Project 2—Blackjack

Due: **7/31/2015** (Sunday @ noon)

Overview

You will be writing a simulated Blackjack game that can take will enable you to practice your 21 skills versus a computer and multiple players.

Specifications

The program *should* have four classes (BlackJackGame, BlackJackPlayer, Deck, Card) to simulate the Blackjack game; its ok to have 3 classes but you need to use at least three. The BlackJackGame class is the class that contains the main.

Each player is dealt two cards face up while the dealer is dealt one card up & one card down (hidden). The player then decides if they want to hit or stay. If your cards add up to be more than 21, then you bust and lose. The dealer has to hit if the dealer has 17 or less. Any player with a higher count than the dealer and less than or equal to 21. If you tie the dealer, then it's a push and you get your bet back.

Each player starts off with 10 chips when they begin the game. The player gets one extra chip if they hit exactly 21. The game is over when the player has 0 chips or they choose to leave the table. You must display the number of chips each player has every time you print their name. You can have each bet be just one chip or allow for multiple chips to be bet *before* the cards are dealt.

After each play, you need to display all the cards on the table. You can only use one deck (each deck contains 52 cards) so that means the same card can't be played twice.

Example

Welcome to James Blackjack game. Cards are dealt.

James's (10) cards: King of Spades | Eight of Spades

Dealer's cards: [hidden] | Ten of Clubs

Hit or Stay? (Enter H or S): S

The Dealer stays

James's (10) cards: King of Spades | Eight of Spades

Dealer's cards: King of Clubs | Ten of Clubs

Dealer wins.

GRADING

Requirement	Points
Able to play 21 vs computer	20
Work on it Solo	10
Used Classes & Objects	10
Have Multiple Players	10
Handle Invalid Input	10
Use toString for Card information	10
Be able to shuffle deck	5
Use ArrayLists to add cards to each Player/Dealer	5
Use comments for methods	5
Handle Double Down	5
Handle Splitting	5
Handle Surrender	5
Total	**100**

GETTING STARTED

Your main should go into the BlackJackGame class. In your main, you should create a new Deck object that contains all the cards (and also allows you to be dealt a card). The Deck object should also allow you to shuffle cards. Back in your main, you should deal out the cards to each Player. The card played to each player should come out of the Deck and is added to a list maintained in the Player object. You then should read in from the console to get input on if the player wants to hit/stay/double down/split/surrender.

RULES

- Dealer must hit on 17 (or anything below)
- Once someone busts, it doesn't matter what the dealer gets
- Double down is when you increase your bet token by double (and have to have more than one token)
- If you get 21, you get 1 extra token
- To split, you have to have more than one token
- Surrendering is allowed.

CHECKPOINTS

Working on a software project is easier if you have intermediate goals to measure your progress against. If you can show a certain amount of progress by each checkpoint. If you are ahead of the schedule, even better!

- Checkpoint 1—**7/27**: main loop that can print out the whole shuffled deck
- Checkpoint 2—**7/29**: able to add cards to each player, determine who won
- Checkpoint 3—**7/31**: able to handle double down, splitting and surrenders.

Extra Features

- Store results to a file, allow user to load previous game results (number of tokens).