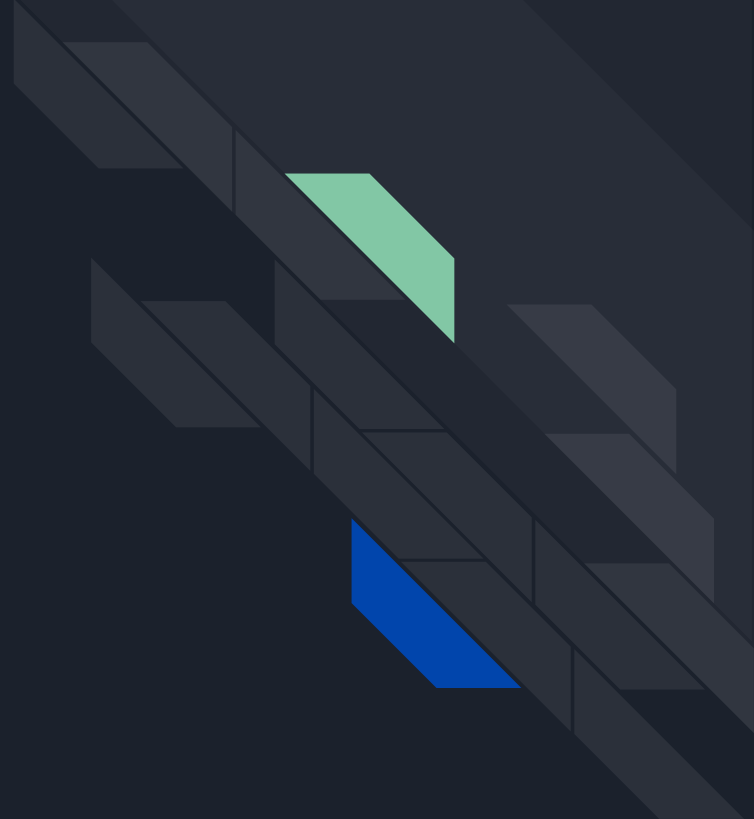




Lab 2: More Python Features

Data Structures and Algorithms for CL III
Nov 8 2019

Classes and Modules





What's a module?

- A Python module is any file containing Python code stored on the hard disk
- Generally, modules are intended to be imported (reused) but strictly any Python file is a module
- In Java, a file typically contains one class
 - In Python a module may contain zero, one or more classes
- To import a module stored in 'somefile.py' write:
 - `import somefile`
- You can also import individual functions, variables or classes with the following:
 - `from somefile import func`



What's a class?

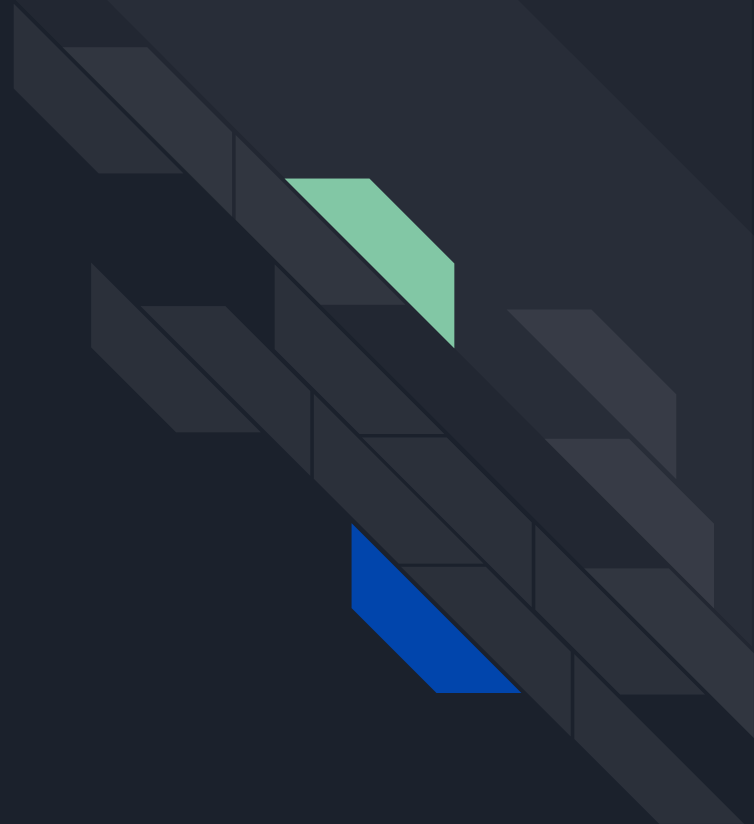
- Classes should be familiar conceptually from Java (templates for objects)
- Inheritance, class and instance variables work intuitively similarly in principle, but there are differences
- A Python function associated with a class is a method
- In Python, you can dynamically add new instance variables even if not defined in the class, as simply as:
 - `example_obj.special_new_var = 5`
- Another major difference is that you can inherit from multiple parents in Python
 - This can create a “diamond problem” of a class having conflicting members from two parents
 - Order of inheritance matters, which can lead to unintuitive results
 - In general, avoid this



Using classes

- To make a class called Student, write simply:
 - `class Student:`
- After the colon, indent the body of the class
- To make an empty class, you can write `pass` as the entire body
 - If you define functions in the class body they become methods, likewise class variables
- To instantiate a class, write:
 - `john = Student()`
- This calls the `__init__` method, if defined like so:
 - `def __init__(self, initvalue1, initvalue2):`
- `self` is a keyword similar to “this” in Java – it refers to the current instance, so you can set instance variables like so:
 - `self.name = initvalue1`

Debugging





Debugging tools

- Debugging can teach you about program flow and lets you suss out really nasty errors more easily
 - Also good for understanding pre-existing algorithms
- The core features of a debugging tool are **stepped execution**, **breakpoints** and **variable watches**
- Use **variable watches** to track how key variable values are updated
- Use stepped execution (one line at a time) to identify in detail where in a function something has gone wrong
- Use breakpoints with custom conditional statements to stop execution in very long or indefinitely executing loops
 - I.e. only check loop execution on long words in an array, if you have narrowed it down to a problem with the number of characters
- Debuggers help you be methodical, but you still have to narrow down faults yourself



Help getting started debugging in VS Code

- https://code.visualstudio.com/docs/python/python-tutorial#_configure-and-run-the-debugger