Homework 6 : Simple 21

(Deadline as per Coursera)

This homework deals with the following topics:
- Setting up the Java programming environment and Eclipse (if you haven't done so already)
- An introduction to Java programming, syntax, and style
- Getting you started with class-based object oriented programming

Installing/Configuring Java & Eclipse

If needed, please reference "Installing/Configuring Java & Eclipse" for information on installing/configuring Java & Eclipse.

The Assignment: Simple 21

This is a simplified version of a common card game, "21". Blackjack is a better-known variant.

In this game, the dealer deals two "cards" to each player, one hidden, so that only the player who gets it knows what it is, and one face up, so that everyone can see it. (Actually, what the other players see is the total of each other player's cards, not the individual cards.)

There are four players: one human player (the person playing the game) and three computer players. To play, the players take turns requesting cards, trying to get as close to 21 as possible, but not going over 21. A player may pass (ask for no more cards). Once a player has passed, he or she cannot later ask for another card. When all players have passed, the game ends.

The winner is the player who has come closest to 21 without exceeding it. In the case of a tie, or if everyone goes over 21, no one wins.

The game is only played once (so it's actually just one "hand").

The "cards" are the numbers 1 through 10 and they are randomly generated, not drawn from a deck of limited size.

Details

We are providing you with a "skeleton" of the game (simple21.zip) -- the classes and methods have been defined, but just about all of the actual code has been deleted. Your task is to finish the program by adding the necessary code. Javadoc-style comments (included) tell you what must be done in each method.

Below are the classes in the project. Note, each class file declares the **simple21** package (at the top) because the entire program sits in the "simple21" subdirectory. Do not remove the package declaration in each file or change the structure of the program.

- **The *GameControl* Class**

  The *GameControl* class controls the entire game. It has a *main* method which serves as the entry point to the entire program and it will be the file you will "run".

  The *GameControl* class will take care of creating the four players, dealing the initial two cards to each player, controlling the play of the game, and printing the final results of the game.

  The *GameControl* class should have (at least) the following instance variables: *human*, *player1*, *player2*, *player3*, and *random*.

- **The *HumanPlayer* Class**

  The *HumanPlayer* class represents a human player in the game.

  The *HumanPlayer* class will take care of looking at his/her own hidden and visible cards, the other players' visible cards, and deciding whether to take another card.

  The *HumanPlayer* class should have (at least) the following instance variables: *name*, *hiddenCard*, *sumOfVisibleCards*, and *passed*.

- **The *ComputerPlayer* Class**

  The *ComputerPlayer* class represents a computer player in the game.

  The *ComputerPlayer* class will take care of looking at its own hidden and visible cards, the other players' visible cards, and deciding whether to take another card.

The *ComputerPlayer* class should also have (at least) the following instance variables: *name*, *hiddenCard*, *sumOfVisibleCards*, and *passed*.

Note: You are always encouraged to write additional helper methods!

Program Output

Print out what the program is doing as it goes along. Here are some runs of my program -- yours should provide similar information.

```
Welcome to Simple 21!
You'll play against 3 other players (computers).
Try to get as close to 21 as possible, without going over.
What is your name? Brandon

Brandon takes a hidden card.
   (It's a 10)
Brandon takes 9
Player 1 takes a hidden card.
Player 1 takes 10
Player 2 takes a hidden card.
Player 2 takes 2
Player 3 takes a hidden card.
Player 3 takes 8

Brandon has 19 total point(s).
Player 1 has 10 visible point(s).
Player 2 has 2 visible point(s).
Player 3 has 8 visible point(s).

Take another card? n
Brandon passes.
Player 1 passes.
Player 2 takes 10
Player 3 passes.

Player 2 takes 9

Player 2 passes.

Game Over.
Brandon has 19 total point(s).
Player 1 has 20 total point(s).
Player 2 has 26 total point(s).
```

```
Player 3 has 18 total point(s).
Player 1 wins with 20 point(s)!
```

```
Welcome to Simple 21!
You'll play against 3 other players (computers).
Try to get as close to 21 as possible, without going over.
What is your name? Brandon

Brandon takes a hidden card.
   (It's a 7)
Brandon takes 9
Player 1 takes a hidden card.
Player 1 takes 10
Player 2 takes a hidden card.
Player 2 takes 10
Player 3 takes a hidden card.
Player 3 takes 7

Brandon has 16 total point(s).
Player 1 has 10 visible point(s).
Player 2 has 10 visible point(s).
Player 3 has 7 visible point(s).

Take another card? y
Brandon takes 5
Player 1 takes 3
Player 2 passes.
Player 3 takes 9

Brandon has 21 total point(s).
Player 1 has 13 visible point(s).
Player 2 has 10 visible point(s).
Player 3 has 16 visible point(s).

Take another card? n
Brandon passes.
Player 1 takes 6
Player 3 passes.

Player 1 passes.

Game Over.
Brandon has 21 total point(s).
Player 1 has 21 total point(s).
Player 2 has 20 total point(s).
Player 3 has 26 total point(s).
Tie, nobody wins.
```

```
Welcome to Simple 21!
You'll play against 3 other players (computers).
Try to get as close to 21 as possible, without going over.
What is your name? Brandon

Brandon takes a hidden card.
   (It's a 8)
Brandon takes 7
Player 1 takes a hidden card.
Player 1 takes 10
Player 2 takes a hidden card.
Player 2 takes 5
Player 3 takes a hidden card.
Player 3 takes 3

Brandon has 15 total point(s).
Player 1 has 10 visible point(s).
Player 2 has 5 visible point(s).
Player 3 has 3 visible point(s).

Take another card? y
Brandon takes 7
Player 1 passes.
Player 2 takes 5
Player 3 takes 10

Brandon has 22 total point(s).
Player 1 has 10 visible point(s).
Player 2 has 10 visible point(s).
Player 3 has 13 visible point(s).

Take another card? n
Brandon passes.
Player 2 passes.
Player 3 takes 6

Player 3 passes.

Game Over.
Brandon has 22 total point(s).
Player 1 has 20 total point(s).
Player 2 has 19 total point(s).
Player 3 has 21 total point(s).
Player 3 wins with 21 point(s)!
```

Comments & Style

Write comments using // for any non-trivial lines of code. In general, all of the style conventions from Python also apply in Java. The main differences are in naming conventions (lowercase and underscores versus camelCase) and the syntax for comments (/* */ versus ''' ''' and // versus #). The content of your comments should be very similar and the length of your lines should not go off the right hand edge of the editing window.

Javadocs

If not already added, please add javadocs to all class definitions, methods, and instance variables.

What to Submit

Please submit all the classes in your entire Java project. Make sure it includes everything in your "src" folder. Do not remove the package declaration in each file and keep the entire program in the "simple21" subdirectory.

Evaluation

1. Did you set up the project correctly? Does it compile and is everything named correctly? (2 pts)
2. Does your code function? Does it do what the specifications require? (9 pts)
    a. Can we play the game? Does the print out make sense?
    b. Does the game accurately pick the winner (if there is one)?
3. Does your code pass our tests? **(**Autograded**)** (5 pts)
4. Did you follow good programming practices? (3 pts)
    a. Did you reuse code to avoid repetition (e.g. put repeated code in a helper method)?
    b. Did you add javadocs to methods and comment all non-trivial code?
    c. Did you indent properly (Cmd+i or Ctrl+i) and use { brackets } correctly for code blocks?
    d. Did you name variables descriptively with camelCase?
5. Did you submit the correct file(s) and nothing else? (1 pt)