

.. oqueue.c ● C outputrestrictedqueue.c C lab5.c ⚡ Untitled-3 C iputrestrictedqueue.c ●

oobj > Lab11.java > Lab11

```
1 import java.util.Scanner;
2
3 class Account{
4     String cName,accNum,accType;
5     public static final String ANSI_RED = "\u001B[31m";
6     public static final String ANSI_GREEN = "\u001B[32m";
7     public static final String ANSI_RESET = "\u001B[0m";
8     Scanner sc = new Scanner(System.in);
9
10    Account(String name,String accNo,String accType){
11        this.cName = name;
12        this.accNum = accNo;
13        this.accType = accType;
14    }
15    Account(){};
16 }
17
18 class CurrentAcc extends Account{
19     double balance = 5000,rate = 0.06;
20     int time = 5;
21     private boolean canwithdraw = false;
22
23     CurrentAcc(String name,String accNo,String accType){
24         super(name,accNo,accType);
25         System.out.println("New Customer: " + cName);
26     }
}
```

```
25      CurrentAcc > CurrentAcc(String, String, String)          Labs.java      Untitled-4      Lab11.java
26      System.out.println("New Customer: " + cName);
27
28  void getBalance() {
29      System.out.format("Your balance: %f\n",balance);
30  }
31
32  void deposit(double amount){
33      char choice;
34      System.out.println("Deposit. Account holder: " + cName + " Amount: " + amount);
35      System.out.println("Approve Deposit?(Y/S): ");
36      choice = sc.next().charAt(0);
37      if(choice == 'Y' || choice == 'y'){
38          balance+=amount;
39          System.out.println(ANSI_GREEN + "Deposit approved. Updated balance: " + balance + ANSI_RESET);
40      }else{
41          System.out.println(ANSI_RED + "Deposit not approved" + ANSI_RESET);
42      }
43  }
44
45  void withdraw(double amount){
46      System.out.println(ANSI_RED + "This account cannot withdraw any funds" + ANSI_RESET);
47  }
48
49  void calcInterest() {};
50  void checkMinAmount(){
51      if(balance < 3000){
52          balance-=500;
53          System.out.println(ANSI_RED + "Balance under minimum amount to be maintained." + ANSI_RESET);
54          System.out.println(ANSI_RED + "Penalty imposed. Updated balance: " + balance + ANSI_RESET);
55      }
56  }
57 }
58 }
```

```
53     System.out.println(ANSI_RED + "Balance under minimum amount to be maintained." + ANSI_RESET);
54     System.out.println(ANSI_RED + "Penalty imposed. Updated balance: " + balance + ANSI_RESET);
55 }
56 }
57 }
58
59 class SavingsAcc extends Account{
60     double balance = 5000,rate = 0.06;
61     private boolean canWithdraw = true;
62
63     SavingsAcc(String name,String accNo,String accType){
64         super(name,accNo,accType);
65         System.out.println("New Customer: " + cName);
66     }
67
68     void getBalance() {
69         System.out.format("Your balance: %f\n",balance);
70     }
71
72     void deposit(double amount){
73         char choice;
74         System.out.println("Deposit. Account holder: " + cName + " Amount: " + amount);
75         System.out.println("Approve Deposit?(Y/S): ");
76         choice = sc.next().charAt(0);
77         if(choice == 'Y' || choice == 'y'){
78             balance+=amount;
79             System.out.println(ANSI_GREEN + "Deposit approved. Updated balance: " + balance + ANSI_RESET);
80         }else{
81             System.out.println(ANSI_RED + "Deposit not approved" + ANSI_RESET);
82         }
83     }
84
85     void calcInterest(){
86         double CI;
87
88         CI = (balance * (Math.pow((1+((rate/12)/100)),12))) - balance;
89         balance+=CI;
90         System.out.println(ANSI_GREEN + "Interest added. Updated balance: " + balance + ANSI_RESET);
91     }
92 }
```

```
ojo > Lab11.java > CurrentAcc > CurrentAcc(String, String, String)
