I. CATALOG INFORMATION

A. Discipline: COMPUTER SCIENCE (CS)

B. Subject Code and Number: CS M20

C. Course Title: Object-Oriented Data Structures and Algorithm Design

D. Credit Course units:

Units: 4

Lecture Hours per week: 3

Lab Hours per week : 3

Variable Units : No

E. Student Learning Hours:

Lecture Hours:

Classroom hours: 52.5 - 52.5

Laboratory/Activity Hours:

Laboratory/Activity Hours 52.5 - 52.5

Total Combined Hours in a 17.5 week term: 105 - 105

F. Non-Credit Course hours per week

G. May be taken a total of: [X] 1 [ ] 2 [ ] 3 [ ] 4 time(s) for credit

H. Is the course co-designated (same as) another course: No [X] Yes [ ]
If YES, designate course Subject Code & Number:

I. Course Description:

Applies object-oriented design and programming methods to abstract data types such as stacks, queues, trees, and graphs. Introduces comparative analysis of searching and sorting algorithms. Covers such topics as recursion, class design, Standard Template Library including sequential and associative containers, algorithms, and function objects.

J. Entrance Skills

*Prerequisite: No [ ] Yes [X] Course(s)
CS M10B

*Corequisite: No [X] Yes [ ] Course(s)

Limitation on Enrollment: No [X] Yes [ ]

Recommended Preparation: No [X] Yes [ ] Course(s)

Other: No [X] Yes [ ]

K. Other Catalog Information:

II. COURSE OBJECTIVES

Upon successful completion of the course, a student will be able to:

		Methods of evaluation will be consistent with, but not limited by, the following types or examples.
1	describe and apply the concepts of data abstraction through the use of classes and structures.	Graded assignments Programming projects Quizzes Exams
2	use class libraries, especially the Standard Template Library.	Graded assignments Programming projects Quizzes Exams
3	analyze problems, develop good program design skills, and implement solutions using object-oriented methodology.	Graded assignments Programming projects Quizzes Exams
4	develop, analyze, and implement complex algorithms.	Graded assignments Programming projects Quizzes Exams
5	describe, design with, and implement advanced dynamic memory concepts.	Graded assignments Programming projects Quizzes Exams
6	describe, evaluate, and implement data structures such as linked lists, stacks, queues, trees, graphs, and related abstract data types.	Graded assignments Programming projects Quizzes Exams
7	analyze, evaluate, and implement sorting and searching concepts.	Graded assignments Programming projects Quizzes Exams
		Graded assignments

8	understand and utilize good programming techniques and integrated development environment (IDE) tools.	Programming projects Quizzes Exams
9	analyze the efficiency of the basic data structures and techniques.	Graded assignments Programming projects Quizzes Exams
10	identify and correct the errors in a given program or program segment pertaining to data structures.	Graded assignments Programming projects Quizzes Exams
11	trace the execution and give the output of a given program or program segment pertaining to data structures.	Graded assignments Programming projects Quizzes Exams

III. COURSE CONTENT

Estimated %	Topic	Learning Outcomes
Lecture (must total 100%)		
7.00%	Queues • Priority Queues • Circular Queues	1, 2, 3, 4, 5, 6, 8, 9
7.00%	Introduction to Data Structures • What a Data Structure is and why they are used • Analyzing algorithms – Big O notation and the concept of software efficiency	1, 2, 4, 5, 8
14.00%	Trees • Binary trees • Search tees • AVL trees • Analysis Heaps • Definition • Basic algorithms • Analysis • Applications	1, 2, 3, 4, 5, 6, 7, 8, 10, 11
7.00%	Hash Tables • Data structures used in Hash Tables • Collision and solutions to collisions	1, 2, 3, 4, 5, 6, 7, 8, 10, 11
17.00%	Linked Lists and Pointers • Advanced dynamic memory concepts	1, 2, 3, 4, 5, 6, 8, 9
12.00%	Stacks • Push • Pop	1, 2, 3, 4, 5, 6, 8, 9
	Recursion	1, 2, 3, 4,

