# GROSSMONT COLLEGE

Official Course Outline

COMPUTER SCIENCE INFORMATION SYSTEMS 250 - INTRODUCTION TO PYTHON PROGRAMMING

1. Course Number Course Title Semester Units Semester Hours

 CSIS 250 Introduction to Python ~~4~~ 3 hours lecture: 48-54 hours

 Programming 3 hours lab: 48-54 hours 96-108 outside-of-class hours

 for lecture

 192-216 total hours

2. Course Prerequisites

 None.

 Corequisite

 None

 Recommended Preparation

 A “C” grade or higher or Pass in CSIS 119 or equivalent**.**

3. Catalog Description

This is an introductory course in Python programming. Topics covered include basic language syntax, functions, control flow, looping, data types, strings, lists, dictionaries, exception handling, software tools and libraries, and functional vs object oriented programming concepts. This course is intended for persons with a prior background in any programming language.

4. Course Objectives

The student will:

a. Design and prepare programs in the Python programming language.

 b. Create functions and modules.

 c. Analyze problems for use of proper data types and logic structures.

 d. Store and manage in-memory data with lists and dictionaries.

 e. Utilize libraries to perform specialized processing.

 f. Write and read text data files with Python.

 g. Edit, compile, run and debug Python programs within an integrated set of software tools.

5. Instructional Facilities

 Standard computer lab with one internet-connected workstation per student with appropriate software installed.

6. Special Materials Required of Student

 Flash/USB drive or cloud storage for backup of in-class work.

COMPUTER SCIENCE INFORMATION SYSTEMS 250 - INTRODUCTION TO PYTHON PROGRAMMING

 Page 2

7. Course Content

a. Historical survey of Python and their place among other programming languages.

 b. Portability between platforms.

 c. Python program structure and language syntax.

 d. Decisions (branching) in Python.

 e. Iteration in Python.

 f. Python operators.

 g. Data structures including standard data types and abstract data types.

 h. Functions and modules.

 i. Input/Output.

 j. Algorithms and Problem Solving.

 k. String manipulation.

 l. Libraries.

 m. Data file reading and writing.

 n. Source editors, compilers, and integrated debugging.

8. Method of Instruction

 a. Lecture

 b. Practical and realistic assignments to apply concepts discussed.

 c. Extensive practice in writing programs to run on computer. Simulation, where applicable, of real-life situations.

9. Methods of Evaluating Student Performance

 a. Projects and hands-on labs: an example would include the writing of a program to solve a problem with a scientific or business situation or an interactive game.

 b. Written tests and quizzes.

 c. Final written or coding examination.

10. Outside Class Assignments

a. Prepare programming projects such as scientific, business, and action game programs in the designated language.

 b.. Textbook reading assignments.

 c. Algorithm and problem-solving exercises.

11. Texts

 a. Required Text(s):

 Lutz, Mark. *Learning Python*, 5th Edition, Sebastopol, CA: O’Reilly Media, 2013.

 b. Supplementary texts and workbooks:

 None.

 Addendum: Student Learning Outcomes

Upon completion of this course, our students will be able to do the following:

a. Given a problem specification, design and develop a Python program solution that successfully uses an ordered set of executable steps to successfully meet program requirements.

b. Compile, run**,** test and debug object-oriented Java programs using an integrated set of software tools.

Date approved by the Governing Board: May 15, 2018