

Apr 01, 16 10:07

wy157.txt

Page 1/22

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/Account.
h Page 1

```
1  /*
2  student name: Weicheng Yin
3  student id: 5009819
4  this is the .h file for CSCI204 ass1
5  */
6
7
8  #ifndef _ACCOUNT_
9  #define _ACCOUNT_
10
11  class Account{
12  private:
13      int idNum;
14      int bsbNum;
15      int accNum;
16      std::string firstName;
17      std::string lastName;
18      double balance;
19      double tWithdraw;
20
21
22  public:
23      Account();
24      Account(int, int, int, std::string, std::string, double);
25
26      int getId();
27      int getBsb();
28      int getAccount();
29      std::string getFirst();
30      std::string getLast();
31      double getBal();
32
33      void modifyBal(int);          //modify the account balance
34
35      //void loadAmount(char *, Account *);
36      //void withdraw(Account *, ATM *);
37      //void findBal(Account *);
38  };
39
40  #endif
```

Apr 01, 16 10:07

wy157.txt

Page 2/22

Apr 01, 16 10:07

wy157.txt

Page 3/22

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/Account.
cpp Page 1

```

1  /*
2  student name: Weicheng Yin
3  student id: 5009819
4  this is the .cpp file for CSCI204 ass1
5  */
6
7
8  #include <iostream>
9  #include <fstream>
10 #include <cstring>
11 #include <cstdlib>
12 #include "Account.h"
13
14
15 using namespace std;
16
17 //constructor
18 Account::Account()
19 {
20     idNum = 0;
21     bsbNum = 0;
22     accNum = 0;
23     balance = 0;
24     firstName = "";
25     lastName = "";
26     tWithdraw = 0;
27 }
28
29 //this is the destructor
30
31
32
33 Account::Account(int id, int bsb, int acc, string fname, string lname, d
34 ouble bal)
35 {
36     idNum = id;
37     bsbNum = bsb;
38     accNum = acc;
39     firstName = fname;
40     lastName = lname;
41     balance = bal;

```

Apr 01, 16 10:07

wy157.txt

Page 4/22

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/Account.
cpp Page 2

```

41
42 }
43
44
45 int Account::getBsb()
46 {
47     return bsbNum;
48 }
49
50
51 int Account::getAccount()
52 {
53     return accNum;
54 }
55
56 double Account::getBal()
57 {
58     return balance;
59 }
60
61 void Account::modifyBal(int cash)
62 {
63     balance -= cash;
64 }
65
66 string Account::getFirst()
67 {
68     return firstName;
69 }
70
71
72 string Account::getLast()
73 {
74     return lastName;
75 }
76
77 int Account::getId()
78 {
79     return idNum;
80 }

```

Apr 01, 16 10:07

wy157.txt

Page 5/22

Apr 01, 16 10:07

wy157.txt

Page 6/22

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/ATM.h Pa
ge 1

```

1  /*
2  student name: Weicheng Yin
3  student id: 5009819
4  this is the .h file for CSCI204 ass1
5  */
6
7
8  #ifndef _ATM_
9  #define _ATM_
10
11  class ATM{
12  private:
13      int atmId;
14      int twentyNum;
15      int fiftyNum;
16
17  public:
18      ATM();
19      ATM(int , int, int);
20
21      int getId();
22      void addFifty(int);
23      void addTwenty(int);
24      void loadAtm(char *, ATM *);
25      int getTwenty();
26      int getFifty();
27      void modifyCash(int, int);           //modify the cash box af
28  };
29
30  #endif

