

NENG 685
PF 4

Fall 2017
Due Oct. 15, 2017

Name: Robert Torzill

On Pre-flights:

- If you work with anyone else, document what you worked on together.
- If you are not using python, then substitute your language of choice when Python is specified.

Do not write in the table to the right.

Problem	Points	Score
1	10	
2	5	
3	5	
4	5	
5	5	
6	5	
Total:	35	

1. (a) (4 points) Describe the difference between class and object. Give an example of a class and an object (in the object-oriented programming sense).
A cat is an object, its overall behavior is its class. If meaning to get in then not entering when allowed.

so the difference is akin to noun + verb, Objects are things
Classes are how objects behave

- (b) (3 points) Define the three key features of object oriented programming.

Encapsulation : owning data

Inheritance : establish a relationship hierarchy

Polymorphism : allow a model to customize its behavior even
if that behavior is based on another model

- (c) (1 point) T/F: All variables, data types, and functions in python are classes.

False, a variable can be an instance of class but
doesn't have to be

- (d) (1 point) T/F: It is good coding practice to call dunder methods explicitly.

False

- (e) (1 point) What python built-in is available to show the attributes of an object?

dir()

2. (5 points) Create a class definition to describe an element. Give it attributes of density, atomic number, and atomic mass. Give it a constructor that requires the atomic number to instantiate the class and a method to print out the state of the object.

Attached

3. (5 points) How can you overrule duck typing? Give your own example of why you might need to do so?

by using the `isinstance()` to check specific objects.

Security, by only allowing a specific type in and not relying
on python to get it right we eliminate certain edge cases.

For example an object shares a method name with a
different type of object and the wrong object gets called
we would get unexpected results.

4. (5 points) Create a class definition to describe an isotope. Have the class inherit from the element class. Give it attributes of number of neutrons, and half-life. Give it a constructor that requires the number of neutrons and half-life to instantiate the class. Give it a method that can print the decay constant (note ensure the method is functional):

$$\lambda = \frac{\ln(2)}{t_{1/2}} \quad \text{Attached}$$

5. (5 points) What are class decorators? Why might you want to use them?

a way to customize a class without ① rewriting it or ② creating a new one. Use for customization and reducing SLOC

6. (5 points) What is one concept that you found difficult in the reading?

Class decorators, the section does n't really go into if the decorator is added dynamically (which would be weird) or if you apply the tag then actually change the class later.