CSE-41310 Course Syllabus

Basic Information

Instructor: Bruce Schurter, MCSD
bruce.ucsd@gmail.com OR beschurter@ucsd.edu
Term: Summer 2020
Units: 3
Meeting Times: Online

Course Description

In this course, you will continue learning Python using advanced object-oriented programming techniques such as: encapsulation and abstraction, method and operator overloading, polymorphism and inheritance, and anonymous types. Other topics include: modules and packages, lambda expressions, decorators, parallelism, concurrency and advanced design patterns.

Goals and Objectives

The goal of this course is to learn how to develop Python programs using advanced object-oriented programming techniques and multithreaded algorithms. Some of the techniques you will learn:

- Advanced function design and functional programming
- Operator overloading
- Class inheritance
- Custom formatting
- Working with modules and packages
- Exception handling
- Metaprogramming
- Parallelism and Concurrency
- Advanced design patterns

Student/Course Requirement

To pass this course the student must complete 8 lab assignments and a final exam. Extra credit will be provided from time to time and is done so at the discretion of the instructor.

Course Materials

Python 3.7 (Anaconda distribution 2019.3 or later preferred)
Jupyter Notebook (included with Anaconda)
Visual Studio Code
PDF reader such as Adobe Reader
Learning Python – Mark Lutz (ISBN-13: 978-1449355739)

Additional material provided by instructor on Blackboard

Grading System

8 labs worth 100 points each (800 total)
1 final exam worth 200 points
Grading Percentage Scale: A: 90 – 100
B: 80 – 89
C: 70 – 79
D: 60 – 69
F: Below 60

Course Structure

Session 1 – Getting Started

This session is intended to ensure you have all the skills necessary from previous Python classes or other experience to continue with the more advanced features of Python.

- Setup
  - Anaconda, Bash, Jupyter Notebook and Visual Studio Code
- Basics Review
  - Variables and objects, expressions, statements, importing modules, running programs
  - Advanced number and string handling
  - Data type conversions
  - Control flow
    - Conditionals
    - Repetition structures

Reading Assignment

Quick Scan: Chapters 1 – 7, 12 – 13: Just ensure you are comfortable with all the topics

Homework

Lab 1 – Success with this lab ensures you are comfortable with the topics necessary for this level course

Session 2 – Modularity

- Functions
  - Parameters, default values, return values
  - Docstrings
  - Recursion
- Classes/objects
  - __init__
  - __str__ and __repr__
  - Properties

    - Operator Overloading
    - Inheritance
- Modules
    - __all__
- Packages
    - __init.py__

Reading Assignment

Chapter 16, 18, 19 (Recursive Functions), 22 – 25, 26 – 29

Homework

Lab 2

Session 3 – Exception Handling, Collection Types and Functional Programming

- Exception Handling
    - try/except/else
    - Custom exception types
    - Assertions
- Collection Types
    - List/Set/Dictionary
- Memory Concepts with Objects
- Lambdas
- Map/filter/reduce
- Comprehensions

Reading Assignment

Chapters 33-36, 4, 19, 20 (Comprehensions)

Homework

Lab 3

Session 4 – More Functional Programming and Metaprogramming

- Advanced Functional Programming
    - Iterators
    - Generators
    - First-class and Higher order functions
        - First-class: functions can be treated as values and passed to other functions
        - High-order: functions that act on or return other functions
    - Closures
    - Curry functions
- Metaprogramming
    - Function and Class Decorators
    - Function Annotations

- Metaclasses
- Descriptors

Reading Assignment

Chapters 19, 32, 38, 39, 40

Homework

Lab 4

Session 5 – Parallelism and Concurrency

- Context Managers
- concurrent.futures module
  - ProcessPoolExecutor (for CPU intensive tasks)
  - ThreadPoolExecutor (for network or I/O intensive tasks)
  - submit, done, and result methods
  - wait and as_completed functions

Reading Assignment

Chapter 34 (with/as Context Managers)

Online Reading

Homework

Lab 5

Session 6 – Parallelism and Concurrency continued

- multiprocessing module
  - queues
  - pipes
  - managers
- Synchronization Techniques
  - events
  - timers
  - locks
  - conditions
  - semaphores
- Coroutines and Asynchronous I/O vs. Parallel Processing
  - The asyncio Event Loop and the Coroutine Scheduler
  - asyncio Future Objects

Reading Assignment

Online Reading

Homework

Lab 6

Session 7 – Design Patterns

- Creational Design Patterns
  - *Revolves around object creation and instantiation*
  - The Factory Method
  - Prototype
  - Object Pool
  - Singleton
- Structural Design Patterns
  - *Deals with class and object composition, defining new ways to compose object to obtain new functionality*
  - Decorator
  - Proxy
  - Façade
  - Private Class Data
- Behavioral Design Patterns
  - *Behavioral Patterns involve communication between objects, how objects interact and fulfil a given task*
  - Chain of Responsibility
  - Observer/Observable

Reading Assignment

Online Reading

Homework

Lab 7

Session 8 – Working with Data

- Pandas
- Numpy
- Matplotlib

Reading Assignment

Online Reading

Homework

Lab 8

Blackboard Online Course Structure

The course is organized using the course menu (left side of your screen):

Announcements	This is the first page you see upon entering your course. Your instructor will post weekly announcements and reminders here.
Introduction	Contains an introduction to the course and instructor biography.
Syllabus	Contains the course outline, learning objectives, weekly assignments and course details.
Lessons	If it's a fully online course, this section will have the instructor's weekly audio/image lectures. The lectures are self-paced and can be replayed like a video movie (start, pause, rewind, etc.).
Discussion Board	Mostly for student collaboration with occasional instructor postings.
Assignments	Labs and the Final Project are located here.
Resources	Additional readings and handouts, web site links, and PowerPoint presentations are here.
Contacts	Instructor, student services and online learning support contact information is listed here.
Tools	Check your grades (My Grades), add a Homepage (Homepage), or access the Blackboard User Manual (User Manual) here.

