CS 009: DATA STRUCTURES AND ALGORITHMS

Originator
mflora

Co-Contributor(s)

Name(s)
Kelbley, Robert
MacIntire, Doug

Justification / Rationale
Adding modalities; updating and adding textbooks; updating entrance skills, course objectives, and methods of instruction/ assessment

Effective Term
Fall 2022

Credit Status
Credit - Degree Applicable

Subject
CS - Computer Science

Course Number
009

Full Course Title
Data Structures and Algorithms

Short Title
DATA STRUCTURES

Discipline

Disciplines List
Computer Science

Modality
Face-to-Face
100% Online
Hybrid

Catalog Description
This is an advanced course in C++ programming. Students design, write, and debug C++ programs using structured programming concepts. Topics covered include pointers; linked lists, unions and data structures; bit operations, user-defined data types; recursion; incorporation of assembly language subroutines; and advanced graphical and animation techniques.

Schedule Description
This course will use an object-oriented language such as C++ to study more advanced programming principles and techniques. Students will learn about important data structures, how to create and evaluate important sorting and searching algorithms, and basic software engineering principles. Prerequisite: CS-007B and MATH-015

Lecture Units
3

Lecture Semester Hours
54

Lab Units
1

Lab Semester Hours
54

In-class Hours
108

Out-of-class Hours
108

Total Course Units
4

Total Semester Hours
216

Prerequisite Course(s)
CS 007B & MATH 015

Required Text and Other Instructional Materials

Resource Type
Book

Open Educational Resource
No

Author
Carrano, F. and Henry, T.

Title
Data Abstraction Problem Solving with C++: Wall and Mirrors

Edition
7

City
San Francisco

Publisher
Pearson

Year
2017

College Level
Yes

Flesch-Kincaid Level
12

ISBN #
9780134463971

Resource Type
Book

Open Educational Resource
No

Author
Vahid, Frank

Title
Data Structures Essentials With C++ Examples

Publisher
zyBooks

Year
2021

ISBN #
unique to each instructor

Resource Type
Book
Open Educational Resource
No

Author
Weiss, M.

Title
Data Structures and Algorithm Analysis in C++

Edition
4th

Publisher
Pearson

Year
2014

College Level
Yes

ISBN #
9780133859638

For Text greater than five years old, list rationale:
Weiss's book is used at UC Riverside.

Class Size Maximum
28

Entrance Skills
Demonstrate proficiency with using programming development environment.

Requisite Course Objectives
CS 007B-Write programs that use each of the following data structures: arrays, records, strings, linked lists, stacks, queues, and hash tables
CS 007B-Design, implement, test, and debug simple programs in an object-oriented programming language

---

Entrance Skills
Understand basics related to recursion.

Requisite Course Objectives
MATH 015-Apply the principles of mathematical induction, direct and indirect deductive methods of proof to explore integers, rational numbers,and real numbers, and their relationships (number theory).

MATH 015-Provide recursive, iterative and explicit solutions to classic discrete mathematical problems.

Entrance Skills

Write programs that use strings and queues.

Requisite Course Objectives

CS 007B-Write programs that use each of the following data structures: arrays, records, strings, linked lists, stacks, queues, and hash tables

---

Entrance Skills

Demonstrate proficiency using object-oriented programming.

Requisite Course Objectives

CS 007B-Design, implement, test, and debug simple programs in an object-oriented programming language

---

Course Content

1 Principles of Programming and Software Engineering
1.1 Software Engineering and Object-Oriented Design
1.2 Achieving a Better Solution
1.3 Key Issues in Programming
2 Recursion: The Mirrors
2.1 Recursive Solutions
2.2 Counting Things
2.3 Searching an Array
2.4 Organizing Data
2.5 Recursion and Efficiency

3 Data Abstraction: The Walls
3.1 Abstract Data Types
3.2 Specifying ADTs
3.3 Implementing ADTs

4 Linked Lists
4.1 Preliminaries
4.2 Programming with Linked Lists
4.3 Variations of the Linked List
4.4 Application: Maintaining an Inventory
4.5 The C++ Standard Template Library
5 Recursion as a Problem-Solving Technique
5.1 Backtracking
5.2 Defining Languages
5.3 The Relationship Between Recursionand Mathematical Induction
6 Stacks
6.1 The Abstract Data Type Stack
6.2 Simple Applications of the ADT Stack
6.3 Implementations of the ADT Stack
6.4 Application: Algebraic Expressions
6.5 Application: A Search Problem
6.6 The Relationship Between Stacks and Recursion

