COWLEY COLLEGE
& Area Vocational Technical School

COURSE PROCEDURE FOR

PYTHON PROGRAMMING
CIS1872 3 Credit Hours

Student Level:
This course is open to students on the college level in either the freshman or sophomore year.

Catalog Description of the Course:
CIS1872 - PYTHON PROGRAMMING (3 hrs)
An introductory course to give computer science majors an introduction to programming in Python. Webpages will be enhanced with CGI programs written in Python. The basics of HTML will be covered. Various problems will be solved using Python.

Prerequisites:
None

Controlling Purpose:
This course is offered to teach programming skills in Python to students. The basic constructs learned in this course will apply to any programming language (with slight modifications).

Learner Outcomes:
Upon completion of this course, the student should be able to write Windows programs using Python. The student should be able to use input/output statements, selection statements, repetition structures, procedures, and arrays. The student will learn how to incorporate exception handling into their programs. The student will learn how to use Python to enhance their web pages. Lastly, they will know how to write to and retrieve data from various sources.

Units Outcomes and Criterion Based Evaluation Key for Core Content:
The following defines the minimum core content not including the final examination period. Instructors may add other content as time allows.

Evaluation Key:

A = All major and minor goals have been achieved and the achievement level is considerably above the minimum required for doing more advanced work in the same field.

B = All major goals have been achieved, but the student has failed to achieve some of the less important goals. However, the student has progressed to the point where the goals of work at the next level can be easily achieved.

C = All major goals have been achieved, but many of the minor goals have not been achieved. In this grade range, the minimum level of proficiency represents a person who has achieved the major goals to the minimum amount of preparation necessary for taking more advanced work in the same field, but without any major handicap of inadequacy in his background.

D = A few of the major goals have been achieved, but the student's achievement is so limited that he is not well prepared to work at a more advanced level in the same field.
F = Failing, will be computed in GPA and hours attempted.
N = No instruction or training in this area.

CHAPTER 1: Introduction to Computers, Internet and World Wide Web Outcomes: Upon completion of this unit, the student will understand what computers are, how they work and how they are programmed. Also the student will understand the evolution of Python.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand basic computer concepts.
						Become familiar with different types of programming languages.
						Become familiar with the history of the Python programming language.

CHAPTER 2: Introduction to Python Programming Outcomes: Upon completion of this unit, the student will understand the Python program development environment. They will write simple programs that incorporate input/output statements, decision-making statements, and that use arithmetic operators.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand a typical Python program-development environment.
						Write simple computer programs in Python.
						Use simple input and output statements.
						Become familiar with fundamental data types.
						Use arithmetic operators.
						Understand the precedence of arithmetic operators.
						Write simple decision-making statements.

CHAPTER 3: Control Structures Outcomes: Upon completion of this unit, the student will be able to write programs that incorporate selection structures, repetition structures, and program control statements.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand basic problem-solving techniques.
						Be able to develop algorithms through the process of top-down, stepwise refinement.
						Be able to use the if, if/else and if/elif/else selection structures to choose among alternative actions.
						Be able to use the while and for repetition structures to execute statements in a program repeatedly.
						Understand counter-controlled repetition and sentinel-controlled repetition.
						Use augmented assignment symbols and logical operators.
						Use the break and continue program control statements.

CHAPTER 4: Functions Outcomes: Upon completion of this unit, the student will be able to write programs that use functions in various ways.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand how to construct programs modularly from small pieces called functions.
						Create new functions.
						Understand the mechanism of exchanging information between functions.
						Introduce simulation techniques using random number generation.
						Understand how the visibility of identifiers is limited to specific regions of programs.
						Understand how to write and use recursive functions, i.e., functions that call themselves.
						Introduce default and keyword arguments.

CHAPTER 5: Lists, Tuples and Dictionaries Outcomes: Upon completion of this unit, the student will be able to write programs that will have data-handling capabilities using data structures. Specifically they will understand how to use lists, tuples, and dictionaries.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand Python sequences.
						Understand the list, tuple and dictionary data types.
						Understand how to create, initialize and refer to individual elements of lists, tuples and dictionaries.
						Understand the use of lists to sort and search sequences of values.
						Be able to pass lists to functions.
						Introduce list and dictionary methods.
						Create and manipulate multiple-subscript lists and tuples.

CHAPTER 6: Introduction to the Common Gateway Interface (CGI) Outcomes: Upon completion of this unit, the student will understand how CGI works and how Python programs can be integrated with web pages.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand the Common Gateway Interface (CGI) protocol.
						Understand the Hypertext Transfer Protocol (HTTP).
						Implement CGI scripts
						Use XHTML forms to send information to CGI scripts.
						Understand and parse query strings.
						Use module cgi to process information from XHTML forms.

CHAPTER 7: Object-Based Programming Outcomes: Upon completion of this unit, the student will understand the basic concepts of object-based programming and how they can be incorporated into Python.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand the software-engineering concepts of "encapsulation" and "data hiding".
						Understand the notions of data abstraction and abstract data types (ADTs).
						Create Python ADTs, namely classes.
						Understand how to create, use and destroy objects of a class.
						Control access to object attributes and methods.
						Begin to appreciate the value of object orientation.

