2] Tossing Coins for a Dollar

For this assignment, you will create a game program using the Coin class from the previous lab. The program should have three instances of the Coin class: one representing a quarter, one representing a dime, and one representing a nickel. When the game begins, your starting balance is $0. During each round of the game, the program will toss the simulated coins. When a coin is tossed, the value of the coin is added to your balance if it lands heads-up. For example, if the quarter lands heads-up, 25 cents is added to your balance. Nothing is added to your balance for coins that land tails-up. The game is over when your balance reaches $1 or more. If your balance is exactly $1, you win the game. You lose if your balance exceeds $1.

Use the same class definition template. Your project should reference to the same class file. If this is complicated, you can create a copy.

The Source Code is in Three Separate Files

Source Code:

File 1 (Named Coin.h):

```
#pragma once
//Coin.h
//DO NOT MODIFY THIS SECTION
#ifndef COIN_H
#define COIN_H
#include <iostream>
class Coin {
private:
	std::string sideup;
public:
	Coin() //constructor
	{
		toss();
	}
	std::string getSideUp() //accesor
	{
		return sideup;
	}
	void toss();
};
#endif // COIN_H
```

File 2 (Named Coin.cpp):

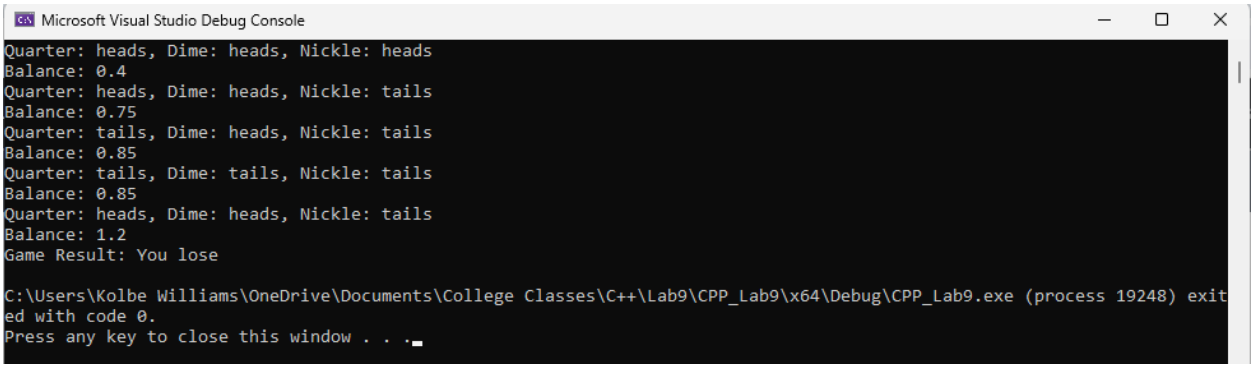
```
#include "Coin.h"
#include <iostream>
#include <ctime>
using namespace std;

void Coin::toss()
{
```

