Duke TIP Summer Studies 2017

Applications, Algorithms, Computers: Modern Programming
Georgia Tech – TERM 2
Instructor: Nitza A. Santiago-Figueroa
TA: Ecclesia Morain

Required Texts:

- Python Crash Course: A Hands-On, Project Based Introduction to Programming

Optional Book:

- *Introduction to Algorithms: Ed. 3* Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein

Course Description:

How does Google instantly find the relevant websites when you type a word into the search bar? How does a city know the most efficient route that their trash collectors should take? Explore the ways that computer scientists develop algorithms, processes, and programs that allow complicated problems to be addressed in meaningful ways. Go beyond simply understanding how to create computer programs and understand the mathematics that drive the adaptation of these programs to many fields of study, including medical sciences, aerospace, business and physical sciences. Some computer programming experience would be beneficial.

Course Objectives/Learning Goals:

- Understand and represent a problem and apply problem solving techniques
- Program effectively in Python, Java, Processing
- Understand and apply computational techniques and concepts like Search Trees, Graphs, Set Theory, Recursion
- Learn how to represent knowledge for a computer to understand

Evaluations:

- Project Presentations
- Research Topic Presentations
- Individual Project Work

Expectations of students:

- Punctually attend the classes and submit assignments.
- No Plagiarism allowed!

(Collaboration and discussions are encouraged for problem solving but "You must submit your own work")

- Take initiatives and participate actively in classroom activities.
- Respect the thoughts and ideas of other students and maintain discipline in the classroom.
- Last but not the least, have curiosity and ask questions!

References:

https://www.learnpython.org/
http://www.cc.gatech.edu/classes/AY2015/cs8803_spring/mainContent/videos.html
Daniel Shiffman – Nature of Code: http://natureofcode.com/book/chapter-1-vectors/

Week 1

Monday – Course Introduction and Overview:

Morning:

- Course Introduction
- Review of Syllabus
- Student Programming Experience Survey
- Course Introduction: Lecture on Programming and Applications
    - What is programming?
    - Various Applications of Programming

Afternoon:

- Introduction to Python
- Computer setup:
    - Python 2.7 or 3.6
- How to Navigate Windows directory via command line tools:
    - Duke Virtual Lab
    - Linux vs OS X vs Windows
- Basic Programming Topics:
    - Intro to Python
    - Hello World!
    - Types, Variables, Operators in Python
    - Conditional statements
    - Loops, Arrays – For, While
    - Scope
    - Methods/Functions in Python
- Practice problems – Python

Evening:

- Practice problems – Python

Tuesday – *Some Programming Groundwork*:

Morning:

- Basics of number systems (hex,decimal,binary)
- Exercises on decimal subtraction, addition, conversion
- Basics of propositional logic
    - Introduction to set theory
    - Subsets, Power sets, Universal set, Null set
    - Set operations – Union, Intersection, Complement
    - Bit manipulation in python
    - Look at some interesting algo problems – power set problem

Afternoon:

- Python Introductory assignment:
    - List Comprehensions and Conditionals
- Practice problems
- Evening:

  Plan T-Shirt Design with Class

Wednesday –Hangman Project:

Morning:

- Practice Problems:
- Spent most of the on introductory project
- People who worked ahead began the hangman project in the afternoon
- Introductory programming project:
    - Hangman begin

Afternoon:

- Introductory programming project:
    - Hangman finish

Evening:

Worked on Hangman

Thursday – *Review of Programming Techniques in Python:*

Morning:

- Finish up introductory exercises and finish up the hangman project
- Go over while loops, nested for loops (loops in general)
- Strings, Lists (looping over)
- booleans, integers, floats, tuples
- Go over list comprehension solutions (got it)
- Go over multi-dimensional lists
- Go over dictionaries
- List-Comprehensions
- The usefulness of print statements

Afternoon:

- Begin Hangman Project

Friday – Finish *OOP* Project:

Morning:

- Introduction to Object Oriented Programming:
    - Classes
    - Objects
    - Inheritance basics
    - Use of public methods/functions/getters and setters
    - Implement a simple vehicle class on the board
- Object Oriented Programming Exercises
- Multidimensional Arrays
- Tuples
- Nested Loops
    - Use of Booleans
    - Use of nested loops

Afternoon:

- Start implementation of path network connections in games for use in path-finding

Evening:

Worked on Path Network project

Saturday – Finish project and movie:

- Finished Path-Network Project
- Watched The Imitation Game

Week 2:

Monday – Introduction to some Data Structures, Algorithm Analysis Basics:

Morning:

- Work on new data structures – Stacks and Queues
- Data Structures- Stacks, Queues overview
- Implementation problems for stacks and queues
    - Worksheet on Stacks and Queues

Afternoon:

- Finish Worksheet on Stacks and Queues

Tuesday – Graphs, Intro to pathfinding with graphs:

Morning:

- Introduction to recursion
- Base Cases and recursion stack
- Thinking recursively
- Recursion problems on the board
- Recursion/dictionary worksheet begin

Afternoon:

- Recursion and Search worksheet finish

Weds – PathFinding:

Morning:

- Introduction to graph algorithms:
    - Breadth First Search
    - Depth First Search
    - Review Recursion and Search Worksheet
    - Introduction of A* Shortest Path Algorithm

Afternoon:

- Begin Pacman programming project for pathfinding - (Stanford and Berkeley Project)

Thursday – Pacman Project:

Morning:

- Work on Pacman programming project
- Path finding break:

Afternoon:

- Continue Pacman programming project

Friday – Search, A*, Beyond classical search:

Morning:

- Data Structures: Trees, Binary Trees, BST, Node Class
- Complete Pacman Programming Project
- Begin making project demo and presentation.

Afternoon:

- Students Demo and present projects and implementation method.

Saturday –Programming Break Day:

- Robo Rally pathfinding game with robotics class

Week 3

Monday – Web development introduction:

Morning:

- A look at basic HTML
- A look at using style sheets/CSS
- Students begin individual website building

Afternoon:

- Students work on individual website building

Tuesday – Continue Web development and 2D game development:

Morning:

- Students work on individual website building OR students make a
- Python – make 2D game using book

Afternoon:

- Continue work on personal Project

Wednesday- Personal Project:

Morning:

- Students work on individual website building or work on 2D game

Afternoon:

- Wrap up personal Project

Thursday –Ethics in Computing:

Morning:

- Ethics Discussion: Autonomy and AI- Where does The Responsibility Lie.
- Awesome Machine Learning vid: http://www.ted.com/talks/blaise_aguera_y_arcas_how_computers_are_learning_to_be_creative

Afternoon:

- Students Demo and Present Websites/2D game

Friday – Presentations and Coding Challenges:

Morning:

- Finish student presentations.

Afternoon:

- Coding Challenge Riddles and Practice Problems

Saturday-Talk to Parents:

- Parent-teacher conferences