CS 1331 Homework 7

Due Wednesday October 17, 2011 8:00PM

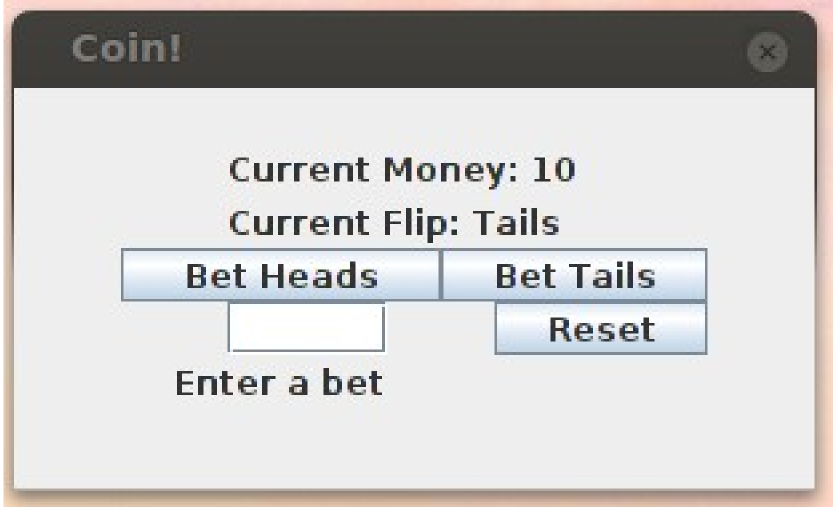
Introduction

This homework will cover buttons, text fields, and event handling.

Be sure to name your classes as required by the instructions. Also be sure to use good coding style and indentation, and to use appropriate and descriptive variable names.

Do not forget javadocing and commenting detailed in homeworks 2 and 3.

Your finished homework might look something like this:



Coin!

Current Money: 10

Current Flip: Tails

5.1: Coin.java

Create a class called `Coin`, which will represent a coin. The coin will have two sides: heads and tails.

1. Create an instance variable to hold the current state of the coin (heads or tails)
   a. You can choose to store this however you see fit: boolean, int, enum...
2. Write a constructor that initializes the coin to heads or tails
3. Complete the following methods:
   a. A flip method that randomly assigns the state of the coin to heads or tails
   b. A getter for the state of the coin

5.2: Player.java

Create a class called `Player` that will represent a player playing our game.

1. Create the following instance variables:
   a. An integer to hold how much money the player has
   b. The coin object that the player is flipping
2. Write a constructor that initializes money to 10 and initializes the coin. Your constructor may (or may not) take in a coin for the player to use.
3. Complete the following methods:
   a. A bet method that takes in a bet and the side of the coin that the player is betting on. It should flip the coin and change the player's money depending on whether the player won or lost the bet.
   b. A getter for the player's money
   c. A reset method that resets money to 10

5.3 CoinPanel.java

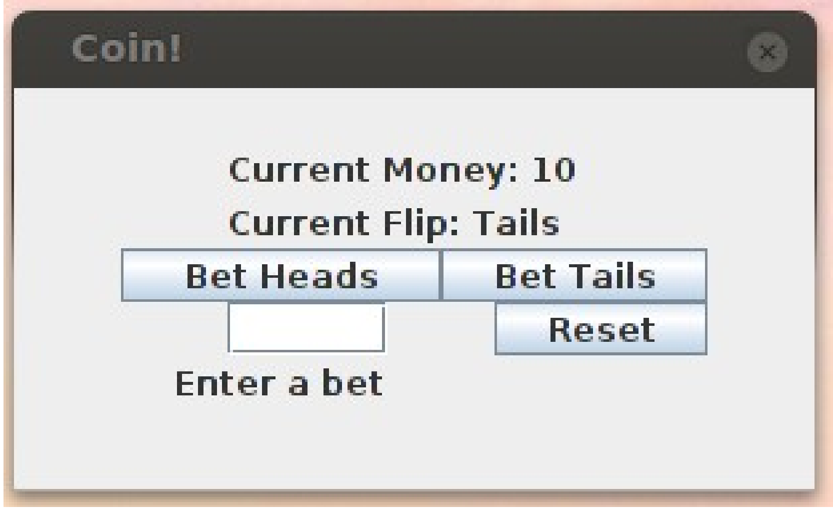
Write a `CoinPanel` class that extends `JPanel` and displays the results of the game. It should meet the following requirements:

- The player's money and the state of the coin is displayed to the `JPanel`
- `JLabels` displays outcome of last toss
- A `JTextField` is displayed for the user to enter a bet
- 2 `JButtons` to bet either heads or tails
  - Each of the Bet buttons must instantiate, using a parameter, a class that implements ActionListener to use as their handler.
  - The listener class should store this passed in value and use it to determine the action it should take (whether it is betting heads or tails).

    - The passed value must be the sole factor evaluated in the actionPerformed method to determine whether you bet heads or tails – you **can not** use get source.
        - Additionally, solutions which write two separate classes for the Bet Heads' and Bet Tails' action listeners will not receive full credit.

