II.A.13.06 See Page 4

CS M135: PROGRAMMING CONCEPTS AND METHODOLOGY II

Originator
lalnaji

Co-Contributor(s)

Name(s)
Nikjeh, Esmaail (enikjeh)

College
Moorpark College

Attach Support Documentation (as needed)
CS M135_ACM - CS2013-final-report.pdf
CS M135_COMP 132-template_342.doc
CS M135_state approval letter_CCC000608535.pdf

Discipline (CB01A)
CS - Computer Science

Course Number (CB01B)
M135

Course Title (CB02)
Programming Concepts and Methodology II

Banner/Short Title
Prog. Concepts and Method. II

Credit Type
Credit

Start Term
Fall 2022

Catalog Course Description
Presents the design of programming applications using software engineering techniques. Discusses the development of large programs, data abstraction and structures and the associated algorithms.

Taxonomy of Programs (TOP) Code (CB03)
0706.00 - Computer Science (transfer)

Course Credit Status (CB04)
D (Credit - Degree Applicable)

Course Transfer Status (CB05) (select one only)
A (Transferable to both UC and CSU)

Course Basic Skills Status (CB08)
N - The Course is Not a Basic Skills Course

SAM Priority Code (CB09)
E - Non-Occupational

Course Cooperative Work Experience Education Status (CB10)
N - Is Not Part of a Cooperative Work Experience Education Program

Course Classification Status (CB11)
Y - Credit Course

Educational Assistance Class Instruction (Approved Special Class) (CB13)
N - The Course is Not an Approved Special Class

Course Prior to Transfer Level (CB21)
Y - Not Applicable

Course Noncredit Category (CB22)
Y - Credit Course

Funding Agency Category (CB23)
Y - Not Applicable (Funding Not Used)

Course Program Status (CB24)
1 - Program Applicable

General Education Status (CB25)
Y - Not Applicable

Support Course Status (CB26)
N - Course is not a support course

Field trips
Will not be required

Grading method
(L) Letter Graded

Alternate grading methods
(P) Pass/No Pass Grading

Does this course require an instructional materials fee?
No

Repeatable for Credit
No

Is this course part of a family?
No

Units and Hours

Carnegie Unit Override
No

In-Class

Lecture
Minimum Contact/In-Class Lecture Hours
52.5
Maximum Contact/In-Class Lecture Hours
52.5

Activity

Laboratory

Total in-Class

Total in-Class
Total Minimum Contact/In-Class Hours
52.5
Total Maximum Contact/In-Class Hours
52.5

Outside-of-Class

Internship/Cooperative Work Experience

Paid

Unpaid

Total Outside-of-Class

Total Outside-of-Class
Minimum Outside-of-Class Hours
105
Maximum Outside-of-Class Hours
105

Total Student Learning

Total Student Learning
Total Minimum Student Learning Hours
157.5
Total Maximum Student Learning Hours
157.5

Minimum Units (CB07)
3
Maximum Units (CB06)
3

Prerequisites
CS M125

Entrance Skills

Entrance Skills
CS M125

Prerequisite Course Objectives
CS M125-describe the basic organization of a computer system.
CS M125-describe the basic components, syntax, and semantics of the C++ programming language.
CS M125-analyze programming problems and design algorithms to solve those problems.
CS M125-identify sequential, selection, and iteration control structures.
CS M125-apply the concepts of structured programming including function usage and parameter passing.
CS M125-apply composite data types such as arrays and structures.
CS M125-demonstrate and understanding of user defined data types such as enumerations and structured data.
CS M125-describe and apply dynamic memory allocation using pointers.
CS M125-identify file input and output.
CS M125-identify good programming practice and style.

Requisite Justification

Requisite Type

Prerequisite

Requisite

CS M125

Requisite Description

Course in a sequence

Level of Scrutiny/Justification

Closely related lecture/laboratory course

Student Learning Outcomes (CSLOs)

	Upon satisfactory completion of the course, students will be able to:
1	analyze programs and program fragments in order to determine what the code is doing, what errors the code may have, and what corrections should be made to the code.
2	apply complex structures such as linked lists, hash tables and trees and graphs and other data structures to solve complex problems.
3	communicate technical solutions to non-technical audience.
4	describe and understand which data items are in which part of memory and how this affects program processing, design, and implementation.

Course Objectives

	Upon satisfactory completion of the course, students will be able to:
1	analyze programs and program fragments in order to determine what the code is doing, what errors the code may have, and what corrections should be made to the code.
2	demonstrate and understand the principles of recursion and be able to use this technique in programs.
3	understand the principles of hash tables and trees.
4	evaluate what tradeoffs are involved when choosing different structures and techniques, including software reuse considerations.
5	compare and contrast object-oriented analysis and design with structured programming analysis and design.
6	describe and understand which data items are in which part of memory and how this affects program processing, design, and implementation.

