COLLEGE OF THE DESERT

Course Code MATH-015

Course Outline of Record

1. Course Code: MATH-015
2. a. Long Course Title: Discrete Mathematics for Computers
   b. Short Course Title: DISCRETE STRUCTURES
3. a. Catalog Course Description:
   This course is an introduction to discrete mathematics and its applications for computer science students. Topics to be covered include logic and sets, methods of proof, relations and functions, combinatorics, probabilities, graph and tree theory, recurrence relations, Boolean algebra, algorithms, and finite-state machines.
   b. Class Schedule Course Description:
   A study in discrete mathematical systems for computer science including methods of proof that shape the foundations of computer science. Includes propositional logic, set and number theory, Boolean Algebra, deductive and inductive proof, functions and relations, combinatorics, discrete probability, graph theory and network models, and efficiency of algorithms.
   c. Semester Cycle (*if applicable*): *N/A*
   d. Name of Approved Program(s):
   - COMPUTER SCIENCE AS Degree and Transfer Preparation
4. Total Units: 4.00 Total Semester Hrs: 108.00
   Lecture Units: 3 Semester Lecture Hrs: 54.00
   Lab Units: 1 Semester Lab Hrs: 54.00
   Class Size Maximum: 35 Allow Audit: Yes
   Repeatability No Repeats Allowed
   Justification 0
5. Prerequisite or Corequisite Courses or Advisories:
   Course with requisite(s) and/or advisory is required to complete Content Review Matrix (CCForm1-A)
   Prerequisite: MATH 012
   Prerequisite: CS 007A
   Advisory: ENG 001A
6. Textbooks, Required Reading or Software: *(List in APA or MLA format.)*
   a. Epp, Susanna S. (2011). *Discrete Mathematics with APplications* (3/e). Cengage. ISBN: 9780495391326
   College Level: Yes
   Flesch-Kincaid reading level: 12
   b. Ensley, Douglas E., J. Crawley (2006). *Discrete Mathematics: Mathematical Reasoning and Proof with Puzzles, Patterns, and Games* (1/e). Wiley. ISBN: 978-047176380
   College Level: Yes
   Flesch-Kincaid reading level: 12
   c. Hunter, David J. (2017). *Essentials of Discrete Mathematics* (3/e). Jones & Bartlett Learning. ISBN: 9781284056242
   College Level: Yes
   Flesch-Kincaid reading level: 12
   d. Rosen, Kenneth (2012). *Discrete Mathematics and its Applications* (7/e). McGraw Hill. ISBN: 978-0-07-3383
   College Level: Yes
   Flesch-Kincaid reading level: *N/A*
   e. Lipschutz, Seymour; M. Lipson (2007). *Schaum's Outline Discrete Mathematics* (3/e). McGraw Hill. ISBN: 9780071615860

College Level: Yes
Flesch-Kincaid reading level: *N/A*

7. Entrance Skills: *Before entering the course students must be able:*

a.
Proficiency with polynomial analysis including the fundamental theorem of algebra and its implications.

- MATH 012 - Analyze polynomial functions in one variable using methods such as end behavior analysis, the factor theorem, the remainder theorem, the theorem on rational zeros, Descartes' rule of signs, the intermediate value theorem, division algorithms, conjugate zeros and the fundamental theorem of algebra.

b.
Proficiency with analysis of trigonometry functions including sum to product and product to sum identities and their use in solving trigonometric equations and analysis of polar functions.

- MATH 012 - Demonstrate an understanding of a rich variety of trigonometric identities including the pythagorean identities, addition identities, the double angle identities, the half angle identities, sum to product and product to sum identities by proof and through the application of these identities to solve trigonometric equations.
- MATH 005 - Use the basic Pythagorean identities to deduce further identities.
- MATH 005 - Analyze geometrically and manipulate algebraically the equations and graphs of the standard and shifted conic sections (as derived from their geometric definitions) including the major/minor axes, foci, directrix, and asymptotes in rectangular, polar and parametric form.

c.
Proficiency with exponential and logarithmic function and their use in modeling growth functions through the properties of exponents and logarithms.

