CS M155: DISCRETE STRUCTURES

Originator
lalnaji

Co-Contributor(s)

Name(s)
Nikjeh, Esmaail (enikjeh)

College
Moorpark College

Discipline (CB01A)
CS - Computer Science

Course Number (CB01B)
M155

Course Title (CB02)
Discrete Structures

Banner/Short Title
Discrete Structures

Credit Type
Credit

Start Term
Spring 2021

Catalog Course Description
Introduces the discrete structures used in computer science with an emphasis on their applications. Covers functions, relations, sets, basic logic, proof techniques, basics of counting, graphs and trees, and discrete probability.

Additional Catalog Notes
Course Credit Limitations - MC, CSU and UC: CS M155 combined with MATH M21; maximum credit, one course.

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)
B - Satisfies Math/Quantitative Reasoning req (CSUGE-B B4, IGETC 2, or 4-yr)

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
(O) Student Option- Letter/Pass
(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 or CS M10A and MATH M07 or (MATH M05 and MATH M06)

Entrance Skills

Entrance Skills
MATH M05 AND MATH M6 OR MATH M07

Prerequisite Course Objectives
MATH M05-state and apply the definition of a function and use the vertical line test.
MATH M05-evaluate functions at both numerical and algebraic domain values.
MATH M05-determine the domain and range of a relation or function given its equation or its graph.
MATH M05-form a new function from original functions using the functional operations of addition, subtraction, multiplication, division, and composition.
MATH M05-use the horizontal line test and the definition of an inverse function to determine whether a pair of functions are inverses.
MATH M05-graph the functions which yield the parabola, the absolute value, the cubic, the square root, the cube root, and ones defined piecewise; solve linear and radical equations, and absolute value equalities and inequalities, and use these graphs to model real-life applications.
MATH M05-test equations of graphs for symmetries about the x-axis, the y-axis, and the origin.
MATH M05-graph a function whose equation yields a translation and/or reflection of a known graph.
MATH M05-graph a parabola given by a quadratic function.
MATH M05-give a rough sketch of the graph of a polynomial function of degree three or larger given its factored form.
MATH M05-determine the domain and range as well as the horizontal and vertical asymptotes of a rational function and use that information to graph it; be able to solve rational equations.
MATH M05-graph exponential and logarithmic functions.
MATH M05-convert equations back and forth from exponential to logarithmic form.
MATH M05-apply the rules of logarithms involving logarithms of products, quotients, powers, and change of base and solve logarithmic functions, and use logarithms to solve real-life application problems.
MATH M05-solve exponential equations which have the same base on both sides and ones that do not have the same base on both sides of the equation by using logarithms, and use exponents to solve real-life application problems.

MATH M05-use the following theorems (over the complex numbers): Remainder, Factor, Fundamental Theorem of Algebra, Rational Roots (with synthetic
division), and Conjugate Roots to solve polynomial equations.
MATH M05-solve systems of linear equations using substitution and addition (elimination) with two and three variables and determine consistency and dependency as germane.
MATH M05-solve systems of nonlinear equations and linear and non-linear systems of inequalities.
MATH M05-identify and analyze the algebraic representations of conic sections to determine their properties and sketch their graphs, including circles, ellipses and hyperbolas.
MATH M05-determine and identify terms for sequences and series, and evaluate sums for both finite and infinite series.
MATH M06-identify special triangles and their related angle and side measures.
MATH M06-evaluate the trigonometric function of an angle in degree and radian measure.
MATH M06-manipulate and simplify a trigonometric expression.
MATH M06-solve trigonometric equations, triangles, and applications.
MATH M06-graph the basic trigonometric functions and apply changes in period, phase and amplitude to generate new graphs.
MATH M06-evaluate and graph inverse trigonometric functions.
MATH M06-prove trigonometric identities.
MATH M06-convert between polar and rectangular coordinates and equations.
MATH M06-calculate powers and roots of complex numbers using DeMoivre's Theorem.
MATH M06-graph polar equations.
MATH M06-represent a vector (a quantity with magnitude and direction) in the form and ai+bj.
MATH M07-graph functions and relations in rectangular coordinates and polar coordinates.
MATH M07-analyze and identify the features of the graphs and/or the equations of functions and relations.
MATH M07-apply transformations to the graphs of functions and relations.
MATH M07-recognize the relationship between functions and their inverses graphically and algebraically.
MATH M07-solve and apply equations including rational, linear, polynomial, exponential, absolute value, radical, and logarithmic, and solve linear, nonlinear, and absolute value inequalities.
MATH M07-solve systems of equations and inequalities.
MATH M07-apply functions to model real world applications.
MATH M07-prove trigonometric identities.
MATH M07-identify special triangles and their related angle and side measures.
MATH M07-evaluate the trigonometric function at an angle whose measure is given in degrees and radians.
MATH M07-manipulate and simplify a trigonometric expression.
MATH M07-solve trigonometric equations, triangles, and their related applications.
MATH M07-graph the basic trigonometric functions and apply changes in period, amplitude, phase shift and vertical shift to generate new graphs.
MATH M07-evaluate and graph inverse trigonometric functions.
MATH M07-convert between polar and rectangular coordinates.
MATH M07-calculate powers and roots of complex numbers using DeMoivre's Theorem.
MATH M07-represent a vector (a quantity with magnitude and direction) in the form and ai+bj, compute the magnitude of a vector, and graph vectors on the xy-plane.
MATH M07-perform vector operations including addition, subtraction, scalar multiplication, and dot product. Determine the angle between two vectors and when vectors are parallel or perpendicular, and compute the projection vector.
MATH M07-write the standard form of a circle given the general equation.
MATH M07-graph plane curves described by parametric equations.
MATH M07-find parametric forms for functions in the plane and eliminate the parameter given curves in parametric form.
MATH M07-work with sequences and series or use the Binomial Theorem or determine the equations of the standard conics or perform partial fraction decomposition.