```

Apr 01, 16 10:07

wy157.txt

Page 7/22

Apr 01, 16 10:07

wy157.txt

Page 8/22

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/ATM.cpp
Page 1

```
1  /*
2  student name: Weicheng Yin
3  student id: 5009819
4  this is the .cpp file for CSCI204 ass1
5  */
6
7
8  #include <iostream>
9  #include <fstream>
10 #include <cstdlib>
11 #include "ATM.h"
12
13 using namespace std;
14
15 ATM::ATM()
16 {
17     atmId = 0;
18     twentyNum = 0;
19     fiftyNum = 0;
20 }
21
22 ATM::ATM(int aId, int aTwenty, int aFifty)
23 {
24     atmId = aId;
25     twentyNum = aTwenty;
26     fiftyNum = aFifty;
27 }
28
29 int ATM::getId()
30 {
31     return atmId;
32 }
33
34 int ATM::getTwenty()
35 {
36     return twentyNum;
37 }
38
39 int ATM::getFifty()
40 {
```

Apr 01, 16 10:07

wy157.txt

Page 9/22

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/ATM.cpp
Page 2

```
41         return fiftyNum;
42     }
43
44     void ATM::addTwenty(int amou)
45     {
46         twentyNum += amou;
47     }
48
49     void ATM::addFifty(int amou)
50     {
51         fiftyNum += amou;
52     }
53
54
55     void ATM::modifyCash(int twenty, int fifty)
56     {
57         fiftyNum -= 50 * fifty;
58         twentyNum -= 20 * twenty;
59     }
```

Apr 01, 16 10:07

wy157.txt

Page 10/22

Apr 01, 16 10:07

wy157.txt

Page 11/22

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/main.cpp
Page 1

```

1  /*
2  student name: Weicheng Yin
3  student id: 5009819
4  this is the main file for CSCI204 ass1
5  */
6
7
8  #include <iostream>
9  #include <fstream>
10 #include <cstring>
11 #include <cstdlib>
12 #include <ctime>
13 #include "ATM.h"
14 #include "Account.h"
15
16 using namespace std;
17
18 void showMenu()
19 {
20     cout << "1. Withdraw" << endl;
21     cout << "2. Find balance" << endl;
22     cout << "3. Add notes" << endl;
23     cout << "4. Quit" << endl;
24     cout << "Please choose: ";
25 }
26
27 void loadAmount(ifstream& fin, Account *accPtr, int accAmount)
28 {
29     fin.ignore();
30     char temp[32];
31     int id, bsb, acc;
32     string firstN, secondN;
33     double bal;
34     for(int i = 0; i < accAmount; i++)
35     {
36         fin.getline(temp, 32, ',');
37         id = atoi(temp);
38         fin.getline(temp, 32, ',');
39         bsb = atoi(temp);

```

Apr 01, 16 10:07

wy157.txt

Page 12/22

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/main.cpp
Page 2

```

41         fin.getline(temp, 32, ',');
42         acc = atoi(temp);
43         getline(fin, firstN, ',');
44         getline(fin, secondN, ',');
45         fin >> bal;
46         fin.ignore();
47         accPtr[i] = Account(id, bsb, acc, firstN, secondN, bal);
48     }
49 }
50
51 void loadAtm(ifstream& fin, ATM *aPtr, int atmNum)
52 {
53     fin.ignore();
54     char temp[32];
55     int id, twenty, fifty;
56     for(int i = 0; i < atmNum; i++)
57     {
58         fin.getline(temp, 32, ',');
59         id = atoi(temp);
60         fin.getline(temp, 32, ',');
61         twenty = atoi(temp);
62         fin.getline(temp, 32);
63         fifty = atoi(temp);
64         aPtr[i] = ATM(id, twenty, fifty);
65     }
66 }
67
68 void withdraw(Account *accPtr, ATM *aPtr, int accAmount, int atmNum, ofs
69 tream &fout)
70 {
71     int bsb, account, amount;
72     int fNotes = 0;
73     int tNotes = 0;
74     cout << "BSB: ";
75     cin >> bsb;
76     cout << "Account: ";
77     cin >> account;
78     cout << "Amount: ";
79     cin >> amount;
80     bool found = false;

