**Course**: Master Python 3: An intensive project based to python programming.

**Instructor**: By Doug Purcell

**Website**: [http://www.purcellconsult.com](http://www.purcellconsult.com/)

**Target audience**: Beginners to programming. No prerequisites except for basic mathematics and general computer know-how.

Instructor's office hours: TBD.

**Course objectives:** To take a beginner in python programming and help them build a concrete foundation of the language. This will be done through lectures, assignments, quizzes, mini coding projects, and a capstone project at the end of the course. The ultimate goal is to build the student’s confidence so that they can feel comfortable building solutions in python for companies.

**Course readings**: The course will mostly be taught from the [*Become a Python Developer*](https://www.amazon.com/Become-Python-Developer-Wrestle-Defeat/dp/0997326298) book. The book is available in e-book and paperback formats on Amazon.

The course’s power points along with the internet and online video sharing services like YouTube can serve as an additional learning aids.

**Suggested study hours**: This is an intensive boot camp style of course. The student will be expected to spend at least 20 hours a week studying, programming, and doing homework in order to solidify their programming skills. The student is expected to complete weekly assignments by their posted due date, but they can submit revisions afterwards.

### Grading:

This is not an accredited course but nevertheless there needs to be a way to track the student's progress to ensure that they are developing better software engineering skills. Here’s the proposition. A combination of *self-grading and* instructor grading will be utilized.

Solutions will be provided for written assignments and quizzes. A quiz may contain true/false, multiple choice, or short coding snippet questions. Correct marks should only be given for correct answers and no partial credit will be given in order to simplify the grading process. Students will grade their own written assignments/quizzes but should still send the instructor their grades.

This will help the instructor understand the issues the student is having. The written assignments/quizzes will use the traditional number grade system but there is not plus or minus. Therefore, here’s how the grading scale looks:

90-100: A

80-89: B

70-79: C

60-69: D

< 60: F

However, for coding projects, the grading scale will be *W*, WB, and *DW*. *W* denotes *work*s, *WB* denotes w*orks but*, and *DW* denotes *doesn’t work.* To clarify on *WB* this means that the program works but is written in a way that makes maintenance of the program difficult. This could include code that technically works but is not written in a clean, testable, maintainable fashion. Read [PEP 8](https://www.python.org/dev/peps/pep-0008) and [Writing clean, testable, high quality code in Python](https://developer.ibm.com/articles/au-cleancode) for more ideas on how to write quality python code. In order to get a certificate for completion the cumulative average for written assignments /quizzes must be at least a B, and all of the coding projects must have a W. Students can improve their scores by resubmitting their programs after the due date.

### Tentative three month schedule:

**Week One**: Python Numbers

Learn about the python number types, python syntax, basic operators, the input() function, the math module, and the print() function.

* Coding Project: Text based calculator.

**Week Two**: Control flow and iteration in python

Learn how to control the sequence in which statements are executed via control flow, and how to repeat segments in computer code using python’s looping constructs.

* Coding project: Number guessing game.
* Written project: Control flow and loops in python.

**Week Three**: Python’s Strings

Gain a deep knowledge of the string data type in python: learn how it’s stored in memory, it’s various operations, the multiple and sometimes tricky ways to format them, and its built in functions.

* Coding project 1: roses and red, violets are blue text based game.
* Coding project 2: Hello translator script.
* Written assignment: String manipulation.

**Week Four**: Python’s core data structures

Data structures help organize and manage data, the building block of programs. Learn about the four key types in python which are: lists, tuples, dictionaries, and sets.

* Coding project 1: Heads or Tails text based simulation game.
* Coding project 2: Student’s grade statistics.
* Written assignment: Python’s data structures.

**Week Five**: Functions and Functional programming

A complete guide to understanding functions in python 3.

* Coding assignment: An assortment of functions.

**Week Six**: Unit testing in python

Learn the different methodologies to testing and the various ways to write tests in python including unittest, pytest, and selenium.

* Coding project: Create a test suite.

**Week Seven**: Classes and object oriented programming in python.

Learn about classes, objects, methods, and the OOP approach behind python programming.

* Coding project: Bank Account simulator in python.

**Week Eight:** Decorators in python.

* Written assignment:

**Week Nine**: Advance python topics.

A curated list of some of the more ubiquitous advanced topics found in python.

* Written assignment: Advance python.

**Week Ten**: The Python standard library.

* Written assignment.

**Week Eleven**: Web scarpers in python I.

* Written assignment

**Week Twelve**:Web scraper in python II.

* Written assignment

**Capstone project**: A web scraper that solves a real world problem. This project will serves as advertisement for the mentee to gain work as a python developer. The code, documentation, and tests for the project needs to be uploaded to GitHub. The mentor and mentee should iron out details of the capstone project no later than week 3.