Requirements

In order to satisfy course requirements, class participants must complete all course assignments on time (on or before the due date), and use graduate level writing/presentation for all written assignments.

IMPORTANT! Late assignments (anything posted or sent after the due date) will be graded -1 point for each day late unless due to a verifiable medical or family emergency. Assignments sent with the wrong naming convention or in the wrong format will be considered late until they are sent correctly. Late assignments will be accepted at the discretion of the instructor and cannot be accepted more than 1 week late. Grades are lowered for less-than-optimal (non-graduate level) grammar, spelling, and presentation. Make sure all references are correctly cited and follow APA or MLA guidelines.

In general, the performance criteria for an A grade for assignments is listed below: The assignment:

- Demonstrates a high level understanding of topics.
- Able to research topics independently to increase understanding.
- Code is well-commented and follows established programming guidelines.
- Does more than copy example code. A good programmer can take example code and apply those techniques to other problems
- Shows originality of thought.
- Uses proper citations for resources, especially for "borrowed" code found online.
- Has no punctuation, grammar, spelling errors. Style, formatting, and appearance add to quality of final product.

Expect and plan for contingencies and technical problems (they WILL happen!). All code will be submitted as a zip file through Blackboard unless otherwise notified by instructor.

Lab Assignments/Projects

Labs – 100 points each (800 total)

There are five labs assigned throughout the course which you must complete to demonstrate your proficiency of the given topic. The labs will consist of well-defined instructions that the student must follow to generate code that will facilitate the learning of the presented material. Much of the work will involve "hand-holding" where little deviation from the instructions will be necessary to accomplish the goals. However, each lab will also involve some amount of student-created code that must meet the stated requirements.

Final Exam – 200 points

The final exam is a comprehensive test that will test your ability to retain the knowledge gained from the course. The exam will draw from various sources including the lecture material/demonstrations, text book and lab assignments.

Grades

No late assignments or quizzes are accepted without explicit instructor approval.

Grades are based on percentage and the letter grades are given as follows (C or better to pass for P/NP):

A+	97-100
A	94-96
A-	90-93
B+	87-89
B	84-86

B-	80-83
C+	77-79
C	74-76
C-	70-73
D+	67-69
D	65-66
F	0-64

You may check your grade anytime by clicking **Course Tools** and then **My Grades**. This will show you the points you have earned so far in this course.

Grade Components

Labs:	80%
Final Exam:	20%
TOTAL	100%

About Discussion Board Participation

The use of the discussion board is entirely at the students' discretion. The instructor does not require its use nor will he/she regularly participate in discussions. If you have specific questions about the course that requires instructor intervention, please contact the instructor via email.

Please do not post partial/complete assignments on the discussion board as it would violate the "Academic Honesty Policy" mentioned below.

About Labs and the Final Exam

All assignments must be submitted via Blackboard as ZIP files containing the entire solution/project directory. To submit the files you must be in the specific lab/project page, not the Assignments overview page. To get to the specific lab/project page, click on the lab or project title on the Assignments page. If you cannot see a "Browse My Computer" button you are not on the correct page.

Labs are typically due by 11:59 PM Pacific Time exactly seven (7) days after availability. This corresponds to the time when the next lesson is available. The labs will still be available after the due date expires; however, this is for your reference only. It is the student's responsibility to ensure all work is completed and submitted in full by the date and time listed in Blackboard.

The final exam will be visible when the last lesson becomes available and is due at the same time as the final lab. Again, refer to Blackboard for the exact dates and times.

You may only attempt the labs and final exam once unless written approval granted by the instructor.

No student may submit code written entirely, or in part, by another student, whether past or present. Plagiarism violates the University's "Academic Honesty Policy" mentioned below.

Student Resources

On any Blackboard screen, there are tabs across the top and one is called the Student Tab. There is information on how to get started as a student and who to contact if you encounter any problems. There are also videos and written instructions on how to do some of the most common things in Blackboard.

Another one of these tabs is called FAQ (Frequently Asked Questions). If you click on the Students Category (on the left), you can find step-by-step directions for everything from sending email to uploading your assignments to posting a reply on the discussion board.

Code of Conduct

All participants in a course at UC San Diego Extension are bound by the ***University of California Code of Conduct***, found at http://www.ucop.edu/ucophome/coordrev/ucpolicies/aos/uc100.html.

Academic Honesty Policy

The University is an institution of learning, research, and scholarship predicated on the existence of an environment of honesty and integrity. As members of the academic community, faculty, students, and administrative officials share responsibility for maintaining this environment. It is essential that all members of the academic community subscribe to the ideal of academic honesty and integrity and accept individual responsibility for their work. Academic dishonesty is unacceptable and will not be tolerated at the University of California. Cheating, forgery, dishonest conduct, plagiarism, and collusion in dishonest activities erode the University's educational, research, and social roles.

If students who knowingly or intentionally conduct or help another student perform dishonest conduct, acts of cheating, or plagiarism will be subject to disciplinary action at the discretion of UC San Diego Extension.