```



Apr 01, 16 10:07

wy157.txt

Page 13/22

```

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/main.cpp
Page 3

81     for(int i = 0; i < accAmount; i++)
82     {
83         if(accPtr[i].getBsb() == bsb)
84         {
85             if(accPtr[i].getAccount() == account)
86             {
87                 found = true;
88                 if(amount > accPtr[i].getBal())
89                 {
90                     cout << "the amount cash is bigger
er than the account balance. Return to main menu." << endl;
91                 }
92                 return;
93             }
94             else if(amount < 20)
95             {
96                 cout << "the amount is invalid.
" << endl;
97             }
98             else
99             {
100                 //compute how many 50 and 20 notes should
be given
101                 int atm = rand() % 5 + 800001;
102                 //generate the random ATM machine
103                 //calculate the cash use the greedy algorithm
104                 int cash = amount;
105                 int fNum = amount / 50;
106                 fNotes = fNum;
107                 cash %= 50;
108                 bool flag = false;
109                 for(int j = 0; j <= fNum && !flag; j++)
110                 {
111                     if((cash % 20) != 0)
112                     {
113                         cash += 50;
114                         tNotes = cash / 20;
115                         fNotes--;
116                     }
117                     else
118                     {
119                         tNotes = cash / 20;
120                         flag = true;
121                     }
122                 }
123             }
124         }
125     }
126 }
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160

```

Apr 01, 16 10:07

wy157.txt

Page 14/22

```

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/main.cpp
Page 4

121     if((tNotes == 0) && (fNotes == 0))
122     {
123         //cant find the proper solution to withdraw the cash
124         cout << "the amount you want to withdraw is invalid." << endl;
125         return;
126     }
127     else{
128         //the cash is valid
129         cout << "You have " << fNotes << " " << "x$50 notes, " << tNotes << "x$20 notes." << endl;
130         //modify the balance of the account
131         accPtr[i].modifyBal(amount - cash);
132         cout << "Your balance is " << accPtr[i].getBal() << endl;
133         //update the ATM cash box
134         for(int j = 0; j < atmNum; j++)
135         {
136             if(atm == aPtr[j].getId())
137             {
138                 aPtr[j].modifyCash(tNotes, fNotes);
139             }
140             //write the log to the ATM.log file
141             //ofstream fout;
142             //fout.open("ATM.log");
143             time_t t = time(0);
144             struct tm * now = localtime(&t);
145             fout << atm << ", " << now->tm_mon + 1 << " " << now->tm_mday << " " << now->tm_hour << " " << now->tm_min << " " << now->tm_sec << ", " << "withdraw" << ", " << amount << "." << endl;
146             cout << "incorrect bank account, return to main menu." << endl;
147             return;
148         }
149     }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }

```

Apr 01, 16 10:07

wy157.txt

Page 15/22

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/main.cpp
Page 5

```

161     }
162
163
164 void findBal(Account *accPtr, int accAmount)
165 {
166     int bsb, account;
167     cout << "BSB: ";
168     cin >> bsb;
169     cout << "Account: ";
170     cin >> account;
171     bool found = false;
172     for(int i = 0; i < accAmount && !found; i++)
173     {
174
175         if(accPtr[i].getBsb() == bsb)
176         {
177             if(accPtr[i].getAccount() == account)
178             {
179                 cout << "Your balance is $" << accPtr[i]
180                 .getBal() << endl;
181                 found = true;
182             }
183         }
184     }
185     if(!found)
186     {
187         cout << "cant find the account details. Return to main m
188         enu" << endl;
189         return;
190     }
191
192 void addNotes(ATM *aPtr, int atmNum, ofstream& fout)
193 {
194     int id, choice, amount, balance ;
195     cout << "ATM ID: ";
196     cin >> id;
197     cout << "Note type(1-$50, 2-$20): ";
198     cin >> choice;
199     cout << "Amount: ";
200     cin >> amount;

