Software Development Using Python 2023-2024 Catal

[ARCHIVED CATALOG]

SDEV 220 - Software Development Using Python

PREREQUISITES: Demonstrated readiness in STEM MATH - Route 1 AND (SDEV 140 - Introduction to Software Development OR CSCI 101 - Computer Science I)
PROGRAM: Software Development
CREDIT HOURS MIN: 3
LECTURE HOURS MIN: 2
LAB HOURS MIN: 2
DATE OF LAST REVISION: Fall, 2020

Provides an understanding of fundamental and advanced concepts of the Python programming language. The emphasis will be on creating industry standard programs using current programming design software. Students will learn basic programming concepts such as sequence, iteration and decision structures; variables and constants; and functions and advanced concepts such as searches, sorts, collections, dictionaries, arrays, and linked lists. Students will learn the concepts of object-oriented programming using classes, inheritance and polymorphism. These skills will be practiced in a hands-on environment. Students will practice skills such as team building, work ethic, communication, documentation, and adaptability.

MAJOR COURSE LEARNING OBJECTIVES: Upon successful completion of this course the student will be expected to:

1. Create programs using the basic structure of the Python language including variables, constants and character strings, arithmetic operators, expressions and control statements.
2. Create programs using Python-based classes, objects, encapsulation, polymorphism, inheritance, and packages.
3. Develop modular, searching, sorting, recursive, and Lambda functions in programs.
4. Construct lists, tuples, dictionaries and sets in programs.
5. Perform complexity analysis in programs.
6. Evaluate the importance of using tools for design, documentation and testing.
7. Demonstrate how to utilize collections, arrays, and linked structures such as stacks, queues, and lists.
8. Evaluate, test and debug Python programs.
9. Create Graphical User Interface (GUI) utilizing event driven programming.
10. Implement stacks, queues, and binary search trees.
11. Design and demonstrate network-based applications with API.
12. Investigate security vulnerabilities in programs and implement secure programming mitigation.

COURSE CONTENT: Topical areas of study include -

- Variables
- Constants
- Character Strings
- Arithmetic operators
- Expressions and statements
- Repetition techniques
- Lists

- Access data files
- Dictionaries
- Class and Object definitions
- Multithreading
- Recursion
- Tools for design, documentation and testing
- Data structures
- Design and Code
- Edit and Compile
- Collections
- GUIs
- Recursion
- Stack, queues, lists
- Sets and Dictionaries
- Search and sort methods
- Big O
- Theta
- NoSQL
- Parsing
- JSON
- XML/YAML

Course Addendum - Syllabus (Click to expand)