8.00%	• Fractal basis of recursion • Backtracking	5, 6, 8, 10, 11
13.00%	Graphs • Definition • Basic algorithms • Analysis • Minimal spanning tree • Shortest path	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
15.00%	Standard Template Libraries (STL) Vectors Stacks Queues MAP	2
Lab (must total 100%)		
16.50%	Project(s) to create a program or programs "from scratch" which will incorporate: • Creating classes and objects • Designing and implementing linked list classes • Using object-oriented concepts such as templates and operator overloading • Using return codes - either constant integers or enumerated data type • Using private class functions to implement common class services • Appropriate use of const and & in parameter passing • Designing and implementing data independence - no coupling between the data and the linked list container (the linked list should be able to hold any type of data) • Designing and implementing front-end, back-end, and middleware classes	1, 2, 3, 4, 5, 8, 9, 10, 11
16.50%	Project(s) to create a program or programs "from scratch" which will incorporate: • Creating classes and objects • Designing and implementing stack and queue classes • Use is-a (inheritance) and has-a (composition) relationships • Use the Standard Template Library • Using object-oriented concepts such as templates and operator overloading • Using return codes - either constant integers or enumerated data type • Using private class functions to implement common class services • Appropriate use of const and & in parameter passing • Designing and implementing data independence	1, 2, 3, 4, 5, 6, 8, 9, 10, 11
16.50%	Project(s) to create a program or programs "from scratch" which will incorporate: • Creating classes and objects • Designing and implementing tree classes • Designing and implementing front-end, back-end, and middleware classes • Use is-a (inheritance) and has-a (composition) relationships • Use the Standard Template Library • Using object-oriented concepts such as templates and operator overloading • Using return codes - either constant integers or enumerated data type • Using private class functions to implement common class services • Appropriate use of const and & in parameter passing • Designing and implementing data independence	1, 2, 3, 4, 5, 6, 7, 8, 10, 11
	Project(s) to create a program or programs "from scratch" which will	

16.50%	incorporate: • Creating classes and objects • Designing and implementing heap classes • Designing and implementing front-end, back-end, and middleware classes • Use is-a (inheritance) and has-a (composition) relationships • Use the Standard Template Library • Using object-oriented concepts such as templates and operator overloading • Using return codes - either constant integers or enumerated data type • Using private class functions to implement common class services • Appropriate use of const and & in parameter passing • Designing and implementing data independence	1, 2, 3, 4, 5, 6, 7, 8, 10, 11
16.50%	Project(s) to create a program or programs "from scratch" which will incorporate: • Creating classes and objects • Designing and implementing hash table classes • Designing and implementing front-end, back-end, and middleware classes • Use is-a (inheritance) and has-a (composition) relationships • Use the Standard Template Library • Using object-oriented concepts such as templates and operator overloading • Using return codes - either constant integers or enumerated data type • Using private class functions to implement common class services • Appropriate use of const and & in parameter passing • Designing and implementing data independence	1, 2, 3, 4, 5, 6, 7, 8, 10, 11
17.50%	Project(s) to create a program or programs "from scratch" which will incorporate: • Creating classes and objects • Designing and implementing graph classes • Designing and implementing front-end, back-end, and middleware classes • Use is-a (inheritance) and has-a (composition) relationships • Use the Standard Template Library • Using object-oriented concepts such as templates and operator overloading • Using return codes - either constant integers or enumerated data type • Using private class functions to implement common class services • Appropriate use of const and & in parameter passing • Designing and implementing data independence	1, 2, 3, 4, 5, 6, 8, 9, 10

IV. TYPICAL ASSIGNMENTS

A. Writing assignments

Writing assignments are required. Possible assignments may include, but are not limited to:	
1	design a linked list class to hold a series of integers. The class should have member functions for appending, inserting, and deleting nodes. Don’t forget to add a destructor that destroys the list. Demonstrate the class with a driver program to test the member functions and display the result after calling each member function.
	write a program that reads two sentences and reads them into two separate queues.

2	The program should then determine whether the sentences are identical by comparing the characters in the queues. When two non-identical characters are encountered, the program should display a message indicating that the sentences are not the same. If both queues contain the same set of characters, message should be displayed indicating that the sentences are identical.
3	a string of characters has balanced parentheses if each right parenthesis occurring in the string is matched with a preceding left parentheses, in the same way that each right brace in a C++ program is matched with a preceding left brace. Write a program that uses a stack to determine whether a string entered at the keyboard has balanced parentheses.

