Python II - Object-Oriented Programming

Course Overview

This intermediate course dives into object-oriented programming (OOP) concepts, GUI development, and modular programming with Python. Students will learn about recursion, generators, decorators, and third-party libraries, culminating in a project where students develop a GUI-based chess game using PyGame.

Course Objectives

- Apply object-oriented programming principles including classes, encapsulation, inheritance, polymorphism, and abstraction.
- Learn best practices for software development, including code organization, documentation, and version control.
- Learn new tools to write more efficient and modular Python code.
- Implement modular programming techniques using Python's third-party libraries.
- Develop graphical user interfaces (GUIs) with Python, focusing on event-driven programming and user interaction.
- Complete a comprehensive project to design and implement a chess game using PyGame, demonstrating mastery of OOP and GUI programming.

Unit 1: Objects and Classes

Introduces OOP with Python, covering classes, inheritance, and polymorphism.

- ☐ Read - What is object-oriented programming?
- ☐ Video - Python Classes in 1 Minute!
- ☐ Tech Class Repo (clone this repository)
    - ☐ Within the Python-II folder you'll find most of the readings and exercises for this course. For each section, review the reading and then complete the exercise on your computer.
- ☐ Reading & Exercise 1.1 - Classes and Objects
- ☐ Reading & Exercise 1.2 - Encapsulation
- ☐ Reading & Exercise 1.3 - Inheritance
- ☐ Reading & Exercise 1.4 - Polymorphism
- ☐ Reading & Exercise 1.5 - Abstraction
- ☐ Reading & Exercise 1.6 - Operator Overloading

Unit 2: Advanced Programming Techniques

Focuses on modular programming techniques and introduces basic graphics with Python in preparation for learning PyGame.

- ☐ Reading & Exercise 2.1 - Recursion
- ☐ Video - Recursion for Python Beginners with Recursive Function Examples
- ☐ Reading & Exercise 2.2 - Scope
- ☐ Reading & Exercise 2.3 - Revisiting Modules
- ☐ Sidenote - DRY (Don't Repeat Yourself)
- ☐ Sidenote - Refactoring Strategies
- ☐ Reading & Exercise 2.4 - Turtle Graphics

Unit 3: PyGame and Version Control Practice

Teaches game development with PyGame, culminating in a chess game project.

- ☐ Video - 13 Advanced (but useful) Git Techniques and Shortcuts
- ☐ Reading & Exercise 3.1 - Intro to PyGame
- ☐ Reading & Exercise 3.2 - Drawing Shapes in PyGame
- ☐ Reading & Exercise 3.3 - Handling User Input in PyGame
- ☐ Reading & Exercise 3.4 - Text and Sound Effects in Pygame
- ☐ Slideshow - What Is Version Control?
- ☐ Set up a new Github repository and share it with me. You can use it to work on assignments as well as personal projects.
- ☐ Final project - Chess Game
    - ☐ Overview of the chess game: rules, board layout, gameplay
    - ☐ Design and implementation of the GUI-based chess game using PyGame and object-oriented programming
    - ☐ Testing and debugging the game
    - ☐ Best Practices for Git Commit Messages as you're working on it