

# COMSC 200

Summer 2024

## Programming Assignment 3

Worth 12.5 points (1.25% of your grade)

**DUE: Monday, 6/24/24 by 11:59 P.M. on Canvas**

**Late Pass Deadline: Thursday, 6/27/24 by 11:59 P.M. on Canvas**

**NOTE:** Your submission for this assignment should be **one .h file, two .cpp files (see above), and one .pdf file with screenshots of your ten sample runs.**

The following naming convention should be used for naming your **class specification file**: `firstname_lastname_Coin.h`

The following naming convention should be used for naming your **class implementation file**: `firstname_lastname_Coin.cpp`

The following naming convention should be used for naming your **application program file**: `firstname_lastname_200_assign3.cpp`

The following naming convention should be used for naming your **sample runs file**: `firstname_lastname_200_assign3.pdf`

For example, if your first name is “James” and your last name is “Smith”, then your four files would be named **James\_Smith\_Coin.h**, **James\_Smith\_Coin.cpp**, **James\_Smith\_200\_assign3.cpp**, and **James\_Smith\_200\_assign3.pdf**.

**COMMENTS – worth 1.25 points (10%) of your programming assignment grade:**

Your program should have at least **ten (10)** different detailed comments explaining the different parts of your program. Each individual comment should be, at a minimum, a sentence explaining a particular part of your code. You should make each comment as detailed as necessary to fully explain your code. You should also number each of your comments (i.e., comment 1, comment 2, etc.).

**Sample Runs – worth 1.25 points (10%) of your programming assignment grade:**

You should submit screenshots of at least **five (5)** different sample runs of your program. You should also number each of your sample runs (i.e., sample run 1, sample run 2, etc.). There is no user input in this program, so you just need to run your program ten times and take a screenshot of the output on the console screen. **NOTE: Your sample runs should be different from my sample runs shown in this write-up for the programming assignment.**

Your solution should use **all three** files listed above.

The **Coin.h** file (class specification file) should contain the class declaration, the declaration of the member variables, and **prototypes** for the constructor and the member functions. Nothing should be implemented in this file.

The **Coin.cpp** file (class implementation file) should contain the **implementation** of the constructor and the member functions.

The **200\_assign3.cpp** file (application program) should contain the **game logic** for the coin toss game. **You should have multiple functions aside from main in your application program.**

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.

**UML Diagram** for the Coin class:

Coin
-sideUp: string    // Stores either “heads” or “tails”
<<Constructor>> +Coin() +toss() +getSideUp(): string

**Make sure to follow this UML diagram in designing your Coin class!**

## Game Logic (200\_assign3.cpp):

The game works as follows (as you will see in the sample runs below):

The player (you) and the computer both begin with a starting balance of \$0.00. For each round of the game, both the player and the computer toss a quarter, a nickel, and a dime. For each type of coin, if the coin lands on heads, the amount of the coin is added to the player's balance. If the coin lands on tails, however, nothing is added to the player's balance for that coin.

For example:

The player gets heads for the quarter, heads for the dime, and tails for the nickel.  $\$0.25 + \$0.10 = \$0.35$  is added to the player's balance for that round.

The computer gets tails for the quarter, tails for the dime, and heads for the nickel. \$0.05 is added to the computer's balance for that round.

The maximum amount that can be added to a player's balance in a single round is  $\$0.25 + \$0.10 + \$0.05 = \$0.40$  (heads for all three coins). The minimum amount that can be added to a player's balance in a single round is \$0.00 (tails for all three coins).

The game ends when **at least one** player has a balance of \$1.00 or more. The following are conditions for win/loss/tie:

- If both players have the same balance at the end, they tie (note that this means both players not only have the same score, but their shared score is \$1.00 or more)
- If one player has a balance of less than \$1.00 when the game ends, **the other player who has a balance of \$1.00 or more wins**. For example, if the player has \$0.90 and the computer has \$1.05, the computer with \$1.05 wins (remember it's not possible for both players to have less than \$1.00 when the game ends)
- If both players have a balance of \$1.00 or more, **the player with the lower score wins**. For example, if the player ends the game with \$1.10, and the computer ends the game with \$1.30, the player with \$1.10 wins.