B. Appropriate outside assignments

Appropriate outside assignments are required. Possible assignments may include, but are not limited to:	
1	research and comment on current computer/technology industry and social trends.
2	exchange your data structure project with your classmate. Evaluate and debug it. Use your sample data to test the program and detect any logic error.

C. Critical thinking assignments

Critical thinking assignments are required. Possible assignments may include, but are not limited to:	
1	evaluate incorrect programs and/or program fragments to determine what errors occur and to correct the syntax and/or semantics of the problems identified.
2	developing algorithms for programming assignments.

V. METHODS OF INSTRUCTION

Methods of instruction may include, but are not limited to:

[X] Distance Education – When any portion of class contact hours is replaced by distance education delivery mode (Complete DE Addendum, Section XV)

[X] Lecture/Discussion

[X] Laboratory/Activity

[X] Other (Specify)
Bring a guest speaker from industry to practice real world problems to develop proper programming skills and techniques
Practice problems to develop proper programming skills and techniques
Projects and/or team work to enhance student understanding of the concepts

[ ] Optional Field Trips

[ ] Required Field Trips

VI. METHODS OF EVALUATION

Methods of evaluation may include, but are not limited to:

☐ Essay Exam	☒ Classroom Discussion	☒ Skill Demonstration
☒ Problem Solving Exam	☐ Reports/Papers/ Journals	☒ Participation
☒ Objective Exams	☒ Projects	☒ Other (specify)

Written homework exercises to demonstrate knowledge of concepts and techniques

Programming assignments requiring students to analyze a problem, determine a solution, implement the solution using a programming language, and test and verify the program

Quizzes where student demonstrate their knowledge of the material

VII. REPRESENTATIVE TEXTS AND OTHER COURSE MATERIALS

Carrano, Frank, and Timothy Henry. Data Abstraction and Problem Solving With C++: Walls and Mirrors. 7th ed. Pearson, 2016.

Weiss, Mark A. Data Structures and Algorithm Analysis in C++. 4th ed. Pearson, 2013.

Microsoft Visual Studio 2017 Express C++. Microsoft, 2017 ed.
Integrated Development Environment (IDE) for designing, writing, running, and testing C++ programs. This application is currently available as a free download from https://visualstudio.microsoft.com/downloads/

VIII. STUDENT MATERIALS FEES

☒ No ☐ Yes

IX. PARALLEL COURSES

College	*Course Number*	*Course Title*	*Units*
CSU Chico	CSCI 211	Programming and Algorithms II	4
CSU Northridge	COMP 182 + 182L	Data Structures and Program Design	4
UC Santa Barbara	CMPSC 24	Problem Solving With Computers II	4
CSU Channel Islands	COMP 151	Data Structures and Program Design	4

X. MINIMUM QUALIFICATIONS

Courses Requiring a Masters Degree:
Master's in computer science or computer engineering OR Bachelor's in either of the above AND Master's in mathematics, cybernetics, business administration, accounting or engineering OR Bachelor's in engineering AND Master's in cybernetics, engineering mathematics, or business administration OR Bachelor's in mathematics AND Master's in cybernetics, engineering mathematics, or business administration OR Bachelor's degree in any of the above AND a Master's degree in information science, computer information systems, or information systems OR the equivalent.

XI. ARTICULATION INFORMATION

A. Title V Course Classification:
   1. This course is designed to be taken either:

☐ Pass/No Pass only (no letter grade possible); or
☒ Letter grade (P/NP possible at student option)

2. Degree status:
   Either ☒ Associate Degree Applicable; or ☐ Non-associate Degree Applicable

B. Moorpark College General Education:

1. Do you recommend this course for inclusion on the Associate Degree General Education list?
   Yes: ☐ No: ☒ If YES, what section(s)?

