Master Course Syllabus for EE 241 (ABET sheet)

Title: PYTHON FOR SIGNAL AND INFORMATION PROCESSING

Credits: 2

https://www.washington.edu/students/crscat/ee.html#ee241

UW Course Catalog Description

Introduction to Python programming for signal and information processing. Basic Python syntax and data types. Packages for data manipulation and visualization. Handling a variety of data formats. Prerequisite: either CSE 142, CSE 143, or CSE 163. Offered: Autumn/Winter/Spring

Coordinators: M. Ostendorf, S. Makhsous, Electrical and Computer Engineering

(Team) Faculty who have or are willing to teach this core course):
Blake Hannaford, Akshay Gadre, Mahmood Hameed, Nathan Kutz, Sep Makhsous, Mari Ostendorf

Goals: To learn computational tools for working with a variety of signals and other information sources to support data analysis and implementation of algorithms for electrical engineering applications.

Learning Objectives: At the end of this course, students will be able to:

1. *Implement* simple programs in Python using Jupyter Notebook;
2. *Read and write* data in different file formats addressing multiple applications;
3. *Understand* how to efficiently work with a variety of data types;
4. *Utilize* standard packages for data processing, such as SciPy, NumPy, pandas, etc.;
5. *Utilize* functions to plot, play, or visually render different signals or information sources.

Textbook:
1. Think Python. Freely available online in HTM L and PD F
2. The Python Tutorial, available from the Python website

Prerequisite courses:
CSE 142, CSE 143, or CSE 163

Prerequisites by Topic:
1. Computer Programming

Lecture Topics:
1. Getting started with Python and Jupyter notebook, installing packages, working with variables, lists, tuples, and arrays (1 week)
2. Audio file I/O and plotting, functions, conditional control, exception handling (2 weeks)
3. Image file I/O and rendering, matrix operations (2 weeks)
4. Application using csv files, pandas, data frames (2 weeks)
5. Testing and debugging (1 week) 6. Application using text or graphs (2 weeks)

Laboratory Topics:

- Python concepts, syntax, and packages are introduced in the context of lab assignments built on four different data sources.

Note: Labs are unlikely to always directly follow the content presented in class, but content for each lab is covered prior to students beginning the lab, even if the lab lags 1-2 weeks behind the lecture material.

Course Structure: The class meets once a week for a 2-hour computer lab section. Outside of the two-hour lab section, students spend an additional hour per week to view assigned online tutorials and videos and 3 hours on average to meet with the instructor or TAs and complete the weekly lab assignments.

Computer Resources: The course uses Python for the laboratory exercises and also for checking homework problems. Students are expected to use their personal computers.

Laboratory Resources: (see Computer Resources)

Grading: Formative assignments include homeworks, in-class exercises, and (in part) laboratory reports. Summative assignments include midterms and final exams. The course syllabus and grading rubrics will clearly identify the goal of each assignment (formative or summative).

- Homework/After Class Exercises/In-Class Exercises: 30%
- Laboratory Reports: 20%
- Exams: 50% (2 exams + final)

ABET Student Outcome Coverage: This course addresses the following outcomes:
H = high relevance, M = medium relevance, L = low relevance to course.

(5) Teams: Ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives. (L) Some computer lab work is conducted in teams.

(6) Experiment: An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions. (M) All labs involve data analysis and visualization.

(7) Learning: An ability to acquire and apply new knowledge as needed, using appropriate learning strategies. (M) Students are expected to use online documentation to learn the Python programming language for use in lab exercises, building on their knowledge of programming in other languages.

Prepared By: Mari Ostendorf, Sep Makhsous
Last Revised: March 7, 2024

Additional information and resources regarding teaching ECE courses (e.g., links to course repositories for materials from previous course offerings; guidelines for using AI tools in courses; syllabus language for course accommodations, etc.) can be found on the UW ECE Intranet:
https://peden.ece.uw.edu/academic-ops/