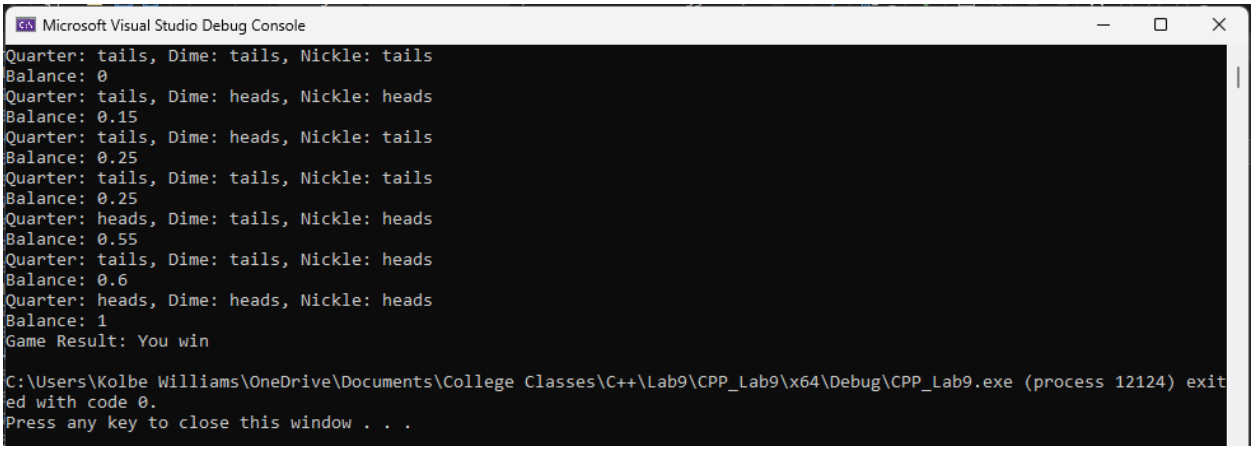
```
	//1. Generate a random number that is either 0 or 1
	int num = rand() % 2;

	//2. Set sideup member variable to either heads or tails depending on the
random number
	if (num == 0)
		Coin::sideup = "heads";
	else
		Coin::sideup = "tails";
}
```

File 3 (Named main.cpp):

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

int main()
{
	srand(time(nullptr));

	//1. Declare and initialize balance variable and create objects of coin class
	double balance = 0.0;
	Coin quarter;
	Coin dime;
	Coin nickle;

	//2. Use a while loop to flip the coins until it is >= 1
	while (balance < 1.0)
	{
		//2.1 Toss each coin and update the balance accordingly
		quarter.toss();
		if (quarter.getSideUp() == "heads")
			balance += 0.25;
		dime.toss();
		if (dime.getSideUp() == "heads")
			balance += 0.10;
		nickle.toss();
		if (nickle.getSideUp() == "heads")
			balance += 0.05;
		//2.2 Print the output of each toss and the balance after each toss
		cout << "Quarter: " << quarter.getSideUp() << ", Dime: " <<
dime.getSideUp()
			<< ", Nickle: " << nickle.getSideUp() << endl;
		cout << "Balance: " << balance << endl;
	}

	//3. Determine if the user won the game by testing if balance is equal to 1
or not and print the result
	if (balance == 1.0)
		cout << "Game Result: You win\n";
	else
		cout << "Game Result: You lose\n";
	return 0;
}
```

Output:


Microsoft Visual Studio Debug Console
Quarter: heads, Dime: heads, Nickle: heads
Balance: 0.4
Quarter: heads, Dime: heads, Nickle: tails
Balance: 0.75
Quarter: tails, Dime: heads, Nickle: tails
Balance: 0.85
Quarter: tails, Dime: tails, Nickle: tails
Balance: 0.85
Quarter: heads, Dime: heads, Nickle: tails
Balance: 1.2
Game Result: You lose
C:\Users\Kolbe Williams\OneDrive\Documents\College Classes\C++\Lab9\CPP_Lab9\x64\Debug\CPP_Lab9.exe (process 19248) exited with code 0.
Press any key to close this window . . .


Microsoft Visual Studio Debug Console
Quarter: tails, Dime: tails, Nickle: tails
Balance: 0
Quarter: tails, Dime: heads, Nickle: heads
Balance: 0.15
Quarter: tails, Dime: heads, Nickle: tails
Balance: 0.25
Quarter: tails, Dime: tails, Nickle: tails
Balance: 0.25
Quarter: heads, Dime: tails, Nickle: heads
Balance: 0.55
Quarter: tails, Dime: tails, Nickle: heads
Balance: 0.6
Quarter: heads, Dime: heads, Nickle: heads
Balance: 1
Game Result: You win
C:\Users\Kolbe Williams\OneDrive\Documents\College Classes\C++\Lab9\CPP_Lab9\x64\Debug\CPP_Lab9.exe (process 12124) exited with code 0.
Press any key to close this window . . .