   ☐ A1 - Natural Sciences - Biological Science
   ☐ A2 - Natural Sciences - Physical Science
   ☐ B1 - Social and Behavioral Sciences - American History/Institutions
   ☐ B2 - Social and Behavioral Sciences - Other Social Behavioral Science
   ☐ C1 - Humanities - Fine or Performing Arts
   ☐ C2 - Humanities - Other Humanities
   ☐ D1 - Language and Rationality - English Composition
   ☐ D2 - Language and Rationality - Communication and Analytical Thinking
   ☐ E1 - Health/Physical Education
   ☐ E2 - PE or Dance
   ☐ F - Ethnic/Gender Studies

C. California State University(CSU) Articulation:

1. Do you recommend this course for transfer credit to CSU? Yes: ☒ No: ☐

2. If YES do you recommend this course for inclusion on the CSU General Education list?
   Yes: ☐ No: ☒ If YES, which area(s)?

   A1 ☐ A2 ☐ A3 ☐ B1 ☐ B2 ☐ B3 ☐ B4 ☐

   C1 ☐ C2 ☐ D1 ☐ D2 ☐ D3 ☐ D4 ☐ D5 ☐

   D6 ☐ D7 ☐ D8 ☐ D9 ☐ D10 ☐ E ☐

D. University of California (UC) Articulation:

1. Do you recommend this course for transfer to the UC? Yes: ☒ No: ☐

2. If YES do you recommend this course for the Intersegmental General Education Transfer Curriculum (IGETC)? Yes: ☐ No: ☒

   IGETC Area 1: English Communication

   ☐ English Composition
   ☐ Critical Thinking-English Composition
   ☐

- [ ] Oral Communication

IGETC Area 2: Mathematical Concepts and Quantitative Reasoning

- [ ] Mathematical Concepts

IGETC Area 3: Arts and Humanities

- [ ] Arts
- [ ] Humanities

IGETC Area 4: Social and Behavioral Sciences

- [ ] Anthropology and Archaeology
- [ ] Economics
- [ ] Ethnic Studies
- [ ] Gender Studies
- [ ] Geography
- [ ] History
- [ ] Interdisciplinary, Social & Behavioral Sciences
- [ ] Political Science, Government & Legal Institutions
- [ ] Psychology
- [ ] Sociology & Criminology

IGETC Area 5: Physical and Biological Sciences (mark all that apply)

- [ ] Physical Science Lab or Physical Science Lab only (none-sequence)
- [ ] Physical Science Lecture only (non-sequence)
- [ ] Biological Science
- [ ] Physical Science Courses
- [ ] Physical Science Lab or Biological Science Lab Only (non-sequence)
- [ ] Biological Science Courses
- [ ] Biological Science Lab course
- [ ] First Science course in a Special sequence
- [ ] Second Science course in a Special Sequence
- [ ] Laboratory Activity
- [ ] Physical Sciences

IGETC Area 6: Language other than English

- [ ] Languages other than English (UC Requirement Only)
- [ ] U.S. History, Constitution, and American Ideals (CSU Requirement ONLY)
- [ ] U.S. History, Constitution, and American Ideals (CSU Requirement ONLY)

XII. REVIEW OF LIBRARY RESOURCES

A. What planned assignment(s) will require library resources and use?

The following assignments require library resources:
Use of the Library's print and online resources to locate current articles in newspapers, magazines and scholarly journals in preparation for a report on various aspects of data and coding practices.

B. Are the currently held library resources sufficient to support the course assignment?

YES: [X] NO: [ ]

If NO, please list additional library resources needed to support this course.

XIII. PREREQUISITE AND/OR COREQUISITE JUSTIFICATION

Requisite Justification for CS M10B

[X] A. Sequential course within a discipline.

1. Use C++ features as virtual functions, operator overloading, and exception handling.

2. Design algorithms using object-oriented techniques to solve programming problems

3. Write well styled programs using object-oriented concepts

4. Use separate compilation techniques in an Integrated Development Environment, such as Visual Studio .NET, to manage programming projects.

5. Describe and use the Standard Template Library (STL)

6. Analyze software problems and develop solutions using testing and debugging techniques

7. Use standard object-oriented programming features as encapsulation, classes, constructors and destructors, inheritance, polymorphism, and member access specifications.

[ ] B. Standard Prerequisite or Corequisite required by universities.

[ ] C. Corequisite is linked to companion lecture course.

[ ] D. Prerequisite or Corequisite is authorized by legal statute or regulation.
Code Section: _____

[ ] E. Prerequisite or Corequisite is necessary to protect the students' health and safety.

[ ] F. Computation or communication skill is needed.

