```cpp
#include "bankAccount.h"
#include <string>
#include <fstream>

using namespace std;

// reads in initial database of 3 objects

int read_accts(bankAccount account[], int max_accts)
{
    int num_accts = 3;
    string fName[3] = {"Sally", "Thomas", "George"};
    string lName[3] = {"Anderson", "Lee", "Lopez"};
    int SSNumber[3] = {256252121, 652521212, 121252652};
    int accountNumb[3] = {1005820, 1005821, 1005822};
    string accountTyp[3] = {"Checking", "Savings", "CD"};
    double accountBalance[3] = {10212.50, 9523.25, 26545.75};


    for (int i = 0; i < max_accts; i++)
    {
        if (i == num_accts)
            break;
        account[i].nameAndNumber.firstAndLast.firstName = fName[i];
        account[i].nameAndNumber.firstAndLast.lastName = lName[i];
        account[i].nameAndNumber.SSNumber = SSNumber[i];
        account[i].accountNumber = accountNumb[i];
        account[i].accountType = accountTyp[i];
        account[i].balance = accountBalance[i];

    }
    return num_accts;
}

void menuDisplay()
{
    cout << '\n';
    cout << "MENU" << '\n';
    cout << "W) Withdrawal\n";
    cout << "D) Deposit\n";
    cout << "N) New Account\n";
    cout << "B) Balance\n";
    cout << "I) Account Info\n";
    cout << "C) Close Account\n";
    cout << "Enter choice: ";
}

//void menu()
//{
//    char choice;
//
//    do
//    {
//        menuDisplay();
//        cin >> choice;
//        while (toupper(choice) != 'W' || toupper(choice) != 'D' || toupper(choice) != 'N' \
//                || toupper(choice) != 'B' || toupper(choice) != 'I' || toupper(choice) != 'C')
//        {
//            cout << "Make a choice of either 'w', 'd', 'n', 'b', 'i', 'c': " << '\n';
//            cin >> choice;
//
//        }
//}
```

```
 64
 65  // finds the requested account by going through account array
 66
 67  int findacct(const bankAccount account[], int num_accts, int requested_account)
 68  {
 69  //cout << "num_accts " << num_accts << '\n';
 70      for (int i = 0; i < num_accts; i++)
 71      {
 72      //cout << "findacct: " << account[i].accountNumber << " " << requested_account
<< '\n';
 73          if (account[i].accountNumber == requested_account)
 74          {
 75          //cout << "findacct match: " << account[i].accountNumber << " " <<
requested_account << '\n';
 76              return i;
 77          }
 78
 79      }
 80      return -1;
 81  }
 82
 83  void withdrawal(bankAccount account[], int num_accts)
 84  {
 85  //cout << "num_accts " << num_accts << '\n';
 86
 87      int input = 0;
 88      int withdrawAmount = 0;
 89  //bool confirm = false;
 90      cout << "Enter Account Number: ";
 91      cin >> input;
 92      int findAccount = findacct(account, num_accts, input);
 93      if (findAccount == -1)
 94      {
 95          cout << "Account Number: " << input << " doesn't exist." << '\n';
 96      }
 97      else
 98      {
 99          cout << "How much do you want to withdraw? : ";
100          cin >> withdrawAmount;
101          if (withdrawAmount > account[findAccount].balance)
102              cout << "Insufficient funds." << '\n';
103          else
104          {
105              cout << "Withdrawing: " << withdrawAmount << " dollars" << '\n';
106              account[findAccount].balance = account[findAccount].balance -
withdrawAmount;
107              cout << "New Balance: " << account[findAccount].balance << '\n';
108          }
109      }
110
111  }
112
113  void deposit(bankAccount account[], int num_accts)
114  {
115      int input = 0;
116      int depositAmount = 0;
117  //bool confirm = false;
118      cout << "Enter Account Number: ";
119      cin >> input;
120      int findAccount = findacct(account, num_accts, input);
121      if (findAccount == -1)
122      {
123          cout << "Account Number: " << input << " doesn't exist." << '\n';
124      }
125      else
126      {
```

```
127          cout << "How much do you want to deposit? : ";
128          cin >> depositAmount;
129          if (depositAmount > account[findAccount].balance)
130              cout << "Insufficient funds." << '\n';
131          else
132          {
133              cout << "Depositing: " << depositAmount << " dollars" << '\n';
134              account[findAccount].balance = account[findAccount].balance +
depositAmount;
135              cout << "New Balance: " << account[findAccount].balance << '\n';
136          }
137      }
138  }
139
140  // creates new account, increments num_accts after creating new account and returns
new num_accts
141
142  int new_acct(bankAccount account[], int num_accts)
143  {
144      int input = 0;
145      int accountType = 0;
146      //bool confirm = false;
147      cout << "Enter a new Account Number: ";
148      cin >> input;
149      int findAccount = findacct(account, num_accts, input);
150      if (findAccount == -1)
151      {
152          account[num_accts].accountNumber = input;
153          cout << "Account Number: " << account[num_accts].accountNumber << '\n';
154          cout << "Enter First Name: ";
155      //cout << "num_accts " << num_accts << '\n';
156          cin >> account[num_accts].nameAndNumber.firstAndLast.firstName;
157          cin.ignore();
158      //cout << account[num_accts].nameAndNumber.firstAndLast.firstName << '\n';