```

Apr 01, 16 10:07

wy157.txt

Page 16/22

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/main.cpp
Page 6

```

201     bool found = false;
202
203     for(int i = 0; i < atmNum && !found; i++)
204     {
205         if(aPtr[i].getId() == id)
206         {
207             found = true;
208             if(choice != 1 && choice != 2)
209             {
210                 cout << "invalid type choice, back to ma
211                 in menu" << endl;
212                 return;
213             }
214             else
215             {
216                 if(choice == 1)
217                 {
218                     if(amount % 50 != 0)
219                     {
220                         cout << "the input amoun
221                         t is invalid, back to main menu" << endl;
222                         return;
223                     }
224                     else
225                     {
226                         aPtr[i].addFifty(amount)
227                         ;
228                         //add total cash for $50 notes
229                     }
230                 }
231                 else if(amount == 2)
232                 {
233                     if(choice % 20 != 0)
234                     {
235                         cout << "the input amoun
236                         t is invalid, back to main menu" << endl;
237                         return;
238                     }
239                     else
240                     {
241                         aPtr[i].addTwenty(amount)
242                         ;
243                     }
244                 }
245             }
246
247             time_t t = time(0); // get time now
248             struct tm * now = localtime( & t );
249
250

```


Apr 01, 16 10:07

wy157.txt

Page 17/22

```

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/main.cpp
Page 7

241             //open the log file and record the log
242             //ofstream fout;
243             //fout.open("ATM.log");
244             fout << id << ", " << now->tm_mday << "/" << (no
w->tm_mon + 1) << "/";
245             fout << (now->tm_year + 1900) << " " << now->tm_
hour << ":";
246             fout << now->tm_min << ":" << now->tm_sec << ",
";
247             if(choice == 1)
248                 fout << "Add $50 notes, ";
249             else
250                 fout << "Add $20 notes, ";
251             fout << amount << ", " << balance << "." << endl
;
252             //fout.close();
253         }
254         if(!found)
255         {
256             cout << "can't find the account numbet, back to
main menu." << endl;
257             return;
258         }
259     }
260 }
261 }
262 //save the record to the "ATM.txt" and "Accounts.txt" file
263 void saveRecord(Account *accPtr, ATM *aPtr, int accAmount, int atmNum)
264 {
265     ofstream fout;
266     fout.open("accounts.txt");
267     fout << accAmount << endl;
268     cout << "account.txt" << endl;
269     for(int i = 0; i < accAmount; i++)
270     {
271         fout << accPtr[i].getId() << ", " << accPtr[i].getBsb()
<< ", " << accPtr[i].getAccount() << ", ";
272         fout << accPtr[i].getFirst() << ", " << accPtr[i].getLas
t() << ", " << accPtr[i].getBal() << endl;
273     }
274     fout.close();
275     fout.open("ATM.txt");
276     cout << "ATM.txt" << endl;
277     fout << atmNum << endl;
278     for(int i = 0; i < atmNum; i++)
279     {
280         fout << aPtr[i].getId() << ", " << aPtr[i].getTwenty() <
< ", " << aPtr[i].getFifty() << endl;

```

Apr 01, 16 10:07

wy157.txt

Page 18/22

```

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/main.cpp
Page 8

281     }
282     fout.close();
283 }
284
285 int main(int argc, char** argv)
286 {
287     char accFile[16];
288     char atmFile[16];
289
290     //get two file name from the command line
291     strcpy(accFile, argv[1]);
292     strcpy(atmFile, argv[2]);
293     //open the log file to record the status
294     ofstream fileWrite;
295     fileWrite.open("ATM.log");
296     //read the record in account.txt file
297     int accNum, atmNum;
298     ifstream finAcc, finAtm;
299     finAcc.open(accFile);
300     finAcc >> accNum;
301
302     Account *accountPtr; //create dynamic array f
or account
303     accountPtr = new Account[accNum];
304     loadAmount(finAcc, accountPtr, accNum);
305     finAcc.close();
306     cout << "Accounts have been loaded" << endl;
307
308     finAtm.open(atmFile);
309     finAtm >> atmNum;
310     ATM *atmPtr; //create dynamic
array for atm
311     atmPtr = new ATM[atmNum];
312     loadAtm(finAtm, atmPtr, atmNum);
313     finAtm.close();
314
315     cout << "ATMs have been loaded" << endl;
316
317
318     int choice;
319     do
320     {