[ ] G. Performance courses: Audition, portfolio, tryouts, etc. needed.

XIV. WORKPLACE PREPARATION

CS M20: Not Applicable

XV. DISTANCE LEARNING COURSE OUTLINE ADDENDUM

1. Mode of Delivery

   ☐ Online (course will be delivered 100% online)

   ☒ Online with onsite examinations (100% of the instruction will occur online, but examinations and an orientation will be scheduled onsite)

   ☒ Online/Hybrid (a percentage of instruction will be held online and the remaining percentage of instruction will be held onsite)

   ☐ Lab activities will be conducted onsite

   ☐ Televideo (Examinations and an orientation will be held onsite)

   ☐ Teleconference

   ☐ Other

2. Need/Justification

   Improve general student access.

3. Describe how instructors teaching this course will ensure regular, effective contact with and among students.

   The instructor will be available online for a minimum of 3 weekday hours/week during the day at specific times. The instructor will communicate with students through the course management system which should include both synchronous tools (such as chat) and asynchronous tools (such as email and discussions).

   Email is a tool primarily used for individual student contact. Students and the instructor can privately contact each other with questions, concerns, etc.

   Discussions is a tool which is primarily used for contacting the entire class. Students can solicit help from other students, for example. Discussions can also be graded encouraging students to participate in the class.

   Use calendar and announcement tools to keep students informed of important events, deadlines, etc.

   Collaborative Learning would involve using software which allows students and the instructor to collaborate in real-time. These sessions can also be recorded so that students who were not able to participate can also benefit from these discussions. The instructor can talk with individual students or with student groups. Students can also collaborate with each other without the instructor.

4. Describe how instructors teaching this course will involve students in active learning.

Materials: These will be available online. Students will be able to download files and view them offline. Instructor can also provide links to websites.

Quizzes: These 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.

Assignments: These may include exercises through which students explore course concepts using the textbook and additional research. There 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 submittal, if necessary.

Email: Email is a tool primarily used for individual student contact. Students and the instructor can privately contact each other with questions, concerns, etc.

Discussions: Discussions is a tool which is primarily used for contacting the entire class. Students can solicit help from other students, for example. Discussions can also be graded encouraging students to participate in the class.

Collaborative Learning: This would involve using software which allows students and the instructor to collaborate in real-time. These sessions can also be recorded so that students who were not able to participate can also benefit from these discussions. The instructor can talk with individual students or with student groups. Students can also collaborate with each other without the instructor.

Grades: Students should be able to view their current grades online at any time.

5. Explain how instructors teaching this course will provide multiple methods of content representation.

   The instructor can provide text, presentation slides, audio/visual material, programming examples, tutorials (which may be live or recorded), and links to websites.

6. Describe how instructors teaching this course will evaluate student performance.

   Through standard techniques such as exercises, programs, quizzes. and exams. The online environment can allow the exercises and programs to be iterative as students can submit their work online and get feedback from the instructor. The instructor can communicate solutions to students by posting them online. Additionally, graded discussions can be used to provide another means of assessment.

XVI. GENERAL EDUCATION COURSE OUTLINE ADDENDUM

CS M20: Not Applicable

XVII. STUDENT MATERIALS FEE ADDENDUM

CS M20: Not Applicable

XVIII. REPEATABILITY JUSTIFICATION TITLE 5, SECTION 55041

CS M20: Not Applicable

XIX. CURRICULUM APPROVAL

Course Information:

Discipline: COMPUTER SCIENCE (CS)

Discipline Code and Number: CS M20

Course Revision Category: Technical Course Revision

Course Proposed By:

Originating Faculty Esmaail Nikjeh 09/06/2018

Faculty Peer: _____

Curriculum Rep: Scarlet Relle 01/08/2019

Department Chair: Erik Reese 10/02/2018

Division Dean: Mary Rees 09/09/2018

Approved By:

Curriculum Chair: Jerry Mansfield 02/08/2019

Executive Vice President: _____

Articulation Officer: Letrisha Mai 01/29/2019

Librarian: Mary LaBarge 01/30/2019

Implementation Term and Year: _____

Approval Dates:

Approved by Moorpark College Curriculum Committee: 02/05/2019

Approved by Board of Trustees (if applicable): _____

Approved by State (if applicable): _____