78     balance+=amount;
79     System.out.println(ANSI_GREEN + "Deposit approved. Updated balance: " + balance + ANSI_RESET);
80 }else{
81     System.out.println(ANSI_RED + "Deposit not approved" + ANSI_RESET);
82 }
83 }
84
85 void calcInterest(){
86     double CI;
87
88     CI = (balance * (Math.pow((1+((rate/12)/100)),12))) - balance;
89     balance+=CI;
90     System.out.println(ANSI_GREEN + "Interest added. Updated balance: " + balance + ANSI_RESET);
91 }
92
93 void withdraw(double amount){
94     char choice;
95
96     if(this.canWithdraw){
97         if(balance < amount){
98             System.out.println("Account balance is lower than amount to be withdrawn");
99             return;
100     }
101     System.out.println("Approve " + cName + "'s request for withdrawal? (Y/N): ");
102     choice = sc.next().charAt(0);
103     if(choice == 'Y' || choice == 'y'){
104         balance-=amount;
105         System.out.println(ANSI_GREEN + "Withdrawal approved. Updated balance: " + balance + ANSI_RESET);
106     }else{
107         System.out.println(ANSI_RED + "Withdrawal not approved" + ANSI_RESET);
108     }
109 }else{
110     System.out.println(ANSI_RED + "Your Account type doesn't allow withdrawals" + ANSI_RESET);
111 }
112 }
113
114 void checkMinAmount(){
115     if(balance < 3000){
116         balance-=500;
117         System.out.println(ANSI_RED + "Balance under minimum amount to be maintained" + ANSI_RESET);
118     }
119 }
```

```
ojo > ● Lab11.java > CurrentAcc > CurrentAcc(String, String, String)
106     }else{
107         System.out.println(ANSI_RED + "Withdrawal not approved" + ANSI_RESET);
108     }
109 }else{
110     System.out.println(ANSI_RED + "Your Account type doesn't allow withdrawals" + ANSI_RESET);
111 }
112 }
113
114 void checkMinAmount(){
115     if(balance < 3000){
116         balance-=500;
117         System.out.println(ANSI_RED + "Balance under minimum amount to be maintained." + ANSI_RESET);
118         System.out.println(ANSI_RED + "Penalty imposed. Updated balance: " + balance + ANSI_RESET);
119     }
120 }
121 }
122
123
124
125 public class Lab11 {
    Run | Debug
126     public static void main(String[] args) {
        int c;
        double temp;
128
129     String name,accNo,accType;
130
131     Scanner sc = new Scanner(System.in);
132
133     System.out.println("Enter Name: ");
134     name = sc.nextLine();
135     System.out.println("Enter Account number: ");
136     accNo = sc.nextLine();
137     System.out.println("Enter Account Type: ");
138     accType = sc.nextLine();
139
140     if(accType.charAt(0) == 'c'){
141         CurrentAcc a = new CurrentAcc(name, accNo, accType);
142         while(true){
143             System.out.println("1. Deposit money\n2. Withdraw money\n3. Display money\n4. Exit");
144         }
145     }
146 }
```

```
... oqueue.c  C outputrestrictedqueue.c  C lab5.c      E Untitled-3  C iputrestrictedqueue.c  C Lab5.java  
oobj > Lab11.java > CurrentAcc > CurrentAcc(String, String, String)  
1  
132     Scanner sc = new Scanner(System.in);  
133  
3     134     System.out.println("Enter Name: ");  
135     name = sc.nextLine();  
136     System.out.println("Enter Account number: ");  
137     accNo = sc.nextLine();  
138     System.out.println("Enter Account Type: ");  
139     accType = sc.nextLine();  
140  
1 141     if(accType.charAt(0) == 'c'){  
142         CurrentAcc a = new CurrentAcc(name, accNo, accType);  
143         while(true){  
144             System.out.println("1. Deposit money\n2. Withdraw money\n3. Display money\n4. Exit");  
145             c = sc.nextInt();  
146             switch(c){  
147                 case 1: {  
