Data Structures Homework #2 Due: Wednesday, Feb. 9 (11:59 PM)

You are to write a Python program that allows a user to play a video poker game that uses dice. The base set of rules is as follows:

- A player starts with $100.
- Each round costs $10 to play. This amount is subtracted from the user's money at the start of the round.
- The player starts the round with five randomly rolled dice.
- The player gets two chances to enhance the hand by rerolling some or all of the five dice.
- At the end of the hand, the player's money is updated according to the following payout schedule:

Hand	Example	Payout
Two Pairs	4, 4, 3, 6, 6	$ 5
Three of a Kind	2, 4, 1, 1, 1	$ 8
Full House (a pair and a three of a kind)	3, 5, 5, 5, 3	$ 12
Four of a Kind	3, 3, 3, 3, 6	$ 15
Straight (1-5 or 2-6)	5, 2, 3, 1, 4	$ 20
Five of a Kind	3, 3, 3, 3, 3	$ 30

A text-based user-interface might go something like this:

```
Welcome to video poker!
You currently have $100.
It costs $10 to play. Do you wish to try your luck (y or n)?  y
Roll 1 Dice: [6, 4, 4, 2, 4]
Positions:    1  2  3  4  5
Enter the positions of the dice you want to reroll (or Enter to stop):  1  4
Roll 2 Dice: [3, 4, 4, 4, 4]
Positions:    1  2  3  4  5
Enter the positions of the dice you want to reroll (or Enter to stop):  1
Roll 3 Dice: [2, 4, 4, 4, 4]
Four of a Kind.  You win $15!
You currently have $105.
Do you wish to try your luck (y or n)? y
. . .
```

Implementation Requirements:

- Use a list of Die or AdvancedDie objects from lab 3 to represent a hand.
- Think about the design before you start to write code. You can use either an object-based design which uses the Die objects in a procedural, top-down design, or a more object-oriented design as described in Chapter 12.
- Use meaningful variable names with good style (i.e., useCamelCase or use_underscores)
- Use docstring comments at the module, class, method, and function levels. Include preconditions and postconditions for all functions and class methods. For each function or method with a precondition, include code to check the precondition and raise an appropriate exception.

Implementation Suggestions:

- Don't procrastinate! Start the project early so if you have problems you can get help.
- Unit test the pieces of your program separately before integrating them. For example, if you have a Boolean function `isStraight(hand)` that takes a hand as a parameter and return True if the hand is a straight (otherwise return False), then test it individually before using it in your program.

Submit your homework electronically at https://www.cs.uni.edu/~schafer/submit/which_course.cgi
The steps for the homework submission system are:

1. Write, debug, and test your program. Save it in a file called **videoPoker.py**

2. Log on to the submission system at: **https://www.cs.uni.edu/~schafer/submit/which_course.cgi**

   (It is very likely that you will get some security certificate warnings when trying to use this. You may add an exception and accept the existing security certificate.) Use the same AD-ITS User name and password you use to log on the lab computers.

3. Select the course and section number of "810:052, Data Structures, Fienup". Click the "Continue".
4. Select the homework that you wish to submit: "HW 2: Video Poker". Click the "Continue" button.
5. Specify how many extra files you want to submit. Just leave it at 0. Click the "Continue" button.
6. Upload your program by Browsing and selecting your videoPoker.py file. Click the "Continue" button.
7. The next page reports on the status of the upload(s). You can always continue to upload a better version of the program until the deadline. The newer file will replace an older file of the same name.