Given:

struct clientData {

int accountNumber;

char lastName[ 16 ]; // c-style string, not string

char firstName[ 11 ];

float balance;

};

Create a random access file of 101 records named “credit.dat” using these code segments in a function to initialize file to empty records:

ofstream creditFile( "credit.dat", ios::out);

clientData blankClient = { 0, "", "", 0.0 };  
  
for ( int i = 0; i <= 100; i++ )  
 creditFile.write( reinterpret\_cast<const char \*>( &blankClient ), sizeof( clientData ) );

creditFile.close();

<ostream> member function *write* - outputs a fixed number of bytes beginning at a specific location in memory to the specific stream (file).

The *write* function expects a first argument of type *const char \*,* hence the use of

*reinterpret\_cast <const char \*>* to convert the address of *blankClient* to *const char \*.*The second argumentof *write* is an integer specifying the number of bytes to be written.Thusthe use of *sizeof( clientData )*. Since size will never change it must be declared as a constant variable; sizeof must only appear once in your program: for the constant declaration.

The first entry in the file will be skipped so that record 1 is at position 1 and not position 0.

Write data into the file [minimum of 15 records, not in 15 consecutive account numbers] getting all data from the user. Truncate user input if it is too long for the data field.

Read data from the file. Loop, asking user for an account number to find (range 1 to 100, 0 to end input. error message if not a valid account number and try again). Print out all data fields for this account.

Update an account. Loop, asking user for an account number to update (error message if not a valid account number and try again) or 0 to quit (do several updates). Update firstname, lastname, or balance (but NOT accountNumber).

Print out all records that do not have accountNumber of 0, formatting output into columns (label columns).