- MATH 012 - Analyze exponential and logarithmic functions by finding an exponential expression based on essential characteristics such as the growth factor and in terms of domain, concavity, intercepts, asymptotes, transformations, and by visualizing these in the construction of a graph for the function.

d.
Proficiency with a wide variety of mathematical relations including rational functions, root functions, symmetric functions and the quadratic relations used in modeling shifted conics.

- MATH 012 - Analyze rational functions in one variable by analyzing the polynomials in the numerator and denominator and interpreting these to find domain, range, intercepts, and asymptotes and visualizing these through the construction of a graph.
- MATH 005 - Analyze geometrically and manipulate algebraically the equations and graphs of the standard and shifted conic sections (as derived from their geometric definitions) including the major/minor axes, foci, directrix, and asymptotes in rectangular, polar and parametric form.
- MATH 012 - Use Polya's problem solving strategies to solve problems, with an emphasis on the algebraic method with appropriate applications of polynomial, rational, root, exponential, logarithmic, trigonometric and inverse trigonometric expressions.

e.
Solve word problems leading to systems of linear and/or non-linear equations in 2 or more variables using methods of elimination and substitution.

- MATH 040 - Solve 2x2 and 3x3 systems of linear equations.
- MATH 012 - Use Polya's problem solving strategies to solve problems, with an emphasis on the algebraic method with appropriate applications of polynomial, rational, root, exponential, logarithmic, trigonometric and inverse trigonometric expressions.

f.
Understanding of basic computer programming concepts and proficiency with introductory level programming skills.

- ENG 001A - Find, read, analyze, evaluate, interpret, and synthesize outside sources, including online information.
- CS 007A - Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions
- CS 007A - Use pseudocode or a programming language to implement, test, and debug algorithms for solving simple problems
- ENG 001A - Read, analyze, and interpret varied texts (i.e. literature, digital forms, visual).
- ENG 001A - Develop ideas coherently in writing through the drafting process.

8. Course Content and Scope:

Lecture:

1. Propositional Logic: Logic of Compound Statements
    1. Logical form and equivalence
    2. Conditional Statements
    3. Tautologies, Valid and Invalid Arguments
    4. Application: Digital Circuits and Logic Programming
2. Logic of Quantified Statements
    1. Predicates
    2. Universal and Existential Quantifiers
    3. Negation of Quantified Statements
    4. Arguments with Quantified Statements
3. Elementary Number Theory and Methods of Proof
    1. Methods of Proof: Direct and Counter-example
    2. Rational Numbers and Divisibility, Modular Arithmetic
    3. Division into Cases and the Quotient-Remainder Theorem
    4. Indirect Argument: Contradiction and Contraposition
    5. Applications: Euclidean algorithm (gcd), Division Algorithm, Infinitude of Primes, Irrationality of 2.
    6. Induction, Strong Induction, and the Well-Ordering Principle
4. Counting and Probability
    1. Counting and Possibility Trees
    2. Combinations and Permutations
    3. Pigeonhole Principle
    4. Pascal's Triangle and the Binomial Theorem
    5. Probability Axioms and Expected Value
    6. Conditional Probability, Bayes' Theorem, and Independence
5. Functions and Recursion
    1. Relations and Functions
    2. Numerical functions, Including Floor/Ceiling Functions, Functions defined on General Sets
    3. One-to-One and Onto Functions, Inverse Functions
    4. Matrices Describing Relations
    5. Recursively-Defined Sequences, Recurrence relations and Finite differences
    6. Relations on Sets, Reflexive Relations, Symmetric and Antisymmetric Relations, and Transitive Relations
    7. Equivalency Relations
    8. Partial Order Relations
    9. Applications: Finite State Machines, Modular arithmetic, Chinese Remainder Theorem, and Cryptography
6. Graphs and Trees
    1. Graphs
    2. Eulerian and Hamiltonian Paths and Circuits, Shortest Path Algorithms
    3. Chromatic and planar graphs
    4. Matrix representations of graphs
    5. Isomorphisms
    6. Trees and Spanning trees
    7. Graph Algorithms, Directed Graphs, Binary Relations, Warshall's Algorithm
    8. Huffman Codes, Aritculation Points, and Computer Networks
7. Set Theory and Boolean Algebra
    1. Basic definitions and properties

2. Countable and Uncountable Sets, Including Countability of Q
3. The Empty Set, Partitions, Power Sets
4. Boolean Algebras, Russell's Paradox, and Halting Problem

8. Efficiency of Algorithms
    1. Real-valued Functions
    2. Big-O, big-Omega, and Big-Theta Notation
    3. Exponential and Logarithmic Functions
    4. Efficiency of Algorithms

Lab: *(if the "Lab Hours" is greater than zero this is required)*

Experiment with constructing algorithms on various computer platforms.

