

Due Sept. 17 (Friday) at 11:59 PM

**Objective:** To learn how documentation programs by writing APIs, using preconditions and postconditions, enforcing preconditions with exceptions, and using pydoc web-based documentation.

As your programs become more complex (larger in size and scope) and longer lasting (i.e., used for years in a production environment instead of just-run-once “toy” programs), you’ll need to pay closer attention to program:

- design - to aid in project development, future maintenance and code reuse
- documentation - to aid fellow developers and future maintainers in understanding and using your software components correctly and effectively
- testing - to aid in software reliability and robustness

**Part A.** Read sections 2.3.6, 2.6.4, 8.1.2, and 12.2 from the Lambert text. Download and unzip the hw3.zip file. This contains:

- a simple Die class (in the simple\_die.py module) for a six-sided die, and
  - an AdvancedDie class (in the module advanced\_die.py module) for a die which can be constructed with any number of sides. The AdvancedDie class inherits from the Die class.
- a) Document the Die and AdvancedDie class at the module, class, method, and function levels. Augment the existing comments to include appropriate preconditions and postconditions as necessary in the documentation of each method as discussed in subsection 12.2.3.

b) For testing certain dice games, suppose we want to extend the AdvancedDie class to include a new method `setRoll` which takes as a parameter a roll value that is used to set a die’s roll to a specified value. We might call this function as:

```
myDie.setRoll(3)    # sets myDie to a roll of 3
```

Implement a new subclass `MoreAdvancedDie` (in the module `more_advanced_die.py`) which inherits from the `AdvancedDie` class, and includes the `setRoll` method. Add any additional methods to the `MoreAdvancedDie` class that are needed to check the precondition(s) necessary for the `setRoll` method.

c) Modify the Die, AdvancedDie Enforce the preconditions by raising appropriate exceptions.

d) View the programmer-authored documentation for the `MoreAdvancedDie` class by typing `help(MoreAdvancedDie)` at the Idle shell prompt after importing the `MoreAdvancedDie` class.

e) Generate a web-page documentation for the `MoreAdvancedDie` class using pydoc as follows:

- Open a command prompt terminal window
- Change directories until you get to the directory containing you `MoreAdvancedDie` class
- Run the command (something like)  

```
c:\python25\lib\pydoc.py -w more_advanced_die > more_advanced_die
```

(Yours might have `pydoc.py` in a different location depending on the version and location of your Python install)
- Double-click on the `more_advanced_die.html` to view it in a web browser.

**Part B.** Read section 12.3 on testing. You are to develop a pyunit unit test for all the methods of the `MoreAdvancedDie` class developed in Part A. See section 12.3.5 and model your unit test after the `TestStudent` class on page 498.

**Submission of Homework:** Zip a directory containing all the Python programming files from Parts A, your `more_advanced_die.html` file, and your pyunit unit-test file from Part B. **Submit this .zip file containing your homework electronically at [http://www.cs.uni.edu/~schafer/submit/which\\_course.cgi](http://www.cs.uni.edu/~schafer/submit/which_course.cgi)**

The steps for the homework submission system are:

1. I'll assume your .zip file is called hw3.zip.
2. Log on to the submission system at: [http://www.cs.uni.edu/~schafer/submit/which\\_course.cgi](http://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" button.
4. Select the homework that you wish to submit: "HW 3: More Advanced Die". Click the "Continue" button.
5. Specify how many extra files you want to submit. Just leave it at 0 since you only have the .zip file to submit. Click the "Continue" button.
6. Upload your program by Browsing and selecting your hw3.zip 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.

(If you run into problems and worry that the submission was unsuccessful, you can email the hw3.zip file as an attachment to [fienup@cs.uni.edu](mailto:fienup@cs.uni.edu))