Jose Capestany

CS126L Section 4

5/13/2016

Lab 12: Blackjack

**1. Problem Statement**

The purpose of this lab is to simulate a game of blackjack by using objects and classes. I have to create BlackjackHand class to hold cards as well as a Blackjack class to simulate the game.

**Requirements (BlackjackHand):**

* **Be able to add cards to the hand**
* **Return the hand as a string**
* **Find the value of the hand**

**Requirements (Blackjack):**

* **Have a chip bank**
* **Be able to draw a card**
* **Hand out starting hands**
* **Be able to HIT or STAND**
* **End the game and win/lose money**

**2. Planning**

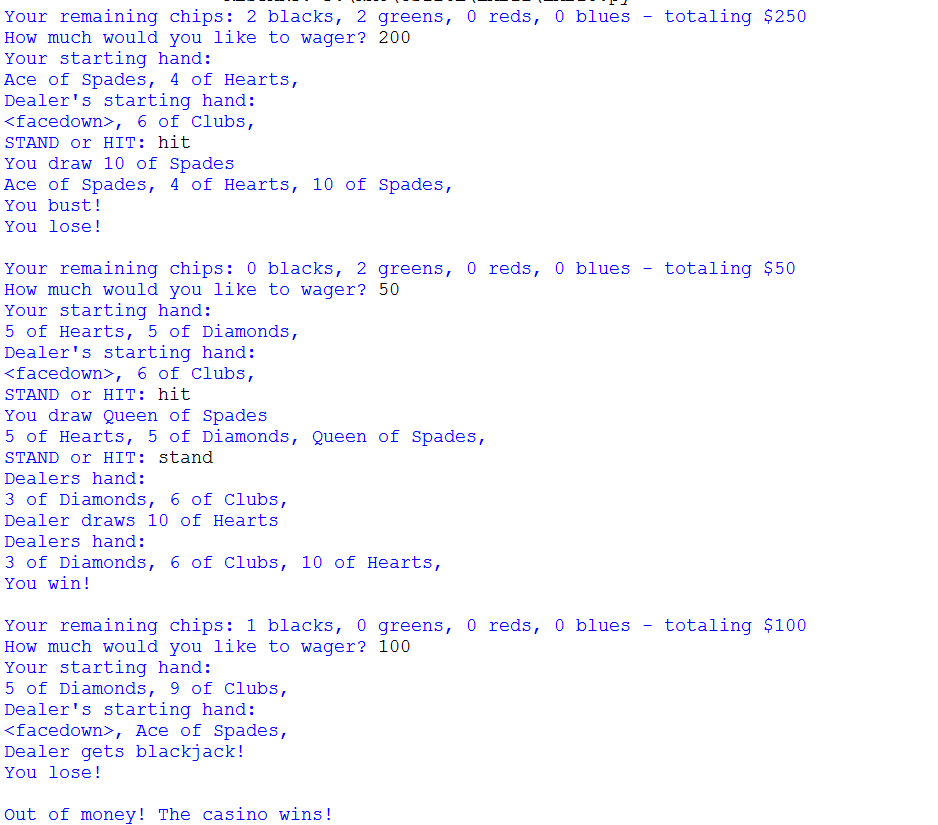
Planning the BlackjackHand class is easy. Adding cards will just use append and finding the value can be done using a simple for loop. Planning the Blackjack class was a little harder. For the draw method I will use random.choice to pick from the deck and then delete the card from the deck. For start\_hand method I planned to just draw two card for the dealer and the player and put it in their hands. For the stand method, I planned for the dealer to just keep drawing until they win or lose. For the hit method, the player will draw a card.

**3. Implementation and Testing**

I was able to implement the methods with the planning I listed in the previous step. I implemented different end\_hand conditions if the user or dealer get blackjack in their starting hand. I mainly used if and elif statements to check for end game conditions. I utilized the previous classes from lab 10 a lot, such as the face\_down method and creating cards with the Card class.

C:\Users\Jose Capestany\Downloads\Capture (1).PNG

*Showing pep8 Compliance*



*Results from the Program*



*Sample of Source Code*

**4. Reflection**

This lab was fairly difficult. It took me a while to figure out how to create a hand and how to add cards to the hand. This is a fun project and I played this a lot. The formatting could use a little a work, but it works. My only peeve on this is that in line 151 I had to access a hidden variable. I couldn’t think of any way to get the job done using methods.