Project 1 – Blackjack
Due: Sunday, October 13, 2013 by 11:59pm

Blackjack (also known as 21) is a multiplayer card game, with fairly simple rules. For this assignment, you will be implementing a simplified version where a user can play against the computer who acts as dealer.

Two cards are dealt to each player. The dealer shows one card face up, and the other is face down. The player gets to see both of his or her cards and the total of them is added. Face cards (Kings, Queens, and Jacks) are worth 10 points, Aces are worth 1 or 11 points, and all other cards are worth their face value. The goal of the game is to get as close to 21 ("blackjack") without going over (called "busting.")

The human player goes first, making his or her decisions based on the single dealer card showing. The player has two choices: Hit or Stand. Hit means to take another card. Stand means that the player wishes no more cards, and ends the turn, allowing for the dealer to play.

The dealer must hit if their card total is less than 17, and must stand if it is 17 or higher.

Whichever player gets closest to 21 without exceeding it, wins.

Example Run

```
Welcome to Jon's Casino!
Please enter your name: Jonathan

The dealer:
? + 10
Jonathan:
4 + 10 = 14

Would you like to "hit" or "stand"? hit

The dealer:
? + 10
Jonathan:
14 + 10 = 24 BUSTED!

You busted. Dealer wins.
Would you like to play again? no
```

Requirements

For this assignment you need to do the following:

- Write a program that plays Blackjack
- Have the program intelligently determine if an Ace should be interpreted as a 1 or an 11. You need to be able to also handle multiple aces. If there are any aces in the hand, and the total exceeds 21, change the 11 to a 1 until there are no more aces worth 11 in the hand or the total is below 21.
- You need to be able to deal from a real deck. That means you need to get the right distribution of cards as well as not double dealing the same card (or > 4 of the same value since we're ignoring suit).

Hints:

- Read the user's responses as a string
- For syscall help, go to recitation, check the class website examples, and visit the MARS documentation page: http://courses.missouristate.edu/KenVollmar/MARS/Help/SyscallHelp.html

Submission

Create a zip file of your source code and a README.txt text file that has your name, your Pitt username and any notes for the TA to help with grading.

Name the file `USERNAME-project1.zip`

Use the lab submission policies to upload your project.