

Name: _____

Classes and Objects

To be reported on canvas. Create a PDF. Include screenshots of code and execution. Include copy-pasteable text of code. Be careful with variable names and indentation. You must use the templates.

Problem 4. Coin Toss Simulator

Write a class named **Coin**. The **Coin** class should have the following member variable:

- A string named **sideUp**. The **sideUp** member variable will hold either “heads” or “tails” indicating the side of the coin that is facing up.

The **Coin** class should have the following member functions:

- A default constructor that randomly determines the side of the coin that is facing up (“heads” or “tails”) and initializes the **sideUp** member variable accordingly.
- A void member function named **toss** that simulates the tossing of the coin. When the **toss** member function is called, it randomly determines the side of the coin that is facing up (“heads” or “tails”) and sets the **sideUp** member variable accordingly.
- A member function named **getSideUp** that returns the value of the **sideUp** member variable.

Write a program that demonstrates the **Coin** class. The program should create an instance of the class and display the side that is initially facing up. Then, use a loop to toss the coin 10 times. Each time the coin is tossed, display the side that is facing up. The program should keep count of the number of times heads are facing up and the number of times tails are facing up, and display those values after the loop finishes.

USE THE NEXT TEMPLATE (MANDATORY) FOR THE CLASS DEFINITION "Coin.h"

```
//Coin.h
//DO NOT MODIFY THIS SECTION
#ifndef CLASSNAME_H
#define CLASSNAME_H
#include <iostream>

class Coin{
private:
    std::string sideup;
public:
    Coin() //constructor
    { toss(); }
    std::string getSideUp() //accessor
    { return sideup; }
    void toss();
```

