Project 3
CSC 1052 - Algorithms and Data Structures II
Grading: 100 points
Due Date: May 4th, 2017

Description: In this project, you will be practicing the use of Java lists and `enum` types by implementing a command line Blackjack game. Using the classes `Card` and `CardDeck`, which will be provided, you will implement a simulator that plays a simplified hand of Blackjack against a computer dealer. The program will terminate when either you win, the dealer wins, or you tie.

Part 1 - Blackjack simulator (80 points)

You have been given two classes from the textbook which have been slightly modified to use Java ArrayLists instead of the textbook implementations of lists (you will also use Java library lists in your project if your implementation requires them). Your job is to simulate a hand of Blackjack according to the following rules. Implement your program in the `main()` method of a class `Blackjack.java`. You will NOT need to modify `Card` or `CardDeck`:

1. The goal of the game is to get the sum of cards in your hand higher than the dealer's without going over 21. Each card is worth the rank value, where face cards are worth 10 and aces are worth either 1 or 11.

2. The game begins by shuffling and dealing two cards to you and two cards to the dealer. The program should reveal both cards in your hand but only the *first* card of the dealer's hand (the other card, called the "hole" card, remains hidden).

3. After the initial cards are dealt, you will repeatedly ask the player if they would like to "hit" or "stay". If the player hits, the program deals them another card and repeats the query. If the player stays, this phase of the game ends. If, when the player asks to hit, the sum value of the cards in their hand exceeds 21, the game immediately ends and the player loses.

4. Once the player stays and has a hand of 21 or less, the dealer plays their hand. The rules for the dealer are that they must "hit" until the sum value in their hand is greater than or equal to 17.

5. If the player's hand is greater than the dealer's OR if the dealer goes over 21, the player wins. If the dealer's hand is greater than the player's the dealer wins. If both hands are equal, the game is a tie.

Here are some tips to help you with your project:

1. Implement your code in phases and test before moving on to the next phase

2. Print out the the player's hand and the value of the hand *every time it changes.*

3. Clearly comment each phase of the game so that I will be able to award full credit for your last project

4. A simple way to terminate your program immediately is with the `System.exit(0)` command. The "0" input parameter indicates normal program termination.

5. You will have to use the `enum` construction as implemented in the class `Card`. Remember, these enumerated types can be treated just like objects, with method calls on the `enum` to learn more information about it.

6. Because aces may be 1 or 11, determining the value of a hand will be challenging. You should implement a separate method `countHand()` that takes your current hand and calculates the highest possible sum that does not exceed 21 (if all possible sums exceed 21, return the lowest, you've gone over). Think carefully about how many possible combinations of aces exist and how many you need to check in `countHand()`.

7. Use the debugger to find logic errors!

Part 2 - PDF write up (20 points)
Create a 1-2 page write up that summarizes the interesting parts of your program. Include any problems or insights that you encountered.

Deliverables: Submit `Blackjack.java` and your write-up on blackboard.