CSC 221 Advanced Python Programming

COURSE DESCRIPTION:

Prerequisites: CSC 121
Corequisites: None

This course introduces advanced computer programming using the Python programming language. Emphasis is placed on the advanced programming concepts including advanced algorithms and programming principles utilizing standard and third-party library tools. Upon completion, students should be able to design, code, test, and debug advanced Python language programs.
Course Hours per Week: Class, 2. Lab, 3. Semester Hours Credit, 3.

LEARNING OUTCOMES:

Upon completing requirements for this course, the student will be able to:

1. Apply advanced Python programming concepts and principles to develop software solutions.
2. Use standard and third-party Python libraries to implement advanced algorithms and programming tasks.
3. Design, code, test, and debug advanced Python programs that meet defined requirements.

OUTLINE OF INSTRUCTION:

I. Review of Python
   - A. Fundamentals
     - i. Data types, variables, expressions, and input/output
     - ii. Loops and iteration
     - iii. Selection statements and Boolean logic
     - iv. Strings, text files, and file operations
     - v. Lists, dictionaries, and data manipulation
   - B. Functions and Recursion
     - i. Function definitions, parameters, arguments, and return values
     - ii. Scope, modular program design, and reusable functions
     - iii. Recursive functions, base cases, and recursive problem solving

II. Graphics and Image Processing
   - A. Drawing with turtle graphics and coordinate systems
   - B. Shapes, colors, and graphical transformations
   - C. Image representation, pixel manipulation, and image operations
   - D. Creating and modifying images and collages

III. Graphical User Interfaces
   - A. GUI components, windows, labels, buttons, and input controls
   - B. Event-driven programming and callback functions

C. GUI layouts, user input, and output
D. Testing and debugging interactive applications

IV. Classes
A. Object-oriented programming concepts and class design
B. Attributes, methods, constructors, and object instances
C. Encapsulation and interactions among objects
D. Applying classes to an interactive simulation or game

V. Data Analysis and Visualization
A. Importing, organizing, and processing data
B. Sorting, filtering, and summarizing datasets
C. Data visualization with Python libraries
D. Interpreting graphical representations of data

VI. Multithreading, Networks, Client/Server Programming
A. Concurrent execution and multithreading concepts
B. Creating and coordinating threads
C. Network communication fundamentals
D. Client/server architecture and socket communication
E. Developing and testing networked applications

VII. Searching, Sorting, and Complexity Analysis
A. Sequential and binary searching
B. Fundamental sorting algorithms and their behavior
C. Algorithm efficiency and time complexity
D. Comparing approaches to searching and sorting
E. Implementing and testing searching and sorting solutions

REQUIRED TEXTBOOK AND MATERIAL:

The textbook and other instructional material will be determined by the instructor.