7 Queues

7.1 The Abstract Data Type Queue
7.2 Simple Applications of the ADT Queue
7.3 Implementations of the ADT Queue
7.4 A Summary of Position-Oriented ADTs
7.5 Application: Simulation

8 Advanced C++ Topics
8.1 Inheritance Revisited
8.2 Virtual Methods and Late Binding
8.3 Friends
8.4 The ADTs List and Sorted List Revisited
8.5 Class Templates
8.6 Overloaded Operators
8.7 Iterators
9 Algorithm Efficiency and Sorting
9.1 Measuring the Efficiency of Algorithms
9.2 Sorting Algorithms and Their Efficiency
10 Trees
10.1 Terminology
10.2 The ADT Binary Tree
10.3 The ADT Binary Search Tree
10.4 General Trees

11 Tables and Priority Queues
11.1 The ADT Table
11.2 The ADT Priority Queue: A Variation of the ADT Table
11.3 Tables and Priority Queues in the STL
12 Advanced Implementations of Tables
12.1 Balanced Search Trees
12.2 Hashing
12.3 Data with Multiple Organizations
13 Graphs
13.1 Terminology
13.2 Graphs as ADTs
13.3 Graph Traversals
13.4 Applications of Graphs

Lab Content
Complete programming assignments incorporating design elements and code.

Course Objectives

	Objectives
Objective 1	Compare iterative and recursive solutions for elementary problems such as factorial
Objective 2	Implement, test, and debug simple recursive functions and procedures.
Objective 3	Determine when a recursive solution is appropriate for a problem.
Objective 4	Implement the user-defined data structures in a high-level language.
Objective 5	Choose the appropriate data structure for modeling a given problem.
Objective 6	Compare alternative implementations of data structures with respect to performance.
Objective 7	Write programs that use each of the following data structures: arrays, strings, linked lists, stacks, queues, and hash tables.

Objective 8	Compare and contrast the costs and benefits of dynamic and static data structure implementations.
Objective 9	Compare and contrast the computational efficiency of the principal algorithms for sorting, searching, and hashing.

Student Learning Outcomes

	Upon satisfactory completion of this course, students will be able to:
Outcome 1	Apply a systematic approach to the design, construction and management of computer programs, emphasizing programming style, documentation, and debugging techniques.
Outcome 2	Demonstrate knowledge of data structures such as stacks, lists, trees, graphs, and queues. Implement (program) these structures in appropriate applications
Outcome 3	Analyze and implement (program) sorting and searching algorithms
Outcome 4	Utilize design principles of object-oriented programming, including encapsulation, inheritance

Methods of Instruction

Method	Please provide a description or examples of how each instructional method will be used in this course.
Collaborative/Team	Take turns role-playing as designer/tester/developer in solving programming challenges to produce software meeting prescribed input/output specification.
Lecture	Data structures and algorithms are introduced in concept and by example.
Laboratory	Students will practice developing the data structures and algorithms introduced in lecture by writing programs that solve problems of varying difficulty. Typically, students may be assigned to work either individually or in small groups to address the problem of writing code to accept input in a specific format and analyze that input produce a desired output.

Methods of Evaluation

Method	Please provide a description or examples of how each evaluation method will be used in this course.	Type of Assignment
Group activity participation/observation	Three or more major projects encompassing at least two weeks for development of complex solutions to complex tasks. Typical problems involve an assignment such as implementing data structures and algorithms using appropriate classes and objects as described in project specification.	In Class Only
Laboratory projects	These will require students to solve problems from the their lab manuals while using data structure and algorithm concepts introduced in lecture. (4 hrs/wk)	In Class Only
Mid-term and final evaluations	There will be a midterm and a final exam, generally in written format, but this may be combined with some computer work. (4 hrs)	In Class Only

Assignments

Other In-class Assignments

1. Take quizzes.
2. Take tests.
3. Participate in discussion.
4. Develop original programs to solve given problems

Other Out-of-class Assignments

1. Read the text. (2hrs/wk)
2. Write descriptions of programs in pseudocode. (0.5 hrs/wk)
3. Complete unfinished lab work. (2 hrs/wk)

Grade Methods

Letter Grade Only

Distance Education Checklist

Include the percentage of online and on-campus instruction you anticipate.

Online %

100

On-campus %

0

What will you be doing in the face-to-face sections of your course that necessitates a hybrid delivery vs a fully online delivery?

Although the course can be offered entirely online, it may also be offered hybrid to take advantage of allowing students to use on-campus technology and the ability to have collaborative activities that are more suited to in-person interaction. Examinations can be given in a controlled location.