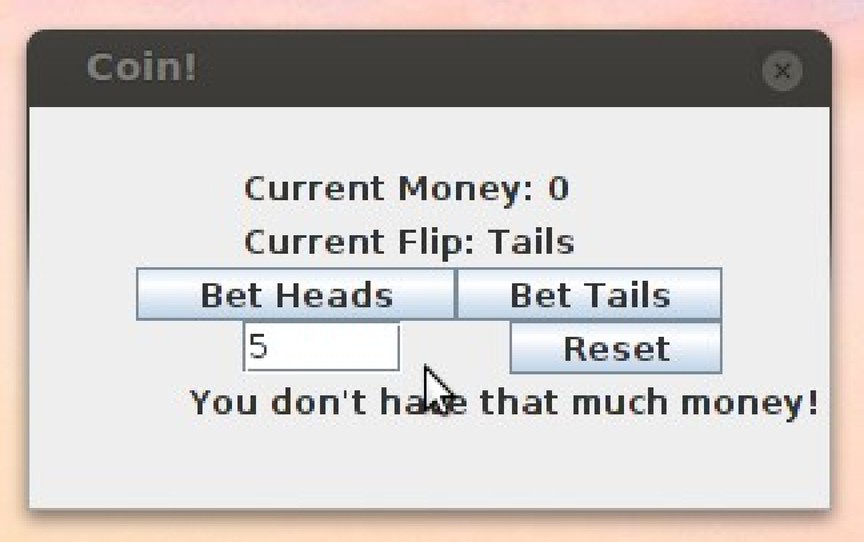
- A `JButton` for the user to reset the money.

The `JPanel` might look something like this:



- When the user hits enter on the text field, or presses the bet button, the coin is flipped using the user’s bet.

- If the user tries to enter a bet that exceeds his current money, a message is displayed that the bet is invalid.



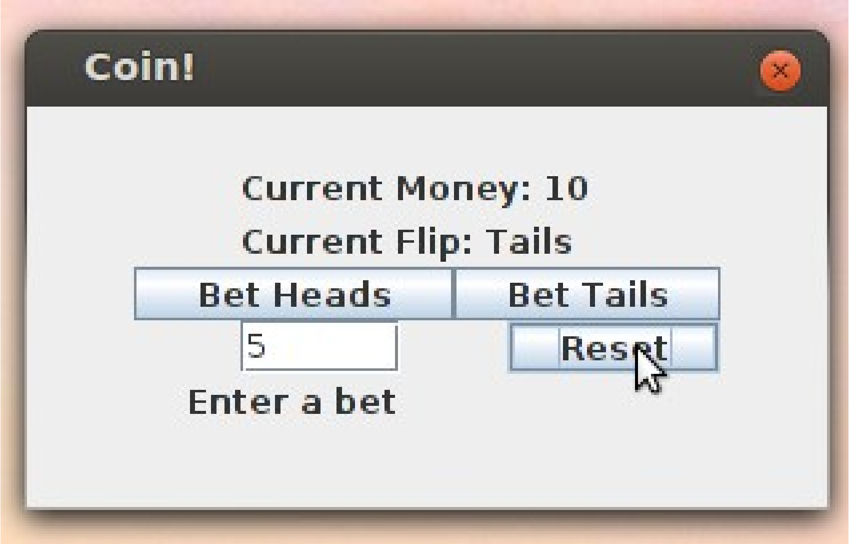
Current Money: 0

Current Flip: Tails

5

You don't have that much money!

- If the user hits “reset,” the money is reset to 10 and displayed.



Current Money: 10

Current Flip: Tails

5

Enter a bet

5.4: CoinPanelMain.java

Finally, create a class named `CoinPanelMain` with a main method that creates a `JFrame` and adds a `CoinPanel` to the frame.

1. Create a `JFrame`, and set the default close operation
2. Create a `CoinPanel`
3. Add the `CoinPanel` to the `JFrame`
4. Pack, and set the visibility of the `JFrame`

Turn-in Procedure

Turn in the following files on T-Square. When you're ready, double-check that you have *submitted* and not just saved as draft.

- Coin.java
- Player.java
- CoinPanel.java
- CoinPanelMain.java
- Any other files needed to run your program

All .java files should have a descriptive javadoc comment.

Don't forget your collaboration statement. You should include a statement with every homework you submit, even if you worked alone.

Verify the Success of Your HW Turn-In

Practice "safe submission"! Verify that your HW files were truly submitted correctly, the upload was successful, and that the files compile and run. It is solely your responsibility to turn in your homework and practice this safe submission safeguard.

1. After uploading the files to T-Square you should receive an email from T-Square listing the names of the files that were uploaded and received. If you do not get the confirmation email almost immediately, something is wrong with your HW submission and/or your email. Even receiving the email does not guarantee that you turned in exactly what you intended.
2. After submitting the files to T-Square, return to the Assignment menu option and this homework. It should show the submitted files.
3. Download copies of your submitted files from the T-Square Assignment page placing them in a new folder.
4. Recompile and test those exact files.
5. This helps guard against a few things.
    a. It helps insure that you turn in the correct files.
    b. It helps you realize if you omit a file or files.**
    (If you do discover that you omitted a file, submit all of your files again, not just the missing one.)
    c. Helps find last minute causes of files not compiling and/or running.

**Note: Missing files will not be given any credit, and non-compiling homework solutions will receive few to zero points. Also recall that late homework (past the grace period of 2 am) will not be accepted regardless of excuse. Treat the due date with respect. The real due date and time is 8 pm Thursday.