

45. Savings account provides compound interest & withdrawal facility but no checkbook facility. Current account provides checkbook facility but no interest facility.

```
import java.util.*;

class Account
{
    String customer_name;
    long auno;
    String aatype;
    double balance;

    public Account(String customer_name, long auno, String aatype)
    {
        this.customer_name = customer_name;
        this.auno = auno;
        this.aatype = aatype;
        this.balance = 0;
    }

    public void display()
    {
        System.out.println("Account no: " + auno);
        System.out.println("Customer name: " + customer_name);
        System.out.println("Account type: " + aatype);
        System.out.println("Balance: " + balance);
    }
}
```

```
class Curacc extends Account
{
    double minbal;
    double servicecharge;

    public Curacc(String customer_name, long auno)
    {
        super(customer_name, auno, "Current");
        this.minbal = 1000;
        this.servicecharge = 50;
    }
}
```

```
public void withdraw (double amount)
```

```
{ if (balance - amt >= minbal)
```

```
{ balance -= amt;
```

```
System.out.println("Withdrawal successful. Current bal: " + balance);
```

```
}
```

```
else
```

```
{ System.out.println("Insufficient balance");
```

```
}
```

```
}
```

```
public void service charge()
```

```
{ if (balance < minbal)
```

```
{ balance -= service charge;
```

```
System.out.println("Service charge so imposed");
```

```
}
```

```
}
```

```
}
```

```
class Service extends Account
```

```
{ double interest;
```

```
public Service (String custom_name, long acno)
```

```
{ super (custom_name, acno, "Savings");
```

```
this.interest = 0.075;
```

```
}
```

```
public void deposit (int)
```

```
{ double interest = balance * interest;
```

```
balance += interest;
```

```
System.out.println("Interest deposited after year");
```

```
}
```

```

public void compound(double initial_amt, int, term)
{
    double compound = initial_amt * Math.pow((1 + interest), term) - initial_amt;
    balance += compound;
    System.out.println("Compound int deposited");
}
}

```

```

public class Bank
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter acc type");
        System.out.println("1. Current, 2. Savings");
        int choice = sc.nextInt();
        System.out.println("Enter name");
        current_name = scanner.nextLine();
        System.out.println("Enter acc no");
        accno = sc.nextLong();
        if (choice == 1)
        {
            Current curAccount = new Current(current_name, accno);
            System.out.println("Enter initial bal");
            double initial_bal = sc.nextDouble();
            curAccount.balance = initial_bal;
            System.out.println("Enter withdrawal amount");
            double withdrawal_amt = sc.nextDouble();
            curAccount.withdrawal_amt = sc.nextDouble();
            curAccount.withdrawal_amt ( curAccount.withdrawal_amt );
            curAccount.saving();
            curAccount.display();
        }
    }
}

```

```

else if (choice == 2)

```

```

{
    Saving sAccount = new Saving(current_name, accno);
    System.out.println("Enter bal");
    double initial_bal = sc.nextDouble();
    sAccount.balance = initial_bal;
}

```

```
System.out.println("Get withdrawal amt");  
double withdrawal_amt = sc.nextDouble();  
Savacc.bal -= withdrawal_amt;  
System.out.println("Withdrawal successful");  
System.out.println("Get int rate");
```

```
float int_rate = sc.nextFloat();
```

```
compute()
```

```
System.out.println("Get the term");
```

```
int term = sc.nextInt();
```

```
Savacc.  
computeInt(initial_bal, term);
```

```
then
```

```
Savacc.display();
```

```
}
```

```
}
```

```
}
```

Algorithm

STEP 1: START

STEP 2: initialize `intownname`, `balance`, `anno`, `account type` in the class `Account`

STEP 3: create parameterized constructor and initialize it.

STEP 4: create a method to display the details of the `intown`.

STEP 5: create a class ~~Save~~^{Car} which is ~~into~~ a child class of the class `Account` and initialize `minim bal`, `service charges`.

STEP 6: create a ~~method~~ constructor to initialize `minim bal` & `service charge`

STEP 7: create a method `withdraw` with the parameter `amount`

```
if (balance - amount >= minim bal)
```

```
{ balance -= amount;
```

```
}
```

```
else
```

```
{ Insufficient balance -
```

```
}
```

STEP 8: create a method `servicecharge`

```
if (balance < minim bal)
```

```
{
```

```
balance -= servicecharge
```

```
}
```

STEP 9: create a class called `Savacc` which is a child class of `Account`
Initialize interest rate to 0.0%.

STEP 10: create a method `compound` with the ~~rate~~ `rate` and `Savacc`

STEP 11: create a class called bank and a main method and take input of customer name, accno, initial balance in savings and current acc.

STEP 12: call all the methods and the display method.

STEP 13: END;

Output:

Enter customer name
abhinav

Enter acc type

1. Savings

2. Current

2

Enter customer acc no
000001

Enter initial bal

15000

withdrawal amt?

yes

Enter amt to withdraw

120

withdrawal successful

Enter interest rate

1.5

Enter term

5

Compound interest has been credited

19/10/24

```
C:\java lab>javac Bank.java
```

```
C:\java lab>java Bank
```

```
Enter the number of customers
```

```
1
```

```
FOR CUSTOMER 1
```

```
Enter name
```

```
cus1
```

```
For savings:
```

```
Enter account number, balance, rate of interest, time
```

```
11
```

```
100000
```

```
9
```

```
5
```

```
Deposit success. Balance is: 200000.0
```

```
Compound interest
```

```
New Balance: 307724.7909800001
```

```
For current:
```

```
Enter account number, balance, presence of check (Enter 1 if check present)
```

```
12
```

```
121000
```

```
1
```

```
Deposit success. Balance is: 242000.0
```

```
For savings account:
```

```
Enter withdrawal amount
```

```
145
```

```
Remaining Balance is: 307579.7909800001
```

```
For current account:
```

```
Enter withdrawal amount
```

```
235
```

```
Withdrawal success. Remaining Balance: 241765.0
```

```
1BM22CS004 ABHINAV INAMDAR
```