159          cout << "Enter Last Name: ";
160          cin >> account[num_accts].nameAndNumber.firstAndLast.lastName;
161          cin.ignore();
162          cout << "Enter Social Security Number: ";
163          cin >> account[num_accts].nameAndNumber.SSNumber;
164          cout << "Enter Account Type: ";
165          cin >> accountType;
166          switch (accountType)
167          {
168              case 1: cout << "Checking" << '\n';
169                        account[num_accts].accountType = "Checking";
170                        break;
171              case 2: cout << "Savings" << '\n';
172                        account[num_accts].accountType = "Savings";
173                        break;
174              case 3: cout << "CD" << '\n';
175                        account[num_accts].accountType = "CD";
176                        break;
177
178          }
179          cout << "Enter Initial Deposit: ";
180          cin >> account[num_accts].balance;
181
182      //cout << account[num_accts].balance << '\n';
183          num_accts += 1;
184      }
185      else
186      {
187          cout << "Account already exists." << '\n';
188
189      }
190      return num_accts;
```

```
191
192 }
193
194 // closes account by emptying out values of object in the array
195
196 int close_acct(bankAccount account[], int num_accts)
197 {
198     int input = 0;
199 //bool confirm = false;
200     cout << "Enter Account Number: " << '\n';
201     cin >> input;
202     int findAccount = findacct(account, num_accts, input);
203     if (findAccount == -1)
204     {
205         cout << "Account Number: " << input << " doesn't exist." << '\n';
206     }
207     else if (account[findAccount].balance > 0)
208     {
209         cout << "There is a non-zero balance in Account Number: " <<
account[findAccount].accountNumber << '\n';
210     }
211     else if (findAccount != -1)
212     {
213         cout << "Closing Account: " << input << '\n';
214         account[findAccount].nameAndNumber.firstAndLast.firstName = "";
215         account[findAccount].nameAndNumber.firstAndLast.lastName = "";
216         account[findAccount].nameAndNumber.SSNumber = 0;
217         account[findAccount].accountType = "";
218         account[findAccount].accountNumber = 0;
219         account[findAccount].balance = 0;
220
221     }
222     return num_accts - 1;
223 }
224
225 void balance(const bankAccount account[], int num_accts)
226 {
227     int input = 0;
228     cout << "Enter Account Number: ";
229     cin >> input;
230     int findAccount = findacct(account, num_accts, input);
231     if (findAccount == -1)
232     {
233         cout << "Account Number: " << input << " doesn't exist." << '\n';
234     }
235     else
236     {
237         cout << "Your balance is: " << account[findAccount].balance << '\n';
238
239     }
240 }
241
242 void account_info(const bankAccount account[], int num_accts)
243 {
244
245     int input = 0;
246     bool found = true;
247     cout << "Enter Social Security Number: ";
248     cin >> input;
249     for (int i = 0; i < num_accts; i++)
250     {
251         if (input == account[i].nameAndNumber.SSNumber)
252         {
253             found = true;
254             cout << "Account Found." << '\n';
255             cout << "Account Info: " << '\n';
```

```
256             cout << "Name: " << account[i].nameAndNumber.firstAndLast.firstName << "
";
257             cout << account[i].nameAndNumber.firstAndLast.lastName << '\n';
258             cout << "SSN: " << account[i].nameAndNumber.SSNumber << '\n';
259             cout << "Account Number: " << account[i].accountNumber << '\n';
260             cout << "Account Type: " << account[i].accountType << '\n';
261             cout << "Account Balance: " << account[i].balance << '\n';
262             break;
263 
264         }
265         else
266         {
267             found = false;
268 
269         }
270     }
271     if (found == false)
272         cout << "Account SSN: " << input << " doesn't exist." << '\n';
273 }
274 
275 void print_accts(const bankAccount account[], int num_accts)
276 {
277     ofstream outfile("outfile.txt");
278 
279     for (int i = 0; i < num_accts; i++)
280     {
281         cout << "Account Info: " << '\n';
282         cout << "Name: " << account[i].nameAndNumber.firstAndLast.firstName << " ";
283         cout << account[i].nameAndNumber.firstAndLast.lastName << '\n';
284         cout << "SSN: " << account[i].nameAndNumber.SSNumber << '\n';
285         cout << "Account Number: " << account[i].accountNumber << '\n';
286         cout << "Account Type: " << account[i].accountType << '\n';
287         cout << "Account Balance: " << account[i].balance << '\n';
288         cout << '\n';
289 
290         outfile << "Account Info: " << '\n';
291         outfile << "Name: " << account[i].nameAndNumber.firstAndLast.firstName << "
";
292         outfile << account[i].nameAndNumber.firstAndLast.lastName << '\n';
293         outfile << "SSN: " << account[i].nameAndNumber.SSNumber << '\n';
294         outfile << "Account Number: " << account[i].accountNumber << '\n';
295         outfile << "Account Type: " << account[i].accountType << '\n';
296         outfile << "Account Balance: " << account[i].balance << '\n';
297         outfile << '\n';
298     }
299 }
```