148                     System.out.println("Enter amount to be deposited: ");  
149                     temp = sc.nextDouble();  
150                     a.deposit(temp);  
151                     a.checkMinAmount();  
152                     break;  
153                 }  
154                 case 2: {  
155                     System.out.println("Enter amount to be withdrawn: ");  
156                     temp = sc.nextDouble();  
157                     a.withdraw(temp);  
158                     a.checkMinAmount();  
159                     break;  
160                 }  
161                 case 3: {  
162                     a.getBalance();  
163                     break;  
164                 }  
165                 case 4: {  
166                     System.exit(0);  
167                     break;  
168                 }  
169             default: System.out.println("Enter the correct options");  
170         }  
    }
```

```
File View Go Run Terminal Help Lab11.java - Desktop - Visual Studio Code
queue.c ● C outputrestrictedqueue.c C lab5.c Untitled-3 C iputrestrictedqueue.c ● Lab9.java Untitled-4 Lab11.java X ▶ ⌂
queue.java
Employee.java
queue.c
outputrestrictedqueue.c
lab5.c
Untitled-3
outputrestrictedqueue.c
Lab9.java
Untitled-4
Lab11.java obj
queue-123
Lab
Folder
Folder (2)
Folder (3)
Folder (4)
Folder (5)
Count.class
RentAcc.class
Lab11.class
Lab11.java
SavingsAcc.class
os
os i need
pics
e
e
PROJECTS
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
}
case 3: {
    a.getBalance();
    break;
}
case 4: {
    System.exit(0);
    break;
}
default: System.out.println("Enter the correct options");
}

} else if(accType.charAt(0) == 's'){
    SavingsAcc a = new SavingsAcc(name, accNo, accType);
    while(true){
        System.out.println("1. Deposit money\n2. Withdraw money\n3. Display money\n4. Exit");
        c = sc.nextInt();
        switch(c){
            case 1: {
                System.out.println("Enter amount to be deposited: ");
                temp = sc.nextDouble();
                a.deposit(temp);
                a.calcInterest();
                a.checkMinAmount();
                break;
            }
            case 2: {
                System.out.println("Enter amount to be withdrawn: ");
                temp = sc.nextDouble();
                a.withdraw(temp);
                a.calcInterest();
                a.checkMinAmount();
                break;
            }
            case 3: {
                a.getBalance();
            }
        }
    }
}
```

```
● Lab11.java > CurrentAcc > CurrentAcc(String, String, String)
    System.out.println("Enter amount to be withdrawn: ");
    temp = sc.nextDouble();
    a.withdraw(temp);
    a.calcInterest();
    a.checkMinAmount();
    break;
}
case 3: {
    a.getBalance();
    break;
}
case 4: {
    System.exit(0);
    break;
}
default: System.out.println("Enter the correct options");
}

}

} else{
    System.out.println("Enter valid type... Exiting");
}
}
```

[PowerShell] https://aka.ms/powershell

```
PS C:\Users\SAI RAMAKRISHNA\Desktop> cd "c:\Users\SAI RAMAKRISHNA\Desktop\oof\" ; if ($?) {<br/>    & .\obj\bin\obj.exe <br/>}<br/>Enter Name:<br/>sai<br/>Enter Account number:<br/>24<br/>Enter Account Type:<br/>savings<br/>New Customer: sai<br/>1. Deposit money<br/>2. Withdraw money<br/>3. Display money<br/>4. Exit<br/>1<br/>Enter amount to be deposited:<br/>5000<br/>Deposit. Account holder: sai Amount: 5000.0<br/>Approve Deposit?(y/s):<br/>y<br/>Deposit approved. Updated balance: 10000.0<br/>Interest added. Updated balance: 10006.001650275042<br/>1. Deposit money<br/>2. Withdraw money<br/>3. Display money<br/>4. Exit<br/>4<br/>PS C:\Users\SAI RAMAKRISHNA\Desktop\oof>
```