Lab Courses

How will the lab component of your course be differentiated from the lecture component of the course?

Lab assignments involve more active learning.

From the COR list, what activities are specified as lab, and how will those be monitored by the instructor?

Lab activities are discussions and assignments that involve solving problems or exploring concepts with other students, with people not part of the course, or under the guidance of the professor or instructional support assistant. Discussions and other assignments that are completed in Canvas are monitored and evaluated by the professor. Assignments that do not take place in Canvas are evaluated by the professor based on write-ups (which may include summaries and feedback from the participants). Anonymous and non-anonymous feedback opportunities will be available to students to allow the professor further monitor effectiveness and appropriateness of activities that take place somewhere other than on the course LMS.

How will you assess the online delivery of lab activities?

Reports and other forms of write-ups will be submitted on the course LMS for evaluation and feedback.

Instructional Materials and Resources

If you use any other technologies in addition to the college LMS, what other technologies will you use and how are you ensuring student data security?

Depending on the textbook used, the professor may choose to use Pearson MyLab and Mastering, zyBooks, WebAssign, Replit, or GitHub. All of these are considered to be safe for use in education for both faculty and students. All can also be integrated with the college LMS (Canvas), which decreases the amount of times students will need to sign-in-and-out of accounts and open them up to data breaches.

If used, explain how specific materials and resources outside the LMS will be used to enhance student learning.

Professors who choose to use Pearson MyLab and Mastering, zyBooks, WebAssign, Replit, or GitHub do so in order to assign pre-written or instructor-created problems that are more complicated than those that can be created in Canvas while still receiving instantaneous feedback.

Effective Student/Faculty Contact

Which of the following methods of regular, timely, and effective student/faculty contact will be used in this course?

Within Course Management System:

Chat room/instant messaging
Discussion forums with substantive instructor participation
Online quizzes and examinations
Private messages
Regular virtual office hours
Timely feedback and return of student work as specified in the syllabus
Weekly announcements

External to Course Management System:

Direct e-mail
Posted audio/video (including YouTube, 3cmediasolutions, etc.)
Synchronous audio/video
Telephone contact/voicemail

For hybrid courses:

Scheduled Face-to-Face group or individual meetings

Briefly discuss how the selected strategies above will be used to maintain Regular Effective Contact in the course.

Faculty will regularly contact students individually and as a group through Canvas messages and/or COD email. Students will also receive regular announcements with information about the course, COD as a whole, or other relevant information.
In discussions and through other lab assignments, students will communicate with each other and their professor regularly and frequently.

If interacting with students outside the LMS, explain how additional interactions with students outside the LMS will enhance student learning.

Students may prefer to contact their professor via email or on the phone, which allows for an improved experience for those who communicate better in those contexts. The professor may direct students to access free supplemental resources as well.

Other Information

Comparable Transfer Course Information

University System

UC

Campus

UC Riverside

Course Number

CS 10C

Course Title

Introduction to Data Structures and Algorithms

Catalog Year

2021-2022

Rationale

Similar course content; currently aligned

University System

CSU

Campus

CSU Long Beach

Course Number

CECS 275

Course Title

Programming and Data Structures in C++

Catalog Year

2021-2022

Rationale

Similar course content; currently aligned

University System

CSU

Campus

CSU San Bernardino

Course Number

CSE 2020

Course Title
Computer Science II

Catalog Year
2021

Rationale
Similar course content

MIS Course Data

CIP Code
11.0701 - Computer Science.

TOP Code
070600 - Computer Science (transfer)

SAM Code
E - Non-Occupational

Basic Skills Status
Not Basic Skills

Prior College Level
Not applicable

Cooperative Work Experience
Not a Coop Course

Course Classification Status
Credit Course

Approved Special Class
Not special class

Noncredit Category
Not Applicable, Credit Course

Program Status
Program Applicable

Transfer Status
Transferable to both UC and CSU

Allow Audit
No

Repeatability
No

Materials Fee
No

Additional Fees?
No

Approvals

Curriculum Committee Approval Date
11/18/2021

Academic Senate Approval Date
12/09/2021

Board of Trustees Approval Date
01/21/2022

Chancellor's Office Approval Date
06/03/2014

Course Control Number
CCC000525441

Programs referencing this course
Liberal Arts: Business and Technology AA Degree (http://catalog.collegeofthedesert.eduundefined/?key=27)
Liberal Arts: Math and Science AA Degree (http://catalog.collegeofthedesert.eduundefined/?key=29)