COMP 1409 – Assignment #4 (90 points)
Due: 11:55 p.m. the night before session 11

1409 Blackjack

Your final assignment is a simplified game similar to Blackjack to be played by a single player. The player will give his/her instructions through the keyboard. This game does not involve betting and the player is not playing against a dealer.

You have been provided with a shell game consisting of four classes:

- Card is complete and requires no changes. This class models one card.
- Deck has an instance variable and several "stub" methods, with comments and signatures. You will complete these methods. This class models a deck of 52 cards.
- InputReader is complete and requires no changes. This class provides a reader to get keyboard input from the user.
- Game has some suggested fields and methods. This class contains the game logic. You will add more private methods (and probably fields) to this class. The game is run from the play() method, which should be the only public method in the class.

The object of our simplified game is to get 21 points in a hand of cards. A hand that goes over 21 is a "bust". If the player busts, the player loses.

The game will begin with a new, shuffled deck. The game will end when the player chooses to stop.

Here are the specifications for a round of 1409 Blackjack:

- Deal two cards to the player.
- Show the player's cards by writing their information to the terminal, e.g.
  ```
  player's hand:
  Queen of Hearts
  Nine of Spades
  ```
- Calculate and display the value of the hand. Face cards (king, queen, jack) are counted as 10. An ace is counted as 11 unless that would cause a bust, in which case it is counted as 1. Other number cards are counted at their numeric value.
- If the player's cards total 21, that constitutes Blackjack and is an immediate win (see below).
- If the player has not scored Blackjack, the player decides whether to "hit" or "stand". Prompt the player to type their choice from the keyboard, and read in their choice. A hit consists of another card dealt from the deck. The player's turn continues until the player decides to stand or the player busts (goes over 21).

- Determine the player's points. A bust is zero points. A stand with a hand of less than 16 is zero points. A stand with a hand of less than 21 is one point. Blackjack (a hand of exactly 21) is five points.
- Ask the player if they want to play another round. Continue until the player chooses to stop.
- If the deck runs out of cards during play, replace it with a new shuffled deck.
- Report the total points earned and the total number of rounds played.

You must use the Game, Card, Deck and InputReader classes provided.

Suggested plan of attack:
- Start with the Deck class. It stores Card objects in an ArrayList, and has several methods to be completed. Test these methods by creating a Deck object in BlueJ. Also use BlueJ's inspect option.
- Work out the logic for the Game class.
- Start early so that you have time to ask questions.
- Post questions in WebCT.
- When you get stuck, walk away and think about the problem while doing something else.
- Test thoroughly.

What to turn in:
- Your completed project.
- A Word (or Wordpad) document describing how you have tested your project. This document should contain a list of any bugs and/or missing functionality that still exist at the time you turn in your project. If you have tested your game thoroughly and have concluded that nothing is missing, provide a brief statement to this effect.

Marks will be given for:
- Comments – appropriate and complete.
- Style – see the style guide Appendix J in your textbook.
- Correctness and completeness – code meets the requirements listed above.
- Testing – documentation of testing is thorough, and any remaining bugs are reported.

Create a .zip file containing your entire BlueJ project and the Word document. Name the .zip file with your name and the assignment number, e.g. "Susan_Wong_Assign_4.zip". Upload the file to WebCT before the cutoff time.