CSE 231 Fall 2008

Programming Project 9

Assignment Overview
This assignment will give you more experience on the use of classes.

This assignment is worth 50 points (5.0% of the course grade) and must be **completed and turned in before 11:59 on Monday, Nov 3rd , 2008.**

Background
You will implement the Solitaire card game called Golf Relaxed. It is a fairly simple game and does not have much strategy, but our goal is to simply enforce the rules for players. To see a copy of the rules, goto http://www.worldofsolitaire.com , click Solitaire at the left-upper corner, click “select game”, and choose “Golf Relaxed”.

More important than anything, familiarize yourself with the game by playing the online version before considering programming. It is much easier to understand the game by playing it rather than reading the rules. A sample implementation, golfRelaxed, is provided.

The game play proceeds as follows:

1. The Start
    a. There is group of cards called the **stock**. After setup, the stock starts with 16 cards.
    b. There is group of cards called the **tableau**. There are 7 rows and initially 5 cards in each row.
    c. There is a group of cards called the **foundation** where cards are placed after being removed from the tableau. The foundation starts with only one card.
2. The Goal
    a. The goal is to move all the cards in tableau to the foundation
3. Rules of Play
    a. You can move the last card in any tableau row to the foundation when the rank of the card you are moving is **adjacent in rank** (**suit does not matter**) of the last card of the foundation. For example, , 5 is adjacent to both a 6 and a 4. 2 is adjacent to both Ace and 3. One special note, Ace is adjacent to both a 2 and a King.
    b. At any time, as long as there is at least one card left in stock, you can deal one card from the stock to the foundation.
    c. All cards in the tableau are visible, but only the last card in each row can be moved.
    d. None of the cards in the stock are visible
    e. Only the last card in the foundation can be played against.
    f. If there are no cards left in any row of the tableau, you win.
    g. If you run out of stock, cards remain in the tableau

Your program allows a user to play the game, ensuring that they follow the rules.

Requirements
Implement the game in Python. You can also try the example game golfRelaxed to get a feel for the game. A proj09Skel is provided for a possible approach to write your solution. Up to you if you use it or not.

Requirements are:

1. Use the provided Card and Deck Class, found in the cards.py file in the project directory.
2. You must use functions in this game. Implementations without functions will not be graded.

3. Create one function to be called play(), which starts the play of the game.
4. If the user makes a move that is illegal, you must inform them of the error and reset the game to the previous position.
5. You must determine if a winning position is achieved. If so, report it and stop the game, printing out a “Winning” message.
6. If you run out of cards to deal and cannot generate the winning position, the game ends and the user loses. You must print out a “Loser” message at that point.
7. The provided demo uses the following prompts:
    a. Number 1-7 means to move the card at the end of that row to the foundation
    b. “d” means to deal 1 more card from stock to foundation
    c. “q” means quit

Card and Deck Classes plus Display function
We provide a module named ***cards*** that contains a card class, a deck class and a display function. We also provide a sample piece of code that demonstrates how to use the cards module. The card and deck classes are general purpose for developing card games so they contain many methods that may not be used in any particular implementation. You are welcome to use all of them, but do not be surprised if there are many that you do not need for this project. For example, in my implementation I used the get_rank() of Card (and the print function through __str__) and the deal(), empty(), shuffle() and constructor method of Deck. That’s about it.

The get_rank() method returns the rank of the card: 1 for ace, 2-10 for number cards 2-10 (respectively), 11 for Jack, 12 for Queen, 13 for King. Unfortunately, Ace’s both adjacent to King and 2 in this game. You will have to deal with that in your program, not change the class.

Deliverables
You must use handin to turn in the file **proj09.py** – this is your source code solution; be sure to include your section, the date, the project number and comments describing your code. Please be sure to use the specified file name, and save a copy of your proj09.py file to your H drive as a backup.

Other good information
1. Play with the provided demo program and get a feel for the game
2. Look carefully at the example cardsDemo.py program. It imports the cards module and uses the two classes and gives you a better idea how you can use them. These classes provide methods you may not need, but they should provide almost any method you do need.
3. When using class methods remember the parenthesis—no error is generated for missing parenthesis, but results will not be what you expect.
4. There are multiple parts to the game (setup, printing, game play, starting). Address each one individually and then put them together.
5. When you deal a card from an empty deck, a card is returned but its rank is 0 and its suit is an empty string. Yet, you should avoid this situation. It’s better to ensure that the deck is not empty before actually deal a card from it.
6. For playing the game, begin by assuming perfect input. Get that working and add error checking later.
7. Add as much error checking as you can. Error conditions could occupy quite a bit of code! The demo program checks for the following errors:
    a) trying to move a card off to the foundation that violates the rules
    b) checking on user input to capture incorrect commands (numbers not in the range 1-7, not a *d* or *q*)
    c) trying to deal more cards when there are no cards left in the stock