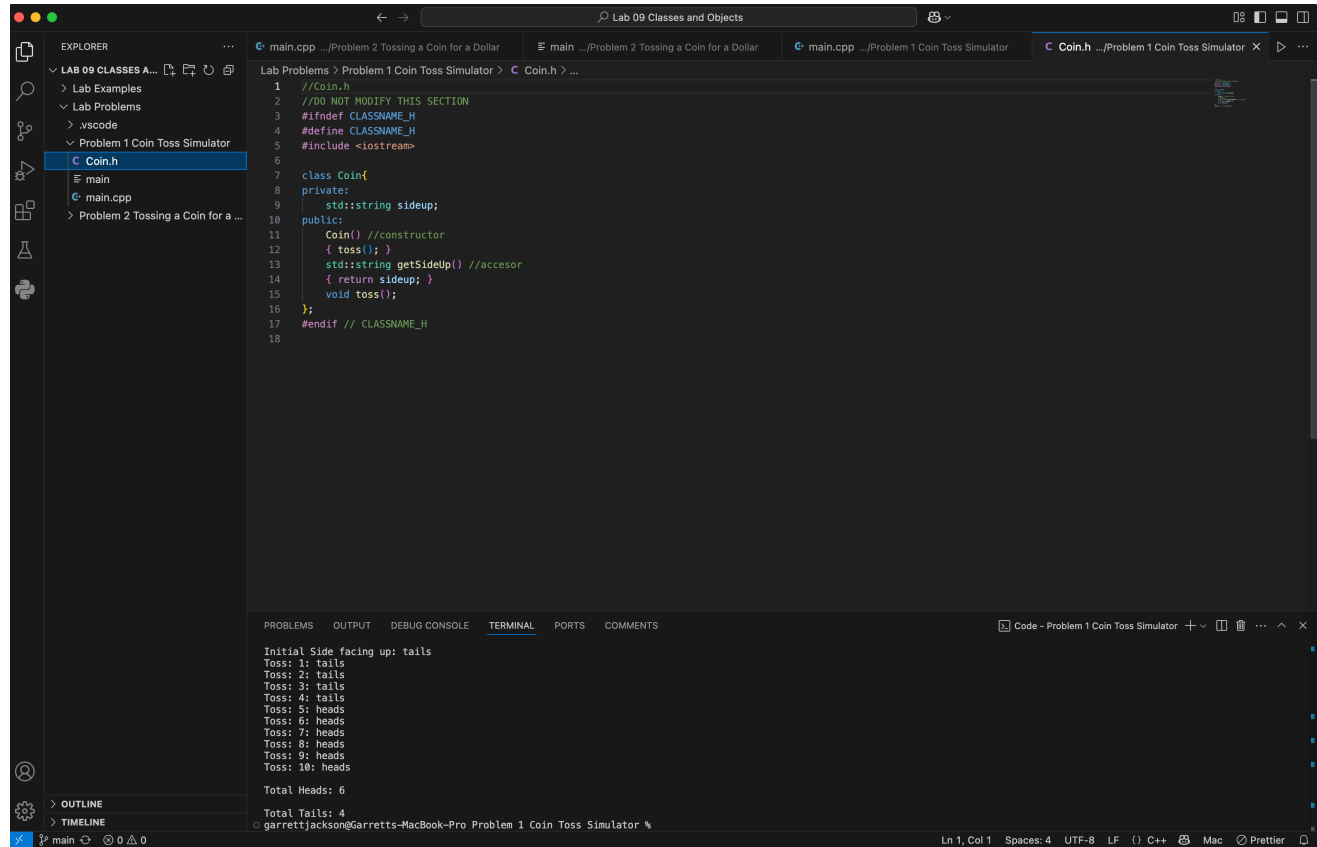
```
};  
#endif // CLASSNAME_H
```

Output:

```
heads  
heads  
heads  
heads  
tails  
heads  
tails  
heads  
heads  
Heads: 8  
Tails: 2
```

The screenshot shows a Visual Studio Code editor window with the following components:

- EXPLORER:** Displays the project structure for 'Lab 09 Classes and Objects', including 'Lab Examples', 'Lab Problems', '.vscode', 'Problem 1 Coin Toss Simulator' (containing 'Coin.h' and 'main.cpp'), 'Problem 2 Tossing a Coin for a ...', and 'Lab 09.pdf'.
- Editor:** Shows the 'main.cpp' file for 'Problem 1 Coin Toss Simulator'. The code includes `<iostream>`, `<cstdlib>`, `<ctime>`, and `"Coin.h"`. It uses the `std` namespace and defines a `main` function. Inside `main`, it seeds a random number generator, creates a `Coin` object, and prints the initial side facing up. It then enters a loop for 10 tosses, printing the result of each toss and updating counters for heads and tails. Finally, it prints the total counts for heads and tails.
- TERMINAL:** Displays the output of the program, showing the initial side facing up (tails), the results of 10 individual tosses (4 heads, 6 tails), and the final totals (Total Heads: 4, Total Tails: 6).
- STATUS BAR:** Shows the current file is 'main.cpp', the editor is in 'main' mode, and the file encoding is UTF-8.



```
1 //Coin.h
2 //DO NOT MODIFY THIS SECTION
3 #ifndef CLASSNAME_H
4 #define CLASSNAME_H
5 #include <iostream>
6
7 class Coin{
8 private:
9     std::string sideup;
10 public:
11     Coin() //constructor
12     { toss(); }
13     std::string getSideUp() //accessor
14     { return sideup; }
15     void toss();
16 };
17 #endif // CLASSNAME_H
18
```

```
Initial Side facing up: tails
Toss: 1: tails
Toss: 2: tails
Toss: 3: tails
Toss: 4: tails
Toss: 5: heads
Toss: 6: heads
Toss: 7: heads
Toss: 8: heads
Toss: 9: heads
Toss: 10: heads
Total Heads: 6
Total Tails: 4
garrettjackson@Garretts-MacBook-Pro Problem 1 Coin Toss Simulator %
```

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

int main()
{
    srand((time(0)));

    Coin flipCoin;

    cout << "Initial Side facing up: " << flipCoin.getSideUp() << endl;

    int headsCount = 0;
    int tailsCount = 0;

    for(int i = 0; i < 10; i++)
    {
        flipCoin.toss();
        string result = flipCoin.getSideUp();
        cout << "Toss: " << i + 1 << ": " << result << endl;

        if(result == "heads")
            headsCount++;
        else
            tailsCount++;
    }
}
```

```
        cout << "\nTotal Heads: " << headsCount << endl
            << "\nTotal Tails: " << tailsCount << endl;

        return 0;
    }

void Coin::toss()
{
    int result = rand() % 2;
    sideup = (result == 0) ? "heads" : "tails";
}
```

Problem 5. Tossing Coins for a Dollar

For this assignment, you will create a game program using the Coin class from the previous problem. 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.

Execution example:

