// harsh fianl project.cpp : This file contains the 'main' function. Program execution begins and ends there.

//

#include "pch.h"

#include <iostream>

//this is the world bank of Hedia Credit

#include<iostream>

#include <string>

#include<ctime>

#include<vector>

#include <fstream>

#include <stdlib.h>

//make clear screen...

#include<stdio.h>

#include<stdlib.h>

using namespace std;

char name[20];

//the blockchain will have 3 main components:

//the transaction dats

//the block class

//the blockchain class

//transaction data

struct transaction\_data

{

float amount;

string sender;

string receiver;

time\_t timestamp;

};

//block class

class block

{

private:

int index;

size\_t blockHash; //size\_t is an unsigned integer type. this is what is returned when the standard library's hash function is used

size\_t previousHash;

size\_t generateHash()

{

hash<string> hash1;

hash<size\_t> hash2;

hash<size\_t> finalhash;

string to\_hash = to\_string(data.amount) + data.receiver + data.sender + to\_string(data.timestamp);

return finalhash(hash1(to\_hash) + hash2(previousHash));

};

public:

//constructor

block(int idx, transaction\_data d, size\_t prevHash)

{

index = idx;

data = d;

previousHash = prevHash;

blockHash = generateHash();

};

//to get original hash

size\_t getHash()

{

return blockHash;

}

//to get previous hash

size\_t getpreviousHash()

{

return previousHash;

}

//trasaction data

transaction\_data data;

//to check if the hash is valid

bool isHashValid()

{

return generateHash() == blockHash;

}

};

//blockchain class

class blockchain

{

private:

block createGenesisBlock() //genesis block is the first block of the blockchain

{

time\_t current;

transaction\_data d;

d.amount = 0;

d.receiver = "None";

d.sender = "None";

d.timestamp = time(&current);

hash<int> hash1;

block genesis(0, d, hash1(0));

return genesis;

}

public:

//public chain

vector<block> chain; //Vectors are same as dynamic arrays with the ability to resize itself automatically when an element is inserted or deleted, with their storage being handled automatically by the container. Vector elements are placed in contiguous storage so that they can be accessed and traversed using iterators.

block\* get\_latest\_block();

//constructor

blockchain()

{

block genesis = createGenesisBlock();

chain.push\_back(genesis);

}

//now lets make public functions

void addBlock(transaction\_data d)

{

int index = (int)chain.size() - 1;

block new\_block(index, d, get\_latest\_block()->getHash());

}

bool is\_chain\_valid()

{

vector<block>::iterator it;

int chainlen = (int)chain.size();

for (it = chain.begin(); it != chain.end(); it++)

{

block currentblock = \*it;

if (!currentblock.isHashValid())

{

return false;

}

if (chainlen > 1)

{

block previousblock = \*(it - 1);

if (currentblock.getpreviousHash() != previousblock.getHash())

{

return false;

}

}

}

return true;

}

};

block \*blockchain::get\_latest\_block()

{

return &chain.back();

}

using namespace std;

int main()

{

cout << "\t\t\t\t!!!!!!WORLD BANK OF HEDIA CREDIT!!!!!!!";

cout << "\n";

cout << "\t\t\t\t$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$";

blockchain creditcoin;

char chh;

char ch[5];

int i = 0;

int j = 0;

cout << "\nAre you a already a member? (y/n): ";

cin >> chh;

if (chh == 'y')

{

system("cls");

label2:

cout << "\t\t\t\t!!!!!!WORLD BANK OF HEDIA CREDIT!!!!!!!";

cout << "\n";

cout << "\t\t\t\t$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$";

int l;

cout << "\n1. deposit credit coins ";

cout << "\n2. transfer money ";

cout << "\n3. make new account ";

cout << "\n4. exit";

cout << "\n\n";

cout << "enter choice ";

cin >> l;

switch (l)

{

case 1:

system("cls");

cout << "\t\t\t\t!!!!!!WORLD BANK OF HEDIA CREDIT!!!!!!!";

cout << "\n";

cout << "\t\t\t\t$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$";

cout << "\n";

goto label6;

break;

break;

case 2:

system("cls");

goto label3;

break;

case 3:

system("cls");

cout << "\t\t\t\t!!!!!!WORLD BANK OF HEDIA CREDIT!!!!!!!";

cout << "\n";

cout << "\t\t\t\t$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$";

cout << "\n";

goto label6;

break;

case 4:

system("cls");

goto label5;

break;

