Enter withdrawal amount : $")
        double withdrawalAmount = scanner.nextDouble(
   curaccount.withdraw (withdrawal Amount);
cur Account. imposeServiceCharge ();
 curAccount. displayBalance ();
 } else if (choice == 2)
{ Sav_Acct savAccount = new SavAcct(customerName,
                                    accno);
 System.out.print(" Enter initial balance: $ ");
 double initialBalance = Scanner. nextDouble();
 savAccount. balance = initialBalance;
  System.out.print (" Enter withdrawal amount: $ ");
 double withdrawalAmount = Scanner. nextDouble();
 SavAccount. balance -= withdrawal Amount;
 System.out.println(" withdrawal successful.
 Current Balance: $" + savAccount. balance);
System. out. print (" Enter interest rate: ");
 double InterestRate = Scanner. nextDouble();
 savAccount. interestRate = interestRate;
 SavAccount. displayBalance ();
 System. out. print (" Enter term (in years) for
 Compound interest calculation: ");
 int term = Scanner. nextInt();
   SavAccount. compoundInterest (initialBalance, terms)
 SavAccount. displayBalance ();
 }
 else {
     System.out.println (" Invalid choice ");
     }
  }
}
```

Algorithm

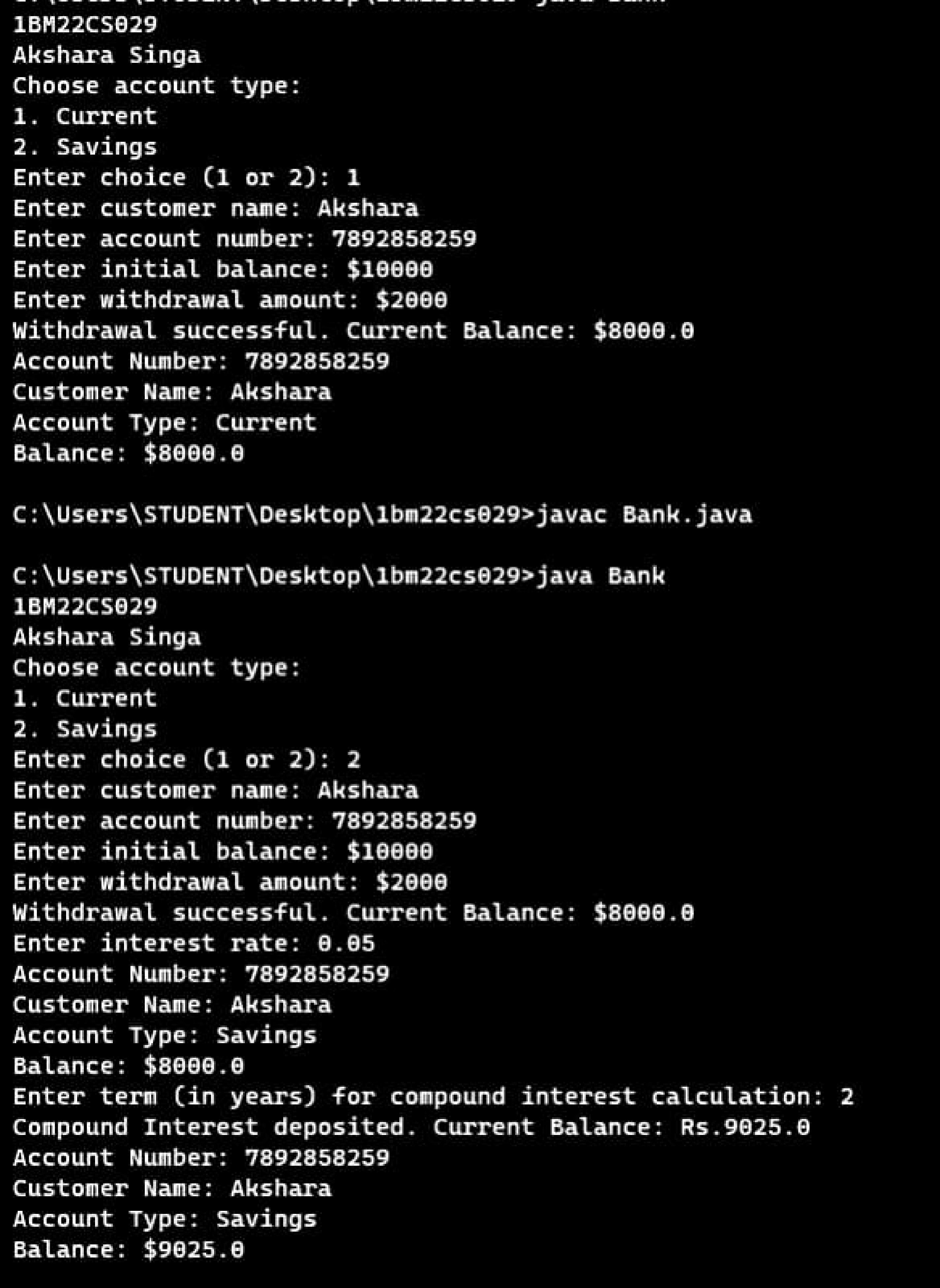
```
Step 1: Start
Step 2: Create class account with attributes
Step 3: Assign values to attribute
Step 4: Create displayBalance method
Step 5: Create current account class extending
        the Account class
Step 6: Create a withdraw method
Step 7: Create a method for imposing penalty
Step 8: Create Savings account class extending
        the Account class
Step 9: Create a method to calculate the
        Compound interest where
        $total = initial * \left(1 + \frac{interest\ rate}{nt}\right)^{nt}$
Step 10: Create a main function which accepts
        ~~the~~ details of the customer and type
        of account
Step 11: Ask for user the type of account and
        call for the required class
Step 12: Stop
```

Proceed 19/11

Output

```
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 1
Enter customer name: Akshara
Enter account number: 7892858259
Enter initial balance: $ 10000
Enter withdrawal amount: $ 2000
Withdrawal succesful. Current Balance: $ 8000.0
Account number: 7892858259
Customer Name: Akshara
Account Type: Current
Balance: $ 8000.0
```

[illegible] 19/11



```
1BM22CS029
Akshara Singa
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 1
Enter customer name: Akshara
Enter account number: 7892858259
Enter initial balance: $10000
Enter withdrawal amount: $2000
Withdrawal successful. Current Balance: $8000.0
Account Number: 7892858259
Customer Name: Akshara
Account Type: Current
Balance: $8000.0

C:\Users\STUDENT\Desktop\1bm22cs029>javac Bank.java

C:\Users\STUDENT\Desktop\1bm22cs029>java Bank
1BM22CS029
Akshara Singa
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 2
Enter customer name: Akshara
Enter account number: 7892858259
Enter initial balance: $10000
Enter withdrawal amount: $2000
Withdrawal successful. Current Balance: $8000.0
Enter interest rate: 0.05
Account Number: 7892858259
Customer Name: Akshara
Account Type: Savings
Balance: $8000.0
Enter term (in years) for compound interest calculation: 2
Compound Interest deposited. Current Balance: Rs.9025.0
Account Number: 7892858259
Customer Name: Akshara
Account Type: Savings
Balance: $9025.0
```