COMS 120 - Introduction to Programming Concepts and Methodologies Course Outline

Approval Date: 03/11/2021
Effective Date: 08/13/2021

SECTION A

Unique ID Number	CCC000169494
Discipline(s)	Computer Science
Division	Career Education and Workforce Development
Subject Area	Computer Studies
Subject Code	COMS
Course Number	120
Course Title	Introduction to Programming Concepts and Methodologies
TOP Code/SAM Code	0707.10 - Computer Programming/Programmer, General* / C - Occupational
Rationale for adding this course to the curriculum	Removed Recommended Preparation MATH 90 per AB 705
Units	3
Cross List	*N/A*
Typical Course Weeks	18

Total Instructional Hours

	Contact Hours
Lecture	54.00
Lab	0.00
Activity	0.00
Work Experience	0.00
Outside of Class Hours	108.00

Total Contact Hours	54
Total Student Hours	162
Open Entry/Open Exit	No
Maximum Enrollment	30
Grading Option	Letter Grade or P/NP

Distance Education Mode of Instruction
On-Campus
Hybrid
Entirely Online

SECTION B

General Education Information:

SECTION C

Course Description

Repeatability May be repeated 0 times

Catalog Description This course is an introduction to the basic principles of programming using an object-oriented programming language. Topics include the software development life-cycle, program design tools and programming environments, documentation, coding, data types, arrays, control structure, algorithms, file input/output, error handling, parameters, as well as principles of testing and designing test data.

Schedule Description This course is an introduction to the basic principles of programming using an object oriented programming language.

SECTION D

Condition on Enrollment

1a. Prerequisite(s): *None*

1b. Corequisite(s): *None*

1c. Recommended: *None*

1d. Limitation on Enrollment: *None*

SECTION E

Course Outline Information

1. Student Learning Outcomes:

A. Define the software development life-cycle and principles of structured programming then design, code, test, and debug a program.

2. Course Objectives: Upon completion of this course, the student will be able to:

A. Describe the software development life-cycle.
B. Describe the principles of structured programming and be able to describe, design, implement, and test structured programs using currently accepted methodology.
C. Explain what an algorithm is and its importance in computer programming.
D.

3. Course Content

A. Software life-cycle including design, development, styles, documentation, testing and maintenance
B. Procedural versus objected-oriented programming
    a. Survey of current languages
C. Program design tools and programming environments
D. Documentation
E. Coding conventions
F. Data types, variables, expressions, and sequential processing
G. Arrays
    a. Declaring and allocating arrays
    b. Multiple-subscripted arrays
H. Control structure

a. Selective structures: if and switch
b. Repetitive structures: loops

I. Algorithms including simple sorting and searching
J. File I/O
   a. Files and streams
   b. Sequential access files
K. Error handling
L. Passing parameters by value and by reference
M. Principles of testing and designing test data
N.

4. Methods of Instruction:

Lecture: Example: Lecture on the history of BASIC programming.

5. Methods of Evaluation: Describe the general types of evaluations for this course and provide at least two, specific examples.

Typical classroom assessment techniques

Exams/Tests -- Complete 4 to 6 exams.
Projects -- Create a program to compute inflation.
Class Work -- Complete Tutorial 1.
Home Work -- Answer selected Review Questions and Programming Questions at the end of the chapter. (See chapter 3 in book.)
Final Exam -- Complete a 50-question multiple choice and true/false based exam.
Mid Term -- Complete a 50-question multiple choice and true/false based exam.

Letter Grade or P/NP

6. Assignments: State the general types of assignments for this course under the following categories and provide at least two specific examples for each section.

A. Reading Assignments
   Read assigned chapters from the textbook.

   Examples using the Deitel book include:
   Read Chapter 1 Introduction to Computers and C++.
   Read Chapter 2 Introduction to C++ Programming;
   Input/Output and Operators.
B. Writing Assignments
   Write assignments from the textbook.
   Complete assigned lab exercises and projects.

   1. Example of Lab Assignment: Code a Card Shuffling and Dealing Simulation Program. (See Page 882 in the Deitel book)

   2. Example of Homework assignment: Complete the following Case Study: A Transaction-Processing Program. (See Page 621 in the Deitel book)
C. Other Assignments
   -

7. Required Materials

A. EXAMPLES of typical college-level textbooks (for degree-applicable courses) or other print materials.

Book #1:
Author: Deitel, P.

Title:	C++ How to Program
Publisher:	Pearson
Date of Publication:	2016
Edition:	10

B. Other required materials/supplies.