Course Content

Lecture/Course Content

10% - Declaration and Types:
• Atomic versus composite
• Primitive data types
• Built-in versus user defined
• Using composite data types such as strings, stacks, heaps, and
pointers and references
30% - Data Structures
• Linked structures such as linked lists and Binary Tree, Binary search tree, Heap, and AVL tree, Graph Theory: Breath First Search (BFS), Depth First Search (DFS), and shortest path algorithms.
• Structures used for sorting and searching including heaps and hash tables
• Queues and stacks
• Strategies for using appropriate structures
5% - Recursion
• What is recursion and the underlying concepts
• Different types of recursion
• When to use recursion versus iteration
15% - Software Engineering:

• Reuse
• Abstraction
• Application design including understanding how to evaluate the tradeoffs required
• Divide and conquer strategies
35% - Object-Oriented Programming (OOP):
• Reasons for OOP
• Separation of implementation and interface
• Encapsulation and data hiding
• Classes and objects
• OOP techniques including inheritance, polymorphism, generalized (parameterized) programming, and operator overloading (where applicable for the programming language being used).
• Overloading versus overriding
5% - Abstraction Mechanisms:
• Data representation in memory including the different areas of memory - data section, data stack, free store (or heap), activation records (stack frames), parameter passing by value and reference, binding, scope (visibility), lifetime, type-checking, and garbage collection
• Data independence using type parameters and parameterized types
- generic programming

Laboratory or Activity Content

Not applicable

Methods of Evaluation

Which of these methods will students use to demonstrate proficiency in the subject matter of this course? (Check all that apply):

Problem solving exercises
Skills demonstrations

Methods of Evaluation may include, but are not limited to, the following typical classroom assessment techniques/required assignments (check as many as are deemed appropriate):

Individual projects
Problem-solving exams
Problem-solving homework
Quizzes
Skills demonstrations
Other (specify)

Other

• Classroom discussion
• Written homework exercises to demonstrate knowledge of concepts and techniques.
• Programming exercises which require students to write code fragments to demonstrate that students can apply the concepts and techniques.

Instructional Methodology

Specify the methods of instruction that may be employed in this course

Distance Education
Lecture
Other (specify)

Specify other method of instruction

• Practice problems to develop proper programming skills and techniques.
• Student/instructor interaction using questions and answers.

Describe specific examples of the methods the instructor will use:

The instructor will have students identify and describe the properties of a variable such as its associated address, value, scope, persistence, and size. The instructor will assign students a project in which students will defend the importance of types and type-checking in providing abstraction and safety. These may include exercises through which students explore course concepts using the textbook and additional research. Other assignments may also be projects which require students to write programs. Students will submit their assignments online and get feedback from the instructor. This can be an iterative process in that students can receive feedback and then be able to improve their submission, if necessary.

Quizzes may be on a weekly or topic basis (or some other time frame as determined by the instructor) where students will test their knowledge of the material.

Representative Course Assignments

Writing Assignments

1. Answer homework questions, using correct grammar, syntax, and terms on concepts such as programming languages, algorithm development, and evaluating and understanding programs.

2. Comment on computer programs, using correct grammar, syntax, and terminology, to document what a program is doing. These comments are in the programs themselves. As an example, comments in a program where each class has a comment which includes:

- The class name
- A list of each of the attributes (variables) associated with the class and what they represent
- An explanation of the expected class behaviors

3. Describe the differences between structured programming and object-oriented programming.

Critical Thinking Assignments

1. Analyze programs to understand the design, implementation, and issues of the programs.

2. Evaluate programs and/or program fragments to determine what errors occur and correct the syntax and/or semantics of the problems identified. Examples might include:

- A program fragment where nodes in a linked list object are being lost
- A program using a binary tree which crashes
- A recursive function whose behavior using certain inputs is unexpected.

Reading Assignments

1. Go over our Chapter 10: "Binary Search Tree" and read about the characteristics of search binary trees. Pay close attention to how new nodes are added and removed.

2. Read and review the following published document explaining the applications of Binary Trees: https://iq.opengenus.org/applications-of-binary-tree/

Skills Demonstrations

1. Using the code provided to you in chapter 1. Add a member function named DisplayGraph that will display the graph in Text Graphics. Take into consideration that the function must take a 2D array as input. The 2D array represents the desired graph to display.

2. Create a PrintManager class that keeps track of print jobs using an array of pointers and using Linked Lists. The PrintManager will print jobs based on First Come First Server basis. Allow the user to user the PrintManager class to add and delete print jobs during run time.

Outside Assignments

Representative Outside Assignments

1. Read articles from professional publications in addition to the text assignments which highlight specific areas such as careers, programming issues, and technology issues.

