Introduction to Programming

Spring 2022

Functions

- Quick Review of Objects
- Example Program: Cannonball
- Defining New Classes
- Data Processing with Class
- **Objects and Encapsulation**
- Widgets
- Animated Cannonball

Encapsulating Useful Abstractions

- Defining new classes (like `MSDie`, `Projectile` and `Student`) can be a good way to modularize a program.
- Once some useful objects are identified, the implementation details of the algorithm can be moved into a suitable class definition.

Encapsulating Useful Abstractions

- The main program only has to worry about what objects can do, not about how they are implemented.
- In computer science, this separation of concerns is known as encapsulation.
- The implementation details of an object are encapsulated in the class definition, which insulates the rest of the program from having to deal with them.

Encapsulating Useful Abstractions

- One of the main reasons to use objects is to hide the internal complexities of the objects from the programs that use them.
- From outside the class, all interaction with an object can be done using the interface provided by its methods.

Encapsulating Useful Abstractions

- Advantage
  - Class can be updated and improved independently
  - No need to worry about “breaking” other parts of the program
  - Just do not change the interface
  - Public Interface
  - Private Implementation
  - Example
  - Car

Putting Classes in Modules

- Sometimes we may program a class that could be useful in many other programs.
  - Example – random, sqrt, sin, cos, etc.
- To reuse a class:
  - Put it in its own module file
  - Documentation to describe how the class can be used
  - No need to look at the code to figure out how to use it.

Module Documentation

- “#” to indicate comments
  - Comments explaining what’s going on in a Python file.
- Python also has a special kind of commenting convention called the docstring.
  - You can insert a plain string literal as the first line of a module, class, or function to document that component.

Module Documentation

- Why use a docstring?
  - Ordinary comments are ignored by Python
  - Docstrings are accessible in a special attribute called `__doc__`.
- Most Python library modules have extensive docstrings.
- Example, if you can’t remember how to use random
  - You can use `__doc__` attribute

Module Documentation

- Docstrings are also used by
  - The Python online help system and
  - By a utility called PyDoc that automatically builds documentation for Python modules
  - You could get the same information like this

Module Documentation

- To see the documentation for an entire module, try typing help(module_name)!

Module Documentation

- """" is a third way that Python allows string literals to be delimited, allowing us to type multi-line strings.
- Example

Working with Multiple Modules

- If you are interactively testing a multi-module Python program, you need to be aware that reloading a module may not behave as you expect.
- When Python first imports a given module, it creates a module object that contains all the things defined in the module (a namespace). If a module imports successfully (no syntax errors), subsequent imports do not reload the module. Even if the source code for the module has been changed, re-importing it into an interactive session will not load the updated version.

Working with Multiple Modules

- The easiest way – start a new interactive session for testing whenever any of the modules involved in your testing are modified. This way you’re guaranteed to get a more recent import of all the modules you’re using.
- If you’re using IDLE, you’ll notice it does this for you by doing a shell restart when you select “run module.”

Class Work

- Modify the student class (see Canvas for Code) by adding a mutator method that records a grade for the student as follows:
  - addGrade (self, gradePoint, credits)
- gradePoint is a float and represents a grade (4.0, 3.5, 3.0, etc)
- credits is a float representing number of credits for class

Class Work

- Use the updated class to implement a simple program for calculating GPA. Your program should create a new student object that has 0 credit and 0 quality points.
- Your program should then prompt the user to enter course information (grade point, and credits) for a series of course and then print out the final gpa achieved.