Entrance Skills

CS M10A OR CS M125

Prerequisite Course Objectives

CS M10A-describe the basic organization of a computer system.
CS M10A-describe the basic components, syntax, and semantics of the C++ programming language.
CS M10A-analyze programming problems and design algorithms to solve those problems.
CS M10A-describe and identify sequential, selection, and iteration control structures.
CS M10A-describe and apply the concepts of structured programming including function usage and parameter passing.
CS M10A-describe and apply composite data types such as arrays and structures.
CS M10A-describe and apply user defined data types such as enumerations and structured data.
CS M10A-describe, analyze, and use the C++ string class.
CS M10A-describe and apply dynamic memory allocation using pointers.
CS M10A-describe and apply file input and output.
CS M10A-describe and identify good programming practice and style.
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
MATH M05 AND MATH M06 OR MATH M07

Requisite Description
Course not in a sequence

Level of Scrutiny/Justification
Content review

Requisite Type
Prerequisite

Requisite
CS M125 OR CS M10A

Requisite Description
Course in a sequence

Level of Scrutiny/Justification
Required by 4 year institution

Student Learning Outcomes (CSLOs)

	Upon satisfactory completion of the course, students will be able to:
1	calculate probabilities of events and expectations of random variables for elementary problems.
2	use tree and graph algorithms to solve problems.
3	apply the rules of inference and methods of proof including direct and indirect proof forms, proof by contradiction, and mathematical induction.
4	apply combinatoric techniques to application problems.

Course Objectives

	Upon satisfactory completion of the course, students will be able to:
1	explain with examples the basic terminology of functions, relations and sets and perform the operations associated with functions, relations and sets.
2	relate practical examples to the appropriate set, function, or relation models and interpret the associated operations and terminology in context.
3	apply binary and n-ary relations to databases and database queries.
4	convert logical statements from informal language to propositional and predicate logic expressions.
5	apply formal methods of symbolic, propositional, and predicate logic; such as calculating the validity of formulae and computing normal forms.
6	use the rules of inference to construct proofs in propositional and predicate logic and describe how symbolic logic can be used to model real life situations or applications including those arising in computing contexts such as software analysis (e.g., program correctness), database queries, and algorithms.

