

```

1  #include <iostream>
2  using namespace std;
3
4  class account
5  {
6  private:
7      string name,atype;
8      int accno;
9  public:
10     void get_acc_details()
11     {
12         cout<<"\nEnter Customer Name:";
13         cin>>name;
14         cout<<"\nEnter Account Number :";
15         cin>>accno;
16         cout<<"\nEnter Account type:";
17         cin>>atype;
18     }
19     void display_details()
20     {
21         cout<<"\n\nCustomer Name : "<<name;
22         cout<<"\n\nAccount Number: "<<accno;
23         cout<<"\n\nAccount Type  : "<<atype;
24     }
25 };
26 class current_acc: public account
27 {
28 private:
29     float balance;
30 public:
31     void c_display()
32     {
33         cout<<"\nBalance : "<<balance<<endl;
34     }
35     void c_deposit()
36     {
37         float deposit;
38         cout<<"\nEnter amount to Deposit:";
39         cin>>deposit;
40         balance=balance+deposit;
41     }
42     void c_withdraw()
43     {
44         float withdraw;
45         cout<<"\n\nBalance : "<<balance<<endl;
46         cout<<"\nEnter the amount to be withdraw :";
47         cin>>withdraw;
48         if(balance>1000){
49             balance=balance-withdraw;
50
51             cout<<"\nBalance Amount After withdraw : "<<balance;
52         }
53         else{
54             cout<<"\nInsufficient balance"<<endl;
55         }
56     }
57 };
58 class savings_acc: public account
59 {
60 private:
61     float sav_balance;
62 public:
63     void s_display()
64     {
65         cout<<"\nBalance : "<<sav_balance<<endl;
66     }

```

```

67 void s_deposit()
68 {
69     float deposit, interest;
70     cout<<"\nEnter amount to Deposit:";
71     cin>>deposit;
72     sav_balance=sav_balance+deposit;
73     interest = (sav_balance*2)/100;
74     sav_balance=sav_balance+interest;
75 }
76 void s_withdraw()
77 {
78     float withdraw;
79     cout<<"\n\nBalance : "<<sav_balance<<endl;
80     cout<<"\nEnter the amount to be withdraw : ";
81     cin>>withdraw;
82     if(sav_balance>500){
83         sav_balance=sav_balance-withdraw;
84
85         cout<<"\nBalance Amount After withdraw : "<<sav_balance;
86     }
87     else{
88         cout<<"\nInsufficient balance"<<endl;
89     }
90 }
91 };
92 int main()
93 {
94     current_acc c1;
95     savings_acc s1;
96     char type;
97     cout<<"\nEnter s for saving customer and c for current a/c customer : ";
98     cin>>type;
99     int choice;
100     if(type=='s' || type=='S')
101     {
102         s1.get_acc_details();
103         while(1)
104         {
105             cout<<"\nChoose your choice "<<endl;
106             cout<<"1) Deposit "<<endl;
107             cout<<"2) Withdraw "<<endl;
108             cout<<"3) Display Balance "<<endl;
109             cout<<"4) Display with full Details "<<endl;
110             cout<<"5) Exit "<<endl;
111             cout<<"\nEnter your choice : "<<endl;
112             cin>>choice;
113             switch(choice)
114             {
115                 case 1:
116                     s1.s_deposit();
117                     break;
118                 case 2:
119                     s1.s_withdraw();
120                     break;
121                 case 3:
122                     s1.s_display();
123                     break;
124                 case 4:
125                     s1.display_details();
126                     s1.s_display();
127                     break;
128                 case 5:
129                     goto end;
130                 default :
131                     cout<<"\nEntered choice is invalid,\n'Try Again'"<<endl;
132             }

```

```

133     }
134 }
135 else if (type=='c' || type=='C')
136 {
137     cl.get_acc_details();
138     while(1)
139     {
140         cout<<"\nChoose your choice "<<endl;
141         cout<<"1) Deposit "<<endl;
142         cout<<"2) Withdraw "<<endl;
143         cout<<"3) Display Balance "<<endl;
144         cout<<"4) Display with full Details "<<endl;
145         cout<<"5) Exit "<<endl;
146         cout<<"\nEnter your choice : "<<endl;
147         cin>>choice;
148         switch(choice)
149         {
150             case 1:
151                 cl.c_deposit();
152                 break;
153             case 2:
154                 cl.c_withdraw();
155                 break;
156             case 3:
157                 cl.c_display();
158                 break;
159             case 4:
160                 cl.display_details();
161                 cl.c_display();
162                 break;
163             case 5:
164                 goto end;
165             default :
166                 cout<<"\nEntered choice is invalid,\n'Try Again'";
167         }
168     }
169 }
170 else{
171     cout<<"\nInvalid Account Selection"<<endl;
172 }
173 end:
174 cout<<"\nThank you for Banking with us...!";
175 return 0;
176 }

```