#include <iostream>

#include <iomanip>

#include <fstream>

#include <cstdlib> // exit function prototype

#include "ClientData.h" // ClientData class definition

using namespace std;

void outputLine( ostream&, const ClientData & ); // prototype

int main()

{

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

// exit program if ifstream cannot open file

if ( !inCredit )

{

cerr << "File could not be opened." << endl;

exit( EXIT\_FAILURE );

} // end if

// output column heads

ClientData client; // create record

int r;

cout << "Enter record to read ";

cin >> r;

inCredit.seekg (sizeof(ClientData)\*(r-1), ios::cur);

// read first record from file

inCredit.read( reinterpret\_cast< char \* >( &client ),

sizeof( ClientData ) );

cout << left << setw( 10 ) << "Account" << setw( 16 )

<< "Last Name" << setw( 11 ) << "First Name" << left

<< setw( 10 ) << right << "Balance" << endl;

outputLine( cout, client );

// read next from file

#include <iostream>

#include <iomanip>

#include <fstream>

#include <cstdlib> // exit function prototype

#include "ClientData.h" // ClientData class definition

using namespace std;

void outputLine( ostream&, const ClientData & ); // prototype

int main()

{

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

// exit program if ifstream cannot open file

if ( !inCredit )

{

cerr << "File could not be opened." << endl;

exit( EXIT\_FAILURE );

} // end if

// output column heads

ClientData client; // create record

int r;

cout << "Enter record to read ";

cin >> r;

//TODO : Ask three options as first record, last record or the record number

// Read the record based on the user value

inCredit.seekg (sizeof(ClientData)\*(r-1), ios::cur);

// read first record from file

inCredit.read( reinterpret\_cast< char \* >( &client ),

sizeof( ClientData ) );

cout << left << setw( 10 ) << "Account" << setw( 16 )

<< "Last Name" << setw( 11 ) << "First Name" << left

<< setw( 10 ) << right << "Balance" << endl;

outputLine( cout, client );

}

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 function outputLine

// end while

} // end main

// display single record

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 function outputLine