}

label3:

cout << "\t\t\t\t!!!!!!WORLD BANK OF HEDIA CREDIT!!!!!!!";

cout << "\n";

cout << "\t\t\t\t$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$";

time\_t data1Time;

transaction\_data data[10];

do

{

cout << "\nwho are you ";

cin >> data[i].sender;

int offset;

string line;

string name;

int left;

int id;

int money;

fstream file;

file.open("C:\\Users\\exam\\Desktop\\users.txt");

if (file.is\_open())

{

while (!file.eof())

{

getline(file, name);

file >> id >> name >> money;

if ((offset = name.find(data[i].sender, 0)) != string::npos)

{

cout << "account found";

cout << endl << name << " has " << money << " in his/her hedia credit account" << endl;

if (money <= 0)

{

cout << "cannot make a transaction";

exit(0);

}

goto label1;

}

}

cout << "sorry could not find account";

int c, d;

for (c = 1; c <= 32761; c++)

for (d = 1; d <= 32761; d++)

{

}

system("cls");

file.close();

}

else

{

cout << "file could not open ";

exit(0);

}

goto label3;

label1:

cout << "\nwho do you want to send it to ";

cin >> data[i].receiver;

cout << "\nenter amount ";

cin >> data[i].amount;

if (money < data[i].amount)

{

cout << "insufficient funds available";

exit(0);

}

left = money - data[i].amount;

cout << "you have " << left << " left in your bank account" << endl;

ofstream files;

files.open("C:\\Users\\exam\\Desktop\\users.txt", ios::app);

files << endl << id << " " << data[i].sender << " " << left;

files.close();

ofstream a;

data[i].timestamp = time(&data1Time);

creditcoin.addBlock(data[i]);

if (creditcoin.is\_chain\_valid() == 1)

{

cout << "\nthe transaction was successful ";

cout << "here is the receipt ";

cout << "\namount: " << data[i].amount;

cout << "\nreceiver: " << data[i].receiver;

a.open("C:\\Users\\exam\\Desktop\\thisisreceipt.txt", ios::out | ios::app);

a << "block" << i << endl;

a << "the sender: " << data[i].sender << endl;

a << "the receiver: " << data[i].receiver << endl;

a << "the sender sent " << data[i].amount << " credit coins to the receiver." << endl;

a.close();

}

else {

cout << "\nerror!!! ";

}

cout << "\ndo you want to continue(yes/no): ";

cin >> ch;

i++;

system("cls");

if (strcmp(ch, "yes") == 0 || strcmp(ch, "Yes") == 0)

{

system("cls");

goto label2;

}

} while (strcmp(ch, "yes") == 0 || strcmp(ch, "Yes") == 0);

goto label5;

}

if (chh == 'n')

{

label6:

int index;

int money;

string name;

ofstream thefile;

thefile.open("C:\\Users\\exam\\Desktop\\users.txt", ios::app);

index = rand() % 1000;

//cout << "enter the id ";

//cin >> index;

cout << "enter name ";

cin >> name;

cout << "enter the amount of money to deposit: ";

cin >> money;

cout << "your id is " << index;

cout << "\n";

thefile << endl << index << " " << name << " " << money;

thefile.close();

char w;

cout << "now do you want to contitue (y/n): ";

cin >> w;

if (w == 'y')

{

system("cls");

goto label2;

}

else

{

system("cls");

cout << "\t\t\t\t!!!!!!WORLD BANK OF HEDIA CREDIT!!!!!!!";

cout << "\n";

cout << "\t\t\t\t$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$";

cout << "\n\nthankyou for banking with us ";

cout << endl;

exit(0);

}

}

label5:

system("cls");

cout << "\t\t\t\t!!!!!!WORLD BANK OF HEDIA CREDIT!!!!!!!";

cout << "\n";

cout << "\t\t\t\t$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$";

cout << "\n\n Thankyou for banking with us ";

cout << endl;

i++;

i = j;

return 0;

}

// Run program: Ctrl + F5 or Debug > Start Without Debugging menu

// Debug program: F5 or Debug > Start Debugging menu

// Tips for Getting Started:

// 1. Use the Solution Explorer window to add/manage files

// 2. Use the Team Explorer window to connect to source control

// 3. Use the Output window to see build output and other messages

// 4. Use the Error List window to view errors

// 5. Go to Project > Add New Item to create new code files, or Project > Add Existing Item to add existing code files to the project

// 6. In the future, to open this project again, go to File > Open > Project and select the .sln file