



## COMS 216 - Programming Concepts and Methodology II Course Outline

Approval Date: 03/10/2022

Effective Date: 08/12/2022

### SECTION A

**Unique ID Number** CCC000567109

**Discipline(s)** Computer Science

**Division** Career Education and Workforce Development

**Subject Area** Computer Studies

**Subject Code** COMS

**Course Number** 216

**Course Title** Programming Concepts and Methodology II

**TOP Code/SAM Code** 0707.10 - Computer Programming/Programmer, General\* /  
D - Possible Occupational

**Rationale for adding this course to the curriculum** Five-year review- matching C ID COMP 132

**Units** 3

**Cross List** N/A

**Typical Course Weeks** 18

**Total Instructional Hours**

#### Contact Hours

**Lecture** 54.00

**Lab** 18.00

**Activity** 0.00

**Work Experience** 0.00

**Outside of Class Hours** 108.00

---

**Total Contact Hours** 72

**Total Student Hours** 180

**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  
Online with Proctored Exams

## SECTION B

**General Education Information:**

## SECTION C

### Course Description

**Repeatability** May be repeated 0 times

**Catalog Description** Application of software engineering techniques to the design and development of large programs; data abstraction and structures and associated algorithms.

**Schedule Description**

## SECTION D

### Condition on Enrollment

#### 1a. Prerequisite(s)

- COMS 215 with a minimum grade of C or better

**1b. Corequisite(s):** *None*

**1c. Recommended:** *None*

**1d. Limitation on Enrollment:** *None*

## SECTION E

### Course Outline Information

#### 1. Student Learning Outcomes:

- A. Demonstrate application of advanced programming techniques and development of large programs.

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

- A. Write programs that use each of the following data structures: arrays, records, strings, linked lists, stacks, queues, and hash tables
- B. Implement, test, and debug simple recursive functions and procedures
- C. Evaluate tradeoffs in lifetime management (reference counting vs. garbage collection)
- D. Explain how abstraction mechanisms support the creation of reusable software components
- E. Design, implement, test, and debug simple programs in an object-oriented programming language
- F. Compare and contrast object-oriented analysis and design with structured analysis and design
- G.

#### 3. Course Content

Course Content:

I. Programming Fundamentals (PF)

PF3. Fundamental data structures  
Minimum coverage time: 12 hours

Topics

- A. Primitive types
- B. Arrays
- C. Records
- D. Strings and string processing
- E. Data representation in memory
- F. Static, stack, and heap allocation
- G. Runtime storage management
- H. Pointers and references
- I. Linked structures
- J. Implementation strategies for stacks, queues, and hash tables
- K. Implementation strategies for trees
- L. Strategies for choosing the right data structure

PF4. Recursion  
Minimum coverage time: 5 hours

Topics

- A. The concept of recursion
- B. Recursive mathematical functions
- C. Simple recursive procedures
- D. Divide-and-conquer strategies
- E. Recursive backtracking
- F. Implementation of recursion

II. Programming Languages (PL)

PL4. Declarations and types  
Minimum coverage time: 3 hours

Topics

- A. The conception of types as a set of values together with a set of operations
- B. Declaration models (binding, visibility, scope, and lifetime)
- C. Overview of type-checking
- D. Garbage collection

PL5. Abstraction Mechanisms  
Minimum coverage time: 3 hours

## Topics

- A. Procedures, functions, and iterators as abstraction mechanisms
- B. Parameterization mechanisms (reference vs. value)
- C. Activation records and storage management
- D. Type parameters and parameterized types - templates or generics
- E. Modules in programming languages

## PL6. Object-oriented programming

Minimum coverage time: 10 hours

## Topics

- A. Object-oriented design
- B. Encapsulation and information-hiding
- C. Separation of behavior and implementation
- D. Classes and subclasses
- E. Inheritance (overriding, dynamic dispatch)
- F. Polymorphism (subtype polymorphism vs. inheritance)
- G. Class hierarchies
- H. Collection classes and iteration protocols
- I. Internal representations of objects and method tables

## III. Software Engineering (SE)

### SE1. Software design

Minimum coverage time: 8 hours

## Topics

- A. Fundamental design concepts and principles
- B. Design strategy
- C.

### 4. Methods of Instruction:

**Discussion:** Discuss why a Software Development Kit is necessary

**Lab:** Write a program that calculates mortgage interest for a 30 year loan

**Lecture:** Discuss the history of Advanced Object-Oriented programming

**Projects:** Write a program that tracks the weather in Napa for 90 days. The program must include high temperature, low temperature, humidity, and precipitation.

**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 -- At least two exams (midterm and final) will be administered with possible multiple choice/essay/true-false/short answer questions

Quizzes -- Possible unannounced quizzes on the course material covered. Possible short answer/true-false/multiple choice/essay questions.

Projects -- Write a program that keeps track of grades. The section has 30 students. Must calculate overall average. Final exam weight set to 25%, Midterm Exam set to 15%, Participation 10%, and Homework at 50%. Should show overall course average.

Home Work -- Example test questions: Essay: Provide a brief history of C++.

Lab Activities -- Example lab activity: Write a program that calculates overtime after 8 hours of work. Regular pay is set to \$12.50 hour. Overtime is set to 1.5. If over 12 hours, pay is set to 2.0.

Final Exam -- Comprehensive exam with possible multiple choice/true-false/short-answer/essay questions

Mid Term -- Comprehensive exam with possible multiple choice/true-false/short-answer/essay questions

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 Chapter 1 related to the history of C++.

B. Writing Assignments

Write a one-page essay on the concept of recursion.

C. Other Assignments

Write a program that uses each of the following data structures: arrays, records, and strings.

## **7. Required Materials**

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

Book #1:

Author: Malik, D. S.

Title: C++ Programming: Program Design Including Data Structures

Publisher: Cengage

Date of Publication: 2017

Edition: 8

**B. Other required materials/supplies.**