7	apply formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles.
8	describe the strengths and limitations of propositional and predicate logic.
9	identify the proof technique used in a given proof and outline the basic structure of each proof technique: direct proof, proof by contradiction, and induction.
10	apply each of the proof techniques (direct proof, proof by contradiction, and induction) correctly in the construction of a sound argument and determine which type of proof is best for a given problem.
11	explain the parallels between the ideas of mathematical and/or structural induction to recursion and recursively defined structures.
12	explain the relationship between weak and strong induction and give examples of the appropriate use of each.
13	apply counting arguments, including sum and product rules, inclusion-exclusion principle and arithmetic/geometric progressions and apply the pigeonhole principle in the context of a formal proof.
14	compute permutations and combinations of a set and interpret the meaning in the context of the particular application.
15	map real world applications to appropriate counting formalisms, such as determining the number of ways to arrange people around a table subject to constraints on the seating arrangement or the number of ways to determine certain hands in cards (e.g., a full house).
16	solve a variety of basic recurrence relations and analyze problems to determine the underlying recurrence relations.
17	perform computations involving modular arithmetic.
18	illustrate by example the basic terminology of graph theory, as well as some of the properties and special cases of each type of graph or tree.
19	demonstrate different traversal methods for trees and graphs, including pre-, post-, and in-order traversal of trees.
20	model a variety of real world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system.
21	show how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting.
22	explain how to construct a spanning tree of a graph.
23	determine if two graphs are isomorphic.
24	calculate probabilities of events and expectations of random variables for elementary problems such as games of chance.
25	differentiate between dependent and independent events.
26	identify a case of the binomial distribution and compute a probability using that distribution.
27	apply the binomial theorem to independent events and Bayes' theorem to dependent events and explain how events that are independent can be conditionally dependent and vice versa.
28	apply the tools of probability to solve problems such as the average case analysis of algorithms or analyzing hashing.

Course Content

Lecture/Course Content

(15%) A. Functions, Relations and Sets

1. Functions - surjections, injections, bijections, inverses, composition
2. Relations - reflexivity, symmetry, transitivity, equivalence relations, binary and n-ary relations
3. Sets - Venn diagrams, union, intersection complements, Cartesian products, power sets, cardinality of finite sets

(20%) B. Basic Logic

1. Propositional logic
2. Logical connectives
3. Truth tables
4. Normal forms - conjunctive and disjunctive
5. Validity of well-formed formula
6. Predicate logic
7. Universal and existential quantification
8. Propositional inference rules - concepts of modus ponens and modus tollens
9. Limitations of propositional and predicate logic

(25%) C. Proof Techniques

1. Notions of implication, equivalence, converse, inverse, contrapositive, negation, and contradiction
2. The structure of mathematical proofs
3. Direct proofs

4. Proof by counterexample
5. Proof by contradiction
6. Mathematical induction
7. Structural induction, weak and strong induction
8. Recursive mathematical definitions
9. Well orderings
10. Program correctness

(12%) D. Basics of Counting

1. Counting arguments
2. Sum and product rule
3. Inclusion-exclusion principle
4. Arithmetic and geometric progressions
5. Fibonacci numbers
6. The pigeonhole principle
7. Permutations and combinations
8. Basic definitions
9. Pascal's identity
10. The binomial theorem
11. Solving recurrence relations
12. Common examples
13. The master theorem

(10%) E. Graphs and Trees

1. Trees - properties and traversal strategies
2. Undirected graphs
3. Directed graphs
4. Weighted graphs
5. Spanning trees/forests
6. Graph isomorphism

(18%) F. Discrete Probability

1. Finite probability space, events
2. Axioms of probability measures
3. Conditional probability, Bayes' theorem, independence
4. Integer random variables
5. Expectation
6. Law of large numbers
7. Variance
8. Conditional independence

Laboratory or Activity Content

n/a

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

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):

Classroom Discussion
Computational homework
Mathematical proofs
Objective exams
Projects
Problem-solving exams
Participation
Quizzes

Instructional Methodology

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

Class discussions
Distance Education
Group discussions
Instructor-guided interpretation and analysis
Lecture

Describe specific examples of the methods the instructor will use:

- Provide detailed step-by-step examples
- Assignment of application problems to computer science
- Provide practice problems to develop proper mathematical skills and techniques needed for computer science
- Provide student interaction for questions and answers
- Use projects and/or group work to enhance student understanding of the concepts

Representative Course Assignments

Writing Assignments

- short answer problems on exams such as stating the results for an application problem.
- graded assignments: in-class and/or homework assignments requiring complete solutions using both written English and symbolic mathematical language or pseudo-code.
- homework problems selected from the discrete structures textbook where answers require a written explanation of the solution such as solving a problem using the pigeonhole principle.

Critical Thinking Assignments

- mapping real world applications to appropriate counting formalisms, such as determining the number of ways to arrange people around a table subject to constraints on the seating arrangement or the number of ways to determine certain hands in cards (e.g., a full house).
- modeling a variety of real world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system.
- applying formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles.

