Prepared by the Department of Mathematics
Date of Departmental Approval: March 29, 2017
Date approved by Curriculum and Programs: April 12, 2017 Effective: Fall 2017

1. **Course Number: CSC105**
   Course Title: Computer Programming I: Python

2. **Description:** In this introduction to the field of computer science, students use projects and teamwork to design, implement, and test programs in Python. Programming style, expression, and documentation are emphasized. Object-oriented programming methodology, graphical user interfaces, debugging techniques, string processing, and basic searching and sorting algorithms are covered. Python provides an introduction to programming for students in any academic discipline.

3. **Student Learning Outcomes (instructional objectives, intellectual skills):** Upon successful completion of this course, students are able to do the following:
   - Develop algorithms and solve problems with Python.
   - Write programs using iteration, conditionals, simple recursion, and basic data structures such as strings, arrays, lists, and dictionaries.
   - Write functions that read and manipulate files and folders by interacting with operating system commands.
   - Employ abstraction in problem solutions by writing programs that make calls to other user-defined functions.
   - Distinguish between storage requirements and functionality of primitive data types and objects.
   - Successfully employ debugging tools, stepwise refinement, and top down design to implement a working solution given specifications to a problem.

4. **Credits:** 3 credits

5. **Satisfies General Education Requirement:** No

6. **Prerequisite:** None

7. **Semester(s) Offered:** Spring

8. **Suggested General Guidelines for Evaluation:** programs, quizzes, tests, group projects

9. **General Topical Outline (Optional):**

1) Elements of Programming
   a. First Program
   b. Variables, Expressions, and Statements
   c. Built-in Types of Data
   d. Conditionals and Loops
2) Structures and 1/0
   a. Strings
   b. Arrays and Lists
   c. Tuples and Dictionaries
   d. Input and Output
3) Functions and Modules
   a. Defining Functions
   b. Modules and Clients
   c. Recursion
4) Object-Oriented Programming
   a. Using Data Types
   b. Creating Data Types
   c. Designing Data Types
5) Algorithms and Data Structures
   a. Performance
   b. Sorting and Searching
   c. Stacks and Queues
   d. Symbol Tables