2. Work on program and program fragments in order to find and correct errors.

Articulation

C-ID Descriptor Number

COMP 132

Status

Approved

Equivalent Courses at 4 year institutions

University	Course ID	Course Title	Units
Cal Poly Pomona	CS240	Data Structures and Algorithms I	4
CSU Los Angeles	CS 2012	Introduction to Programming II	5
CSU Northridge	COMP 182, 182L	Data Structures and Program Design	3, 1
CSU Chico	CSCI 211	Programming and Algorithms II	4

CSU Dominguez Hills	CSC 123	Introduction to Computer Science and Programming II	4
CSU Long Beach	CECS 274	Object Oriented Programming and Data Structures	3

District General Education

A. Natural Sciences

B. Social and Behavioral Sciences

C. Humanities

D. Language and Rationality

E. Health and Physical Education/Kinesiology

F. Ethnic Studies/Gender Studies

Course is CSU transferable
Yes

CSU Baccalaureate List effective term:
F2020

CSU GE-Breadth

Area A: English Language Communication and Critical Thinking

Area B: Scientific Inquiry and Quantitative Reasoning

Area C: Arts and Humanities

Area D: Social Sciences

Area E: Lifelong Learning and Self-Development

Area F: Ethnic Studies

CSU Graduation Requirement in U.S. History, Constitution and American Ideals:

UC TCA

UC TCA
Proposed
Approved

Date Proposed:
02/05/2019

Effective term:
Fall 2020

IGETC

Area 1: English Communication

Area 2A: Mathematical Concepts & Quantitative Reasoning

Area 3: Arts and Humanities

Area 4: Social and Behavioral Sciences

Area 5: Physical and Biological Sciences

Area 6: Languages Other than English (LOTE)

Textbooks and Lab Manuals

Resource Type
Textbook

Classic Textbook
Yes

Description
Gaddis, Tony. Starting Out With C++: From Control Structures Through Objects. 8th ed. Pearson, 2014.

Resource Type
Textbook

Classic Textbook
Yes

Description
Deitel, Paul, and Harvey Deitel. C++: How to Program. 10th ed. Pearson, 2017.

Library Resources

Assignments requiring library resources
Locate articles in professional and academic publications, using the Library's print and online resources, on such topics as careers and issues in programming and technology.

Sufficient Library Resources exist
Yes

Distance Education Addendum

Definitions

Distance Education Modalities
Hybrid (51%–99% online)
100% online

Faculty Certifications

Faculty assigned to teach Hybrid or Fully Online sections of this course will receive training in how to satisfy the Federal and state regulations governing regular effective/substantive contact for distance education. The training will include common elements in the district-supported learning management system (LMS), online teaching methods, regular effective/substantive contact, and best practices.
Yes

Faculty assigned to teach Hybrid or Fully Online sections of this course will meet with the EAC Alternate Media Specialist to ensure that the course content meets the required Federal and state accessibility standards for access by students with disabilities. Common areas for discussion include accessibility of PDF files, images, captioning of videos, Power Point presentations, math and scientific notation, and ensuring the use of style mark-up in Word documents.
Yes

Regular Effective/Substantive Contact

Hybrid (51%–99% online) Modality:

Method of Instruction	Document typical activities or assignments for each method of instruction
Synchronous Dialog (e.g., online chat)	Instructor may be available on certain day(s) with certain time frames to help students and answer questions via online chat.
Asynchronous Dialog (e.g., discussion board)	Instructor will post a question(s), students will respond to the question(s).
E-mail	Instructor will email students with announcements about the course or an upcoming event. Students in turn may email the instructor with their questions. Students will email assignments to the instructor.
Other DE (e.g., recorded lectures)	Instructor may recorded the lectures and post them for students to view within a specified time frame to be ready for the accompanying assignment. Students will upload their assignment to the course webpage.

100% online Modality:

Method of Instruction	Document typical activities or assignments for each method of instruction
Asynchronous Dialog (e.g., discussion board)	Instructor will post a question(s), students will respond to the question(s).
Synchronous Dialog (e.g., online chat)	Instructor may be available on certain day(s) with certain time frames to help students and answer questions via online chat.
E-mail	Instructor will email students with announcements about the course or an upcoming event. Students in turn may email the instructor with their questions. Students will email assignments to the instructor.
Other DE (e.g., recorded lectures)	Instructor may recorded the lectures and post them for students to view within a specified time frame to be ready for the accompanying assignment. Students will upload their assignment to the course webpage.

Examinations

Hybrid (51%–99% online) Modality
On campus
Online

Primary Minimum Qualification
COMPUTER SCIENCE

Review and Approval Dates

Department Chair
MM/DD/YYYY

Dean
MM/DD/YYYY

Technical Review
MM/DD/YYYY

Curriculum Committee
MM/DD/YYYY

DTRW-I
MM/DD/YYYY

Curriculum Committee
MM/DD/YYYY

Board
MM/DD/YYYY

CCCCO
MM/DD/YYYY

Control Number
CCC000608535

DOE/accreditation approval date
MM/DD/YYYY