CODEIT Python Cloud Software Engineering Course Syllabus

Course Description:
The goal of this course is to teach software engineering concepts through the Python Cloud Engineering language. In this course, you will learn fundamentals and advanced Python and AWS Cloud methodologies. The course will conclude with a dissertation project which will be a final project of your choosing for potential employers to look at and certification exams in AWS and Python. This course will go for 16 weeks.

Don't be afraid! You don't have to be a math wiz in any capacity to learn code!

When: Monday - Friday 4:00 pm - 8:00 pm

Instructors:

- shawnday.dl@gmail.com
- paulsun096@gmail.com
- elijahknsubuga@gmail.com

Contact Information: We will be communicating through Slack and occasionally email

What you will learn:

- Python programming language
- AWS Cloud Solutions
- Programmatic logical thinking

What is Python?

- Python is an interpreted high-level programming language for general-purpose programming created by Guido Van Rossum and was first released in 1991

There will be a weekly recap quiz's to reinforce new topics

There will be a number of projects that will be assigned as the course moves on to reinforce topics

Resources:

- http://github.com
- http://pythontutor.com
- http://python.org
- https://www.tutorialspoint.com/python/index.htm

PLEASE READ:

This course is a chance to make a serious career change (google python developer salary), however as with any programming course, the class alone will not be enough. If you really want to make this change, **you need to practice**! 30 minutes to an hour of review or even external learning will go a very long way and will make it easier for you towards the end of the course when your capstone is due. The more effort you put in, the better chances of you getting hired quickly!

Now onto the Syllabus:

Structure of Class:

2 hrs-Teaching, 2 hrs-In class coding

1 weekly quiz + assignment

Grading:	
Quizzes	10%
Assignments	20%
Midterm 1	20%
Midterm 2	20%
Capstone Project	30%

Week 1: Starting to code – Python and IDLE

- Installation and setup
- Rationale (why Python?)
    - libraries
    - show code snippets
    - companies using it
    - use cases
- thinking like a programmer
    - primitive expressions
        - ints, strings
    - combining expressions
        - arithmetic, concatenation
    - abstracting expressions

        - variables, (pure) functions
- assignments
    - temperature converter/bmi calculator

Week 2: Simple Programs

- bools
- if statements
- numbers
    - limitations of floats
    - use cases for modulo and floordiv
    - type conversion, rounding
    - math library
    - use cases for diff number formats (hex, octal, bin)
- how to read error messages
- reading documentation, googling
- assignment
    - cost of international calls
    - triangle classification

Week 3: Strings, lists, Dictionaries

- string cookbook
    - escapes
    - formatting
    - indices & slicing
- lists
    - use cases
    - comprehensions
    - matrices?
    - reference equality gotchas
- dictionaries
    - use cases
    - hashable req
- common patterns
    - `in` and `not in` operators
    - when to pick which data structure
    - common methods & funcs
        - `len`, `sorted`
    - `[::-1]`
- `for`-loops
    - range
    - the iterable abstraction
- assignments
    - custom `max`/how to represent a phonebook/password strength calculator/fizzbuzz

Week 4: OOP

- basics of OOP
    - motivation
- instances/objects
- Encapsulation
- Inheritance
- Polymorphism

- assignments
    - game entities
    - points, rational numbers
    - extending `list`

Week 5: Bringing it all together

- Exception handling
    - bugs vs expected errors
- file IO
- Debugging and testing
- introspection with `dir`
- Modules (random, datetime, os, sys)
- lambdas
- code style
- assignments
    - parsing .csv
    - using pathlib

Week 6: Development Environments

- upgrade from idle
    - editor
        - pycharm, vscode
    - terminal
    - debugger, linter, language server protocol
- virtual environments
- git
    - and github
    - project structure
- pip & what you can get
    - requests, django/flask, numpy, pandas, tensorflow
- basic terminal interaction
- assignments
    - personalize your dev environment

Week 7: Automation

- APIs
    - hiding secret keys in environment variables
- twitter bot project (tweepy)
- pwnedpasswords checker
    - requests, hashlib
- assignments
    - use {twilio, smtp} to send an {sms, email}

Week 8: Web scraping

- Scraping basics
    - use cases, ethics
    - json overview
    - how to inspect sites

- beautifulsoup
    - overview
    - selectors
- scraping hacker news project
- assignments
    - find top movies on IMDB since 1980

Week 9: Web Fundamentals

- how the web works
- building a {blog, portfolio} website
    - django/flask
    - html/css templates
    - database integration
- assignment
    - customize the page we build in class

Week 10: Data science fundamentals

- Jupyter notebooks
- Kaggle datasets
- Data wrangling with pandas
- Visualization w/ seaborn & bokeh
- brief introduction to machine learning
    - use cases
    - scikit-learn
    - simple predictions using iris dataset
- Assignment
    - scrape a site for data to wrangle

Week 11: Algorithms

- Recursion
- Search & sort algorithms & their complexities
    - {linear, binary} search
    - {insertion, merge, quick}sort
- fibonacci impls
- towers of hanoi, pascal's triangle
- assignment
    - walk through an explanation of {quicksort?} and why it works

Week 12: Data structures

- Recognizing use cases and complexities of:
    - Arrays
    - Stacks
    - Queues
    - Hash tables
- Knowing how to implement:
    - {Singly, Doubly} linked list
    - binary tree
    - graphs
        - DFS and BFS algorithms
- assignment
    - check if a string of parentheses "()))((())())" is balanced or not (use a stack)

Week 13: DevOps

- Cloud instances (AWS)
- Dockerizing projects
    - using docker-compose to spin up a database image

- MySQL / PostgreSQL
- deploying to a domain
- kubernetes scaling
- assignment
    - deploy a “hello world” app for production, with all that entails

Week 14: Capstone Project

- Entire week devoted to capstone project
- Requirements:
    - Project must have backend/frontend and serve a purpose
    - Code copied from the internet with no referencing will get a 0
    - Entire projects copied from the internet get a 0
    - Any code used from another source must be referenced with a link to the website
    - Code that does not work will get a 0
    - The student is expected to understand all their code (even that which was copied) and be able to explain it
    - Capstone is due Friday before class
    - Presentations 10-20 minutes on Friday