```cpp
#include <iostream>
#include <fstream>
#include <iomanip>
#include <vector>
#include <map>
#include <iterator>
#include <algorithm>
#include <numeric>
#include <ctime>
#include <locale>

#include "transaction.hpp"
#include "account.hpp"
#include "viewhelper.hpp"

#define DATA_FILE "money.dat"

using namespace std;

/**
 * This function will read in the data file and pull all of the account
 * info and transaction data back into memory.
 * @param[out] accounts - This vector will populated with accounts
 * @param[in] is - The input stream to read data from
 * @return - the number of transactions read in
 */
int readInDataFile(vector<Account>& accounts, istream& is) {
  string leadin;
  int transactionCount = 0;
  while(is) {
    is >> leadin;
    if (leadin.empty()) {
      continue;
    }
    else if (equal(begin(leadin), end(leadin), begin(BEGIN_ACCOUNT))) {
      Account a;
      is >> a;
      accounts.push_back(a);
    }
    else if (equal(begin(leadin), end(leadin), begin(BEGIN_TRANSACTION))) {
      Transaction t;
      is >> t;
      ++transactionCount;
      for_each(begin(accounts), end(accounts), [&t](Account& a) {
            string an = t.getAccountName();
            if (equal(begin(an), end(an), begin(a.getName()))) {
              a.addTransaction(t);
            }
          });

    }
    else {
      cerr << "Unknown header field found: " << leadin << "\n";
    }

    leadin.clear();
  }

  return transactionCount;
}

/**
 * This function will output all accounts and transactions to our file
 * so that they can be loaded at a later time.
 *
 * @param[out] accounts - A vector refernce that holds the accounts to output
 * @param[in] os - The output stream to write the data to
 */
void outputDataToFile(vector<Account>& accounts, ostream& os) {
  for (Account& a : accounts) {
    os << a;
    for (const Transaction& t : a.getTransactions()) {
      os << t;
    }
  }
}

int main() {
  //Set cin and cout to use the system locale

  std::locale loc(""); //default locale should be system locale

  cout.imbue(loc);
  cin.imbue(loc);

  bool quit = false;
  vector<Account> accounts;
  ifstream is(DATA_FILE);
  //is.imbue(locale(is.getloc(), new Delimiter()));

  if (is) {
    int transactions = readInDataFile(accounts, is);
    cout << "Read in " << transactions << " transactions\n";
  }

  is.close();

  printBanner();

  //Main program loop
  while(!quit) {
    string command = prompt();


    if (command.empty()) {
      continue;
    }
    else if (equal(begin(command), end(command), begin("quit"))) {
      quit = true;
      break;
    }
    else if (equal(begin(command), end(command), begin("ledger"))) {
      for(const Account& a : accounts) {
        printAccountLedger(a);
      }
    }
    else if (equal(begin(command), end(command), begin("deposit"))) {
      Transaction t = readTransaction();
      addToAccount(t, accounts);
      cout << "Deposit complete\n\n\n";
    }
    else if (equal(begin(command), end(command), begin("withdraw"))) {
      Transaction t = readTransaction(true);
      addToAccount(t, accounts);
      cout << "Widthdraw complete\n\n\n";
    }
    else if (equal(begin(command), end(command), begin("new"))) {
      Account a = readAccount();
      accounts.push_back(a);
      cout << "Account created\n\n\n";
    }
    else if (equal(begin(command), end(command), begin("delete"))) {
      deleteAccount(accounts);
      cout << "Account deleted\n\n\n";
    }
    else if (equal(begin(command), end(command), begin("help"))) {
      printHelp();
    }
    else {
      cerr << "Unknown command\n";
    }
  }

  ofstream ofs(DATA_FILE);
  outputDataToFile(accounts, ofs);
  ofs.close();

  return 0;
}
```