```

Apr 01, 16 10:07

wy157.txt

Page 19/22

Mar 27 00:08 2016 /packages/turnin/submit/CSCI204/1.closed/wy157/final/main.cpp
Page 9

```

321             showMenu();
322             cin >> choice;
323             switch(choice){
324                 case 1: withdraw(accountPtr, atmPtr, accNum, atm
Num, fileWrite);
325                     break;
326                 case 2: findBal(accountPtr, accNum);
327                     break;
328                 case 3: addNotes(atmPtr, atmNum, fileWrite);
329                     break;
330                 case 4: saveRecord(accountPtr, atmPtr, accNum, a
tmNum);
331                     break;
332                 default: cout << "invalid choice, Please input c
hoice again." << endl;
333                     showMenu();
334             }
335
336
337             }while(choice != 4);
338
339             fileWrite.close();
340             delete [] accountPtr;
341             delete [] atmPtr;
342
343             return 0;
344
345     }
```

Apr 01, 16 10:07

wy157.txt

Page 20/22

Apr 01, 16 10:07

wy157.txt

Page 21/22

```
*****Now attempt Compilation *****
*****End of Compilation *****
```

```
*****Now attempt to execute the program*****
```

```
Reading a.out
Reading ld.so.1
Reading rtcaihook.so
Reading libstdc++.so.6.0.9
Reading libm.so.2
Reading libgcc_s.so.1
Reading libc.so.1
Reading libdl.so.1
Reading rtcaudit.so
Reading libmapmalloc.so.1
Reading libgen.so.1
Reading libm_hwcapi.so.2
Reading libc_psr.so.1
Reading rtcboot.so
Reading librtc.so
Reading librt.so.1
Reading libaio.so.1
Reading libmd.so.1
Reading libmd_psr.so.1
leaks checking - ON
Running: a.out accounts.txt ATM.txt
(process id 8130)
RTC: Enabling Error Checking...
RTC: Running program...
Accounts have been loaded
ATMs have been loaded
1. Withdraw
2. Find balance
3. Add notes
4. Quit
Please choose: BSB: Account: Amount: You have 2x$50 notes, 3x$20 notes.Your balance is 2140.2
1. Withdraw
2. Find balance
3. Add notes
4. Quit
Please choose: BSB: Account: Your balance is $2140.2
1. Withdraw
2. Find balance
3. Add notes
4. Quit
Please choose: ATM ID: Note type(1-$50, 2-$20): Amount: 1. Withdraw
2. Find balance
3. Add notes
4. Quit
Please choose: account.txt
ATM.txt
Checking for memory leaks...
errors are being redirected to file 'a.out.errs'

Actual leaks report (actual leaks: 0 total size: 0 bytes
)

Possible leaks report (possible leaks: 0 total size: 0 bytes
)

RTC output redirected to logfile 'a.out.errs'
```

Apr 01, 16 10:07

wy157.txt

Page 22/22

```
execution completed, exit code is 0
*****NEnd of the program*****
```

```
*****The log file contains*****
```

```
800004, 1/4/2016 10:7:33, withdraw, 802134, 11213465, 160.
800001, 1/4/2016 10:7:33, Add $50 notes, 20000, 11213465.
*****End of log file*****
```



```
*****
Comments:
*****
```



```
*****
Total mark:
*****
Marked by: Qun Chen
```

