Output :

```
Enter customer name: Shreyash
Enter account number: 2
Enter the type of Account : savings
---MENU---
1. Deposit
2. Withdraw
3. Display Account Details
4. Exit
Enter your choice : 1
Enter deposit amount: 1000
Deposit Successful
[illegible]
Interest computed and deposited. : 10[illegible]50.0
Enter your choice : 2
Enter withdrawal amount: 500
Withdraw Successful. Balance = 550.0
Enter your choice : 3
Customer name: Shreyash
Account number: 2
Type of Account: Savings
Balance: 550.0
```

```
switch (ch) {
   case 1:
      System.out.println("Enter deposit
                account");
   double depoAmt = sc.nextDouble();
   userAccount.deposit(depositAmount);
   break;
 case 2:
    System.out.println("Enter withdrawal
          amount");
   double withdrawalAmt = sc.nextDouble();
  userAccount.withdraw(withdrawAmt);
  break;
 case 3:
   System.out.println("Customer name :"
            userAccount.customername);
   System.out.println("Account number:
            userAccount.accountNumber);
  System.out.println("Type of Account:"+
           userAccount.accountType);
 System.out.println("Balance:"+
          userAccount.balance);
case 4:
   System.out.println("Exiting prog");
   sc.close();
   System.exit(0);
  }
}
 }
}
```

```
public class Main{
        public static void main(String args[]){
            Scanner sc = new Scanner(System.in);
        int ch;
        Account userAccount = null;
        System.out.println("Enter customer
            name: ");
        String customername = sc.next();
        System.out.println("Enter account number:
            ");
        ~~Stoo~~ int accountNumber = sc.nextInt();
    System.out.print("Enter the type of
            Account : ");
        String accounttype = sc.next();
    if(accountType.equals("savings")){
            userAccount = new SavAcct(
                customerName, accountNumber);}
    else if (accountType.equals("current")){
            userAccount = new CurAcct(
                customerName, accountNumber);}
    else {
            System.out.println("Invalid account
                        type");
            sc.close();
            System.exit(0);
    }
 while(true){
        System.out.println("---MENU---\n
            1. Deposit \n 2. Withdraw \n 3.
            Display Account Details \n 4. Exit");
    System.out.print("Enter your choice: ");
    ch = sc.nextInt();
```

```
class SavAcct extends Account {
    double interestRate;
    Savacct(String customer name, int
                accountnumber) {
        super(customerName, accountNumber,
                "savings")
        this.interestRate = 0.05;

    }
    @Override
    void deposit(double amount){
        super.deposit(amount);
        computeInterest();
    }
    void computeInterest(){
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest computed
          and deposited. Balance: " + balance);
    }
    void withdraw(double amount) {
        if(amount <= balance) {
            balance -= amount;
            System.out.println("Withdraw
        Successful. Balance= " + balance);
    }
    else {
        System.out.println("Insufficient
            funds for withdrawal");
    }
  }
}
```

```
void displayBalance() {
        System out println("Account Balance: " +
                    balance);
    }
void withdraw(double amount) {}
}
class CurrAcct extends Account {
    double minbal;
    double service Charge;
    CurrAcct( String customername, int accountNumber)
    {
        super(customer name, account number,
                    "Current");
        this minBalance = 1000;
        this servicecharge = 50;
    }
    @Override
    void deposit(double amount){
        super.deposit(amount);
        checkMinimumBalance();
    }
   @Override
    void displayBalance(){
        super.displayBalance();
        checkMinimumBalance();
    }
    private void checkMinimumBalance {
      if(balance < minBalance) {
                balance -= serviceCharge;
            System.out.println("Service
                charge Imposed: Balance: "+
                            balance);
      }
    }
}
```

→ Develop a JAVA prog to create a class Bank that maintains 2 kind of Account for its customers, savings and current account. saving acc provides compound interest and withdraw facilities. The current account provides check book facility. current acc holder should maintain minimum balance otherwise service charge imposed. class Account contains customer name, account number, type of Account and classes derived are curr-acct and sav-acct.

```
import java.util.Scanner;

class Account {
        String customerName;
        int accountNumber;
        String accountType;
        double balance;
}     Account(String customerName, int account
      Number, String accountType) {
      this. customerName = customerName;
      this accountNumber= customerNumber;
      this. accountType = accountType;
      this balance = 0;
      }
      void deposit(double amount) {
      balance += amount;
      System.out.println("Deposit successful");
     }
```