As you will see from the sample runs, there is **NO** user input for this game. Again, the game automatically ends when one or both players have a balance of \$1.00 or more.

The output of your program should follow the **same format** shown in the sample runs below. In particular, at the beginning of the program, the starting balance for both the player and the computer should be displayed. Then after the three coins have been tossed for both the player and the computer, the balance for both the player and the computer should be displayed. At the end of the program, the ending balance for both the player and the computer should be displayed. Finally, the outcome of the game (player won, computer won, or tie) should be displayed.

**NOTE:** The first thing that you should do in your application program (200\_assign3.cpp) is **create three different Coin objects**: one for the quarter, one for the dime, and one for the nickel. Remember that both the player and the computer toss the **same coins**, so only a total of three Coin objects are needed.

### Sample Run 1:

```
Your starting balance: $0.00
The computer's starting balance: $0.00

Your balance after round 1: $0.40
The computer's balance after round 1: $0.15

Your balance after round 2: $0.40
The computer's balance after round 2: $0.20

Your balance after round 3: $0.45
The computer's balance after round 3: $0.35

Your balance after round 4: $0.55
The computer's balance after round 4: $0.70

Your balance after round 5: $0.70
The computer's balance after round 5: $0.70

Your balance after round 6: $0.85
The computer's balance after round 6: $0.80

Your balance after round 7: $1.10
The computer's balance after round 7: $1.10

Your ending balance: $1.10
The computer's ending balance: $1.10

Tie! Nobody wins.

Press any key to continue . . .
```

## Sample Run 2:

```
Your starting balance: $0.00
The computer's starting balance: $0.00

Your balance after round 1: $0.15
The computer's balance after round 1: $0.05

Your balance after round 2: $0.20
The computer's balance after round 2: $0.30

Your balance after round 3: $0.50
The computer's balance after round 3: $0.30

Your balance after round 4: $0.65
The computer's balance after round 4: $0.65

Your balance after round 5: $0.95
The computer's balance after round 5: $0.75

Your balance after round 6: $1.10
The computer's balance after round 6: $1.05

Your ending balance: $1.10
The computer's ending balance: $1.05

Sorry! The computer won.

Press any key to continue . . .
```



### Sample Run 3:

```
Your starting balance: $0.00
The computer's starting balance: $0.00

Your balance after round 1: $0.10
The computer's balance after round 1: $0.15

Your balance after round 2: $0.35
The computer's balance after round 2: $0.40

Your balance after round 3: $0.70
The computer's balance after round 3: $0.40

Your balance after round 4: $0.85
The computer's balance after round 4: $0.75

Your balance after round 5: $0.85
The computer's balance after round 5: $0.75

Your balance after round 6: $1.20
The computer's balance after round 6: $0.85

Your ending balance: $1.20
The computer's ending balance: $0.85

Congratulations! You won.

Press any key to continue . . .
```

## Sample Run 4:

```
Your starting balance: $0.00
The computer's starting balance: $0.00

Your balance after round 1: $0.10
The computer's balance after round 1: $0.25

Your balance after round 2: $0.15
The computer's balance after round 2: $0.50

Your balance after round 3: $0.40
The computer's balance after round 3: $0.50

Your balance after round 4: $0.70
The computer's balance after round 4: $0.50

Your balance after round 5: $0.75
The computer's balance after round 5: $0.80

Your balance after round 6: $0.90
The computer's balance after round 6: $0.90

Your balance after round 7: $1.30
The computer's balance after round 7: $0.95

Your ending balance: $1.30
The computer's ending balance: $0.95

Congratulations! You won.

Press any key to continue . . .
```

## Sample Run 5:

```
Your starting balance: $0.00
The computer's starting balance: $0.00

Your balance after round 1: $0.15
The computer's balance after round 1: $0.40

Your balance after round 2: $0.50
The computer's balance after round 2: $0.40

Your balance after round 3: $0.75
The computer's balance after round 3: $0.75

Your balance after round 4: $1.00
The computer's balance after round 4: $0.80

Your ending balance: $1.00
The computer's ending balance: $0.80

Congratulations! You won.

Press any key to continue . . .
```