```
Quarter: heads, Dime: heads, Nickel: tails
Balance: 0.35
Quarter: tails, Dime: heads, Nickel: heads
Balance: 0.5
Quarter: tails, Dime: tails, Nickel: heads
Balance: 0.55
Quarter: tails, Dime: tails, Nickel: tails
Balance: 0.55
Quarter: tails, Dime: heads, Nickel: heads
Balance: 0.7
Quarter: tails, Dime: heads, Nickel: tails
Balance: 0.8
Quarter: tails, Dime: heads, Nickel: tails
Balance: 0.9
Quarter: heads, Dime: tails, Nickel: heads
Balance: 1.2
Game result: You lose
```

```
1 //Coin.h
2 //DO NOT MODIFY THIS SECTION
3 #ifndef CLASSNAME_H
4 #define CLASSNAME_H
5 #include <iostream>
6
7 class Coin{
8 private:
9     std::string sideup;
10 public:
11     Coin() //constructor
12     { toss(); }
13     std::string getSideup() //accessor
14     { return sideup; }
15     void toss();
16 };
17 #endif // CLASSNAME_H
18
```

```
65 //Display a message based on the balance total
66 if(balance == 1.0)
67 {
68     cout << "Balance is exactly $1.00.\n\nYou win!" << endl;
69 }else{
70     cout << "Balance is over $1.00.\n\n Your balance is $" << balance << endl;
71 }
72
73 return 0;
74 }
75
76 void Coin::toss()
77 {
78     //Allow the function to generate a random number
79     int result = rand() % 2;
80     //When the modulus of the number is 0 or 1, make the sideup Heads or Tails
81     sideup = (result == 0) ? "heads" : "tails";
82 }
83
```

--- Round 3 ---
Quarter: Tails. No money earned.
Nickel: Tails. No money earned.
Dime: Heads. 10 cents earned
Current balance: \$0.8
--- Round 4 ---
Quarter: Heads. 25 cents earned
Nickel: Heads. 5 cents earned
Dime: Tails. No money earned
Current balance: \$1.1
Balance is over \$1.00.
Your balance is \$1.1

#include <iostream>

#include <ctime>

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

int main()
{
    // Represent the instances of the class
    Coin quarter;
    Coin nickel;
    Coin dime;
    double balance = 0.0;
    int round = 1;

    //Generate a random seed
    srand(time(0));

    // Loop for tossing the coin and counting the amount of rounds played.
    do{
        cout << "\n--- Round " << round << " ---" << endl;

        //Declare the 3 instances for the 3 required coins
        quarter.toss();
        nickel.toss();
        dime.toss();

        // Detect that quarter tossed is a head or tails, if heads add it's value
        if(quarter.getSideUp() == "heads")
        {
            //Add coin value to the balance
            balance += 0.25;
            cout << "Quarter: Heads. 25 cents earned" << endl;
        }else{
            cout << "Quarter: Tails. No money earned." << endl;
        }

        // Detect that nickel tossed is a heads or tails, if heads add it's value
        if(nickel.getSideUp() == "heads")
        {
            //Add coin value to the balance
            balance += 0.05;
            cout << "Nickel: Heads. 5 cents earned" << endl;
        }else{
            cout << "Nickel: Tails. No money earned." << endl;
        }

        // Detect that dime tossed is heads or tails, if heads add it's value
        if(dime.getSideUp() == "heads")
        {
```

```
//Add the coin value to the balance
balance += 0.10;
cout << "Dime: Heads. 10 cents earned" << endl;
}else{
    cout << "Dime: Tails. No money earned" << endl;
}

//Display current balance
cout << "Current balance: $" << balance << endl;

round++;

}while( balance < 1.0);

//Display a message based on the balance total
if(balance == 1.0)
{
    cout << "Balance is exactly $1.00.\n\nYou win!" << endl;
}else{
    cout << "Balance is over $1.00.\n\n Your balance is $" << balance << endl;
}

return 0;
}

void Coin::toss()
{
    //Allow the function to generate a random number
    int result = rand() % 2;
    //When the modulus of the number is 0 or 1, make the sideup Heads or Tails
    sideup = (result == 0) ? "heads" : "tails";
}
```