

COMS-216: PROGRAMMING CONCEPTS & METHODOLOGY II

Effective Term

Fall 2022

BOT Approval

03/10/2022

SECTION A - Course Data Elements**CB04 Credit Status**

Credit - Degree Applicable

Discipline**Minimum Qualifications****And/Or**

Computer Science (Master's Degree)

Subject Code

COMS - Computer Studies

Course Number

216

Department

Computer Studies

Division

Business and Computer Studies

Full Course Title

Programming Concepts & Methodology II

Short Title

Prog Concepts & Methodology II

CB03 TOP Code

0707.10 - *Computer Programming

CB08 Basic Skills Status

NBS - Not Basic Skills

CB09 SAM Code

D - Possibly Occupational

Rationale

Five-year review- matching C ID COMP 132

SECTION B - Course Description**Catalog Course Description**

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

SECTION C - Conditions on Enrollment**Open Entry/Open Exit**

No

Repeatability

Not Repeatable

Grading Options

Letter Grade or Pass/No Pass

Allow Audit

Yes

Requisites**Prerequisite(s)**

Completion of COMS-215 with a minimum grade of C.

Requisite Justification**Requisite Description**

Course in a Sequence

Subject

COMS

Course #

215

Level of Scrutiny

Content Review

Upon entering this course, students should be able to:

Completion of COMS-215 with a minimum grade of C.

SECTION D - Course Standards**Is this course variable unit?**

No

Units

3.00000

Lecture Hours

54.00

Lab Hours

18.00

Outside of Class Hours

108

Total Contact Hours

72

Total Student Hours

180

Distance Education Approval**Is this course offered through Distance Education?**

Yes

Online Delivery Methods

DE Modalities	Permanent or Emergency Only?
Entirely Online	Permanent
Hybrid	Permanent
Online with Proctored Exams	Permanent

SECTION E - Course Content**Student Learning Outcomes**

Upon satisfactory completion of the course, students will be able to:

1. Demonstrate application of advanced programming techniques and development of large programs.

Course Objectives

Upon satisfactory completion of the course, students will be able to:

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

Course Content

Course Content:

I. Programming Fundamentals

(PF) PF3. Fundamental data structures

Minimum coverage time: 12 hours

Topics

1. Primitive types
2. Arrays
3. Records
4. Strings and string processing
5. Data representation in memory
6. Static, stack, and heap allocation
7. Runtime storage management
8. Pointers and references
9. Linked structures
10. Implementation strategies for stacks, queues, and hash tables
11. Implementation strategies for trees
12. Strategies for choosing the right data structure

PF4. Recursion

Minimum coverage time: 5 hours

Topics

1. The concept of recursion
2. Recursive mathematical functions
3. Simple recursive procedures
4. Divide-and-conquer strategies
5. Recursive backtracking
6. Implementation of recursion II. Programming Languages

(PL) PL4. Declarations and types

Minimum coverage time: 3 hours

Topics

1. The conception of types as a set of values together with a set of operations
2. Declaration models (binding, visibility, scope, and lifetime)
3. Overview of type-checking
4. Garbage collection

PL5. Abstraction Mechanisms

Minimum coverage time: 3 hours

Topics

1. Procedures, functions, and iterators as abstraction mechanisms
2. Parameterization mechanisms (reference vs. value)
3. Activation records and storage management
4. Type parameters and parameterized types - templates or generics
5. Modules in programming languages

PL6. Object-oriented programming

Minimum coverage time: 10 hours

Topics

1. Object-oriented design
2. Encapsulation and information-hiding
3. Separation of behavior and implementation
4. Classes and subclasses
5. Inheritance (overriding, dynamic dispatch)
6. Polymorphism (subtype polymorphism vs. inheritance)
7. Class hierarchies
8. Collection classes and iteration protocols
9. Internal representations of objects and method tables

III. Software Engineering (SE)**SE1. Software design**

Minimum coverage time: 8 hours

Topics

1. Fundamental design concepts and principles
2. Design strategy

Methods of Instruction**Methods of Instruction**

Types	Examples of learning activities
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.

Online Adaptation

Types	Examples of learning activities
Activity	
Directed Study	
Discussion	
Group Work	
Individualized Instruction	
Journal	
Lab	

Instructor-Initiated Online Contact Types

Announcements/Bulletin Boards
 Chat Rooms
 Discussion Boards
 E-mail Communication
 Telephone Conversations
 Video or Teleconferencing

Student-Initiated Online Contact Types

Chat Rooms

Discussions
Group Work

Course design is accessible
Yes

Methods of Evaluation

Methods of Evaluation

Types	Examples of classroom assessments
Exams/Tests	At least two exams (midterm and final) will be administered with possible multiple choice/essay/true-false/short answer questions. 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.
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.
Homework	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

Assignments

Reading Assignments

Read Chapter 1 related to the history of C++.

Writing Assignments

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

Outside-of-Class Assignments

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

SECTION F - Textbooks and Instructional Materials

Material Type

Textbook

Author

Malik, D. S

Title

C++ Programming: Program Design Including Data Structures

Edition/Version

8th

Publisher

Cengage

Year

2017

SECTION G - Diversity, Equity and Inclusivity

Course Codes (Admin Only)

CB00 State ID

CCC000567109

CB10 Cooperative Work Experience Status

N - Is Not Part of a Cooperative Work Experience Education Program

CB11 Course Classification Status

Y - Credit Course

CB13 Special Class Status

N - The Course is Not an Approved Special Class

CB23 Funding Agency Category

Y - Not Applicable (Funding Not Used)

CB24 Program Course Status

Program Applicable

Allow Pass/No Pass

Yes

Only Pass/No Pass

No