CHAPTER 8: Customizing Classes Outcomes: Upon completion of this unit, the student will be able to utilize classes within their Python programs.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand how to write special methods that customize a class.
						Be able to represent an object as a string.
						Use special methods to customize attribute access.
						Understand how to redefine (overload) operators to work with new classes.
						Learn when to, and when not to, overload operators.
						Learn how to overload sequence operations.
						Learn how to overload mapping operations.
						Study interesting, customized classes.

CHAPTER 9: Object-Oriented Programming: Inheritance Outcomes: Upon completion of this unit, the student will be able to incorporate inheritance into their Python programs.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Create new classes by inheriting from existing classes.
						Understand how inheritance promotes software reusability.
						Understand the notions of base class and derived class.
						Understand the concept of polymorphism.
						Learn about classes that inherit from base-class object.

CHAPTER 10: Graphical User Interface Components: Part 1 Outcomes: Upon completion of this unit, the student will be able to write programs that incorporate basic GUI interfaces. Also they will understand how to work with layout managers.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand the design principles of graphical user interfaces.
						Use the Tkinter module to build graphical user interfaces.
						Create and manipulate labels, text fields, buttons, check boxes and radio buttons.
						Learn to use mouse events and keyboard events.
						Understand and use layout managers.

CHAPTER 11: Graphical User Interface Components: Part 2 Outcomes: Upon completion of this unit, the student will be able to write programs that use more advanced GUI interfaces.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Create a scrolled list of items from which a user can make a selection.
						Create scrolled text areas.
						Create menus and popup menus.
						Create and manipulate canvases and scales.

CHAPTER 12: Exception Handling Outcomes: Upon completion of this unit, the student will understand how to use exception handling to make their programs more robust and dependable.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand exceptions and error handling.
						Use the try statement to delimit code in which exceptions may occur.
						Be able to raise exceptions.
						Use except clauses to specify exception handlers.
						Use the finally clause to release resources.
						Understand the Python exception class hierarchy.
						Understand Python's traceback mechanism.
						Create programmer-defined exceptions.

CHAPTER 13: String Manipulation and Regular Expressions Outcomes: Upon completion of this unit, the student will be able to write programs that manipulate strings in various ways. Also the student will understand how to use regular expressions.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand text processing in Python.
						Use Python's string data-type methods.
						Manipulate and search string contents.
						Understand and create regular expressions.
						Use regular expressions to math patterns in strings.
						Use metacharacters, special sequences and grouping to create complex regular expressions.

CHAPTER 14: File Processing and Serialization Outcomes: Upon completion of this unit, the student will be able to write programs that access files.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Create, read, write and update files.
						Become familiar with sequential-access file processing.
						To understand random-access file processing via module shelve.
						Specify high-performance, unformatted I/O operations.
						Understand the differences between formatted and raw data-file processing.
						Build a transaction-processing program with random-access file processing.
						Serialize complex objects for storage.

CHAPTER 17: Database Application Programming Interface (DB-API) Outcomes: Upon completion of this unit, the student will be able to write programs that access databases in various ways including querying, inserting, and updating.						
A	B	C	D	F	N	Specific Competencies Demonstrate the ability to:
						Understand the relational database model.
						Understand basic database queries using Structured Query Language (SQL).
						Use the methods of the MySQLdb module to query a database, insert data into a database and update data in a database.

Projects Required:
Projects will vary according to the instructor.

Text Book:
Contact Bookstore for current textbook information.

Materials/Equipment Required:
On-line Delivery: Python Programming Language (free download off of the internet)

Attendance Policy:
Students should adhere to the attendance policy outlined by the instructor in the course syllabus.

Grading Policy:
The grading policy will be outlined by the instructor in the course syllabus.

Maximum Class Size:
Based on current occupancy.

Course Time Frame:
The U.S. Department of Education, Higher Learning Commission and the Kansas Board of Regents define credit hour and have specific regulations that the college must follow when developing, teaching and assessing the educational aspects of the college. A credit hour is an amount of work represented in intended learning outcomes and verified by evidence of student achievement that is an institutionally-established equivalency that reasonably approximates not less than one hour of classroom or direct faculty instruction and a minimum of two hours of out-of-class student work for approximately fifteen weeks for one semester hour of credit or an equivalent amount of work over a different amount of time, The number of semester hours of credit allowed for each distance education or blended hybrid courses shall be assigned by the college based on the amount of time needed to achieve the same course outcomes in a purely face-to-face format.

Refer to the following policies:

402.00 Academic Code of Conduct
263.00 Student Appeal of Course Grades
403.00 Student Code of Conduct

Disability Services Program:
Cowley College, in recognition of state and federal laws, will accommodate a student with a documented disability. If a student has a disability which may impact work in this class which requires accommodations, contact the Disability Services Coordinator.