

## 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, March 25<sup>th</sup>, 2013.**

### Background

You will implement the Seahaven Towers card game. Our goal is to simply enforce the rules for players. To see a copy of the rules and play the game, goto <http://www.greenfelt.net/sht>

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

The game play proceeds as follows. Note that our rules are slightly more restrictive than the online rules in that we allow only one card at a time to be moved (makes your program easier to write):

1. The Start
  - a. One standard 52-card deck is used.
  - b. All cards are initially dealt to the tableau. All cards are visible. The tableau has 10 columns. (the first two columns have six cards, the rest have five.)
  - c. There are 4 foundations – one foundation for each suit. Initially they are empty.
  - d. There are 4 cells. Initially they are empty. Each cell can **only contain 1 card**. The cells are places to temporarily place cards during the play of the game.
2. The Goal
  - a. Move all the cards to the Foundations.
3. Rules of Play
  - a. Only one card at a time can be moved.
  - b. Foundation
    - i. Each foundation holds only one suit and is built up from Ace(low) to King.
    - ii. You can move a card to the foundation from a cell or the tableau.
    - iii. Once a card is on the foundation it cannot be moved off.
  - c. Tableau
    - i. The card at the bottom of a column may be moved to an open cell, a foundation or another column of the tableau.
    - ii. Moving a card from one tableau column to another follows these rules
      1. A card can only be moved to the bottom of a column
      2. When you move a card to a column in the tableau you can only build down by rank and by the same color. For example, you can move a Two of Hearts onto a Three of Hearts (the pile goes down by rank, and same color)
    - iii. Empty columns may be filled only by a King but of any suit
  - d. Cell
    - i. One cell spot can only contain 1 card
    - ii. The card may be moved to the tableau or the foundation.

Your program allows a user to play the game, ensuring that they follow the rules. It does not play the game itself, just allows the user to follow the rules!!!

## Requirements

Implement the game in Python. You can also try the example game *seahaven.py* to get a feel for the game. To run that you need to copy **Both** *seahaven.py* and *seahaven\_cpython.pyc* to your computer. Requirements are:

1. Use the provided Card and Deck classes, found in the cards.py file in the project directory. **Do not modify the cardsBasic .py program** as you will only turn in proj09.py. Therefore your solution must `import cardsBasic` into your proj09.py.
2. You must use ***functions*** in this game.
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 let user choose another move.
5. You must determine if a winning position is achieved. If so, report it and stop the game, printing out a “Winning” message.
6. The provided demo uses the following prompts, **which you are also required to use**:
  - t2f T F - move from Tableau T to Foundation F (T and F are ints)
  - t2c T C - move from Tableau T to Cell C (T and C are ints)
  - t2t T1 T2 - move from Tableau T1 to Tableau T2 (T1 and T2 are ints)
  - c2t C T - move from Cell C to Tableau T (C and T are ints)
  - c2f C F - move from Cell C to Foundation F (C and F are ints)
  - 'h' for help
  - 'q' to quit

The demo program checks for the following errors, **which you are also required to check for**:

  - a) trying to move cards that violate the rules
  - b) checking on user input to capture incorrect commands
  - c) whether the move command is in correct format
7. The demo program provides a template for the output format of your program. **You are required to use the output format of the demo program for your program**

## How to Get Started

To provide a starting point we have provided a framework named *seahavenStart.py*. Using this framework is optional, and modifying it is allowed as long as you meet the game specifications (especially printing the game). However, it sure makes it easier to start here!!!

## Card and Deck Classes plus Display function

We provide a module named *cardsBasic* that contains a Card class and a Deck class. We also provide a sample piece of code that demonstrates how to use the cardsBasic module. 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. Note the method `equal_suit()` as well.

## Deliverables

You must use handin to turn in the file **proj08.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 proj08.py file to your H drive as a backup.

## Other good information

Notes:

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. For playing the game, begin by assuming perfect input. Get that working and add error checking later.
6. Add as much error checking as you can. Error conditions could occupy quite a bit of code!
7. The program does not need to print every card in the foundations. Only printing the top card will be ok.