Reading Assignments

- assigned reading material and homework problems from the discrete structures textbook.
- assigned readings of technical journal articles on discrete structures and their use in computer science.

Outside Assignments

Representative Outside Assignments

- group or individual discrete structures projects related to computer programming.
- additional problem sets provided by the instructor such as problems on logic, on discrete probability, or on the shortest path problem.

Articulation

C-ID Descriptor Number

COMP 152

Status

Approved

Equivalent Courses at 4 year institutions

University	Course ID	Course Title	Units
CSU Long Beach	CECS 228	Discrete Structures with Computing Applications	3
UC Los Angeles (UCLA)	MATH 61	Introduction to Discrete Structure	4
CSU Fresno	MATH 114	Discrete Structures	3
CSU Dominguez Hills	CSC 281	Discrete Structures	3
CSU Northridge	COMP 256/256L	Discrete Structures for Computer Science/Lab	3/1

Comparable Courses within the VCCCD

CS V17 - Discrete Structures
MATH V52 - Discrete Structures

District General Education

A. Natural Sciences

B. Social and Behavioral Sciences

C. Humanities

D. Language and Rationality

D2. Communication/Analytical Thinking
Approved

E. Health and Physical Education/Kinesiology

F. Ethnic Studies/Gender Studies

Course is CSU transferable
Yes

CSU Baccalaureate List effective term:
Fall 2016

CSU GE-Breadth

Area A: English Language Communication and Critical Thinking

Area B: Scientific Inquiry and Quantitative Reasoning

B4 Mathematical/Quantitative Reasoning
Approved

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
Approved

IGETC

Area 1: English Communication

Area 2A: Mathematical Concepts & Quantitative Reasoning

Area 2A: Mathematical Concepts & Quantitative Reasoning
Approved

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

Description
Rosen, Kenneth. *Discrete Mathematics and Its Applications.* 8th ed., McGraw-Hill, 2018.

Resource Type
Textbook

Description
Gersting, Judith L. *Mathematical Structures for Computer Science*. 7th ed., Freeman, 2015.

Resource Type
Textbook

Description
Lipschutz, Seymour, and Marc Lipson. *Schaum's Outline of Discrete Mathematics.* Rev. 3rd ed., McGraw-Hill, 2009.

Resource Type
Textbook

Description
Johnsonbaugh, Richard. *Discrete Mathematics*. 8th ed., Pearson, 2017.

Library Resources

Assignments requiring library resources
Assignments that involve the Library's print and online resources to find current articles in periodicals related to the applications of discrete structures.

Sufficient Library Resources exist
Yes

Example of Assignments Requiring Library Resources
Locate and summarize a current article related to the application of discrete structures in modeling.

Distance Education Addendum

Definitions

Distance Education Modalities
Hybrid (51%–99% online)
Hybrid (1%–50% 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 (1%—50% online) Modality:

Method of Instruction	Document typical activities or assignments for each method of instruction
Asynchronous Dialog (e.g., discussion board)	Instructor will post weekly discussion questions in Canvas. Students need to respond to these DQs by posting substantive messages in order to participate in the class.
Synchronous Dialog (e.g., online chat)	Instructor may be available on certain day(s) with certain time frames to lecture, 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 web page.

Hybrid (51%—99% online) Modality:

Method of Instruction	Document typical activities or assignments for each method of instruction
Asynchronous Dialog (e.g., discussion board)	Instructor will post weekly discussion questions in Canvas. Students need to respond to these DQs by posting substantive messages in order to participate in the class.
Synchronous Dialog (e.g., online chat)	Instructor may be available on certain day(s) with certain time frames to lecture, 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 web page.

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 weekly discussion questions in Canvas. Students need to respond to these DQs by posting substantive messages in order to participate in the class.
Synchronous Dialog (e.g., online chat)	Instructor may be available on certain day(s) with certain time frames to lecture, 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 web page.

Examinations

Hybrid (1%–50% online) Modality
Online
On campus

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

Primary Minimum Qualification
COMPUTER SCIENCE

Review and Approval Dates

Department Chair
4/28/2020

Dean
6/26/2020

Technical Review
10/1/2020

Curriculum Committee
10/6/2020

DTRW-I
MM/DD/YYYY

Curriculum Committee
MM/DD/YYYY

Board
MM/DD/YYYY

CCCCO
11/09/2020

Control Number
CCC000572319

DOE/accreditation approval date
MM/DD/YYYY