Reality Check: Week 12

Word Problem (Question #1)

Following is a C++ program, which will create a random-access file and perform reading & writing operations into a file. Write the code in Visual Studio and generate its output.

// "ClientData.h" Header file

```
#define CLIENTDATA_H
#include <string>
using namespace std;

class ClientData
{
public:
     ClientData(int = 0, string = "", string = "", double = 0.0);

     void setAccountNumber(int);
     int getAccountNumber() const;

     void setLastName(string);
     string getLastName() const;

     void setFirstName(string);
     string getFirstName() const;

     void setBalance(double);
     double getBalance() const;
private:
     int accountNumber;
     char lastName[15];
     char firstName[15];
     double balance;

};
```

// “ClientData.cpp” C++ File Implementing Class Definition.

```
#include <iostream>
#include <string>
#include "ClientData.h"
using namespace std;

ClientData::ClientData(int accNumValue, string lnamevalue, string
fnamevalue, double balancevalue)
{
      setAccountNumber(accNumValue);
      setLastName(lnamevalue);
      setFirstName(fnamevalue);
      setBalance(balancevalue);
}

int ClientData::getAccountNumber() const
{
      return accountNumber;
}

void ClientData::setAccountNumber(int accNumValue)
{
      accountNumber = accNumValue;
}

string ClientData::getLastName() const
{
      return lastName;
}

void ClientData::setLastName(string lnameValue)
{
      int length = lnameValue.size();
      length = (length < 15 ? length : 14);
      lnameValue.copy(lastName, length);
      lastName[length] = '\0';
}

string ClientData::getFirstName() const
{
      return firstName;
}

void ClientData::setFirstName(string fnameValue)
{
    int length = fnameValue.size();
    length = (length < 15 ? length : 14);
    fnameValue.copy(firstName, length);
    firstName[length] = '\0';
}

double ClientData::getBalance() const
{
    return balance;
}

void ClientData::setBalance(double balanceValue)
{
    balance = balanceValue;
}
```

// “ClientDataMain.cpp” C++ file with Main () Function

```
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include "ClientData.h"
using namespace std;

void FileWrite();
void FileRead();
void outputline(ostream &, const ClientData &);

int main()
{
    ofstream outCredit("credit.dat", ios::out | ios::binary);

    if (!outCredit)
    {
        cout << "File could not be opened..!!!" << endl;
        exit(1);
    }

    ClientData blankClient;

    for (int i = 0; i < 100; i++)
    {
        outCredit.write(reinterpret_cast <const char*>
(&blankClient), sizeof(ClientData));
    }
    outCredit.close();

    FileWrite();
    FileRead();

    cout << endl;
    return 0;
}

void FileWrite()
{
    int accountNumber;
    string lastName;
    string firstName;
    double balance;

    ofstream outCredit("credit.dat", ios::out | ios::binary);

    if (!outCredit)
    {
        cout << "File could not be opened..!!!" << endl;
        exit(1);
    }

    cout << "Enter Account Number (1 to 100), 0 to End Input)\n?
";

    ClientData Client;
    cin >> accountNumber;

    while (accountNumber > 0 && accountNumber <= 100)
    {
        cout << "Enter LastName, FirstName, Balance: \n? ";
        cin >> lastName;
        cin >> firstName;
        cin >> balance;

        Client.setAccountNumber(accountNumber);
        Client.setLastName(lastName);
        Client.setFirstName(firstName);
        Client.setBalance(balance);
```

```
          outCredit.seekp((Client.getAccountNumber() - 1) *
sizeof(ClientData));

          outCredit.write(reinterpret_cast <const char*>
(&Client), sizeof(ClientData));

          cout << "Enter Account Number (1 to 100), 0 to End
Input)\n? ";
          cin >> accountNumber;
     }

     outCredit.close();
}

void FileRead()
{
     ifstream inCredit("credit.dat", ios::in | ios::binary);

     if (!inCredit)
     {
          cout << "File could not be opened..!!!" << endl;
          exit(1);
     }

     cout << left << setw(10) << "Account " << setw(16) << " Last
Name" << setw(11)
          << "First Name " << left << setw(10) << right << "
Balance" << endl;

     ClientData Client;

     inCredit.read(reinterpret_cast <char*> (&Client),
sizeof(ClientData));
     while (inCredit && !inCredit.eof())
     {
          if (Client.getAccountNumber() != 0)
          {
               outputline(cout, Client);
          }

          inCredit.read(reinterpret_cast <char*> (&Client),
sizeof(ClientData));
     }
}
```

```
void outputline(ostream &output, const ClientData &record)
{
     output << left << setw(10) << record.getAccountNumber() <<
setw(16) << record.getLastName()
           << setw(11) << record.getFirstName() << setw(10) <<
setprecision(2) << right
           << fixed << showpoint << record.getBalance() << endl;
}
```

//End of Program.