Experiment with solving assigned problems both singly and as part of collaborative teams.

Practice answering quiz problems in a timed environment.

9. Course Student Learning Outcomes:
    1.
    Evaluate the truth and falsity of mathematical statements employing deductive and inductive proof techniques.

    2.
    Analyze the relationships among counting techniques (combinatorics), discrete probability, sets, Boolean algebra, propositional logic, and the construction of digital circuits.

    3.
    Evaluate graphs, trees, and networks in terms of efficiency, redundancy, and similiarity.

    4.
    Evaluate and prove the efficiency of computer algorithms.

10. Course Objectives: *Upon completion of this course, students will be able to:*
    a. 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).
    b. Provide recursive, iterative and explicit solutions to classic discrete mathematical problems.
    c. Illustrate functional similarities of set theory, discrete probability, propositional logic, Boolean algebra, and digital circuits.
    d. Apply graph theory and principles of combinatorial analysis to network models.
    e. Create and search Eulerian and Hamiltonian graphs.
    f. Construct and use matrices to model relations and graphs.
    g. Create and manipulate trees and specifically spanning trees to find their minimized forms.
    h. Identify and solve discrete probability and combinatorial problems.
    i. Identify and solve recurrence relations including equivalence relations and partial orderings.
    j. Compare the efficiency of common sorting and searching algorithms in terms of big-O, big-Omega, and big-Theta notation.
    k. Convert mathematical algorithms into computer programs.
    l. Use modular arithmetic to prove natural number properties and understand principles of encryption.

11. Methods of Instruction: *(Integration: Elements should validate parallel course outline elements)*
    a. Activity
    b. Collaborative/Team
    c. Discussion
    d. Laboratory
    e. Lecture
    f. Technology-based instruction

12. Assignments: *(List samples of specific activities/assignments students are expected to complete both in and outside of class.)*
    In Class Hours: 108.00
    Outside Class Hours: 108.00
    a. In-class Assignments
        1. Take Quizzes
        2. Take Tests
        3. Participate in Discussion and Lab Assignments
    b. Out-of-class Assignments
        1. Read the Textbook
        2. Finish Incomplete Lab Work
        3. Complete Daily Assigned Homework
        4. Construct Programs to Demonstrate Understanding of Material and Applications

13. Methods of Evaluating Student Progress: *The student will demonstrate proficiency by:*
    - Written homework
    - Laboratory projects
    - Computational/problem solving evaluations
    - Group activity participation/observation
    - True/false/multiple choice examinations
    - Mid-term and final evaluations
    - Student participation/contribution

14. Methods of Evaluating: Additional Assessment Information:

15. Need/Purpose/Rationale -- *All courses must meet one or more CCC missions.*
    PO-GE C4.b - Language & Rationality (Communication & Analytical Thinking)
    Gather, assess, and interpret relevant information.
    Apply logical and critical thinking to solve problems; explain conclusions; and evaluate, support, or critique the thinking of others.
    IO - Scientific Inquiry
    Analyze quantitative and qualitative information to make decisions, judgments, and pose questions.

16. Comparable Transfer Course

University System	Campus	Course Number	Course Title	Catalog Year
CSU	CSU Long Beach	MATH 163	Discrete Mathematics for Computers	2010
CSU	CSU San Bernardino	MATH 272	Discrete Mathematics for Computers	2009
UC	UC Riverside	CS 113	Discrete Mathematics for Computers	2010
UC	UC Santa Cruz	CMPE 16	Applied Discrete Structures	2009

17. Special Materials and/or Equipment Required of Students:

18. Materials Fees: ☐ Required Material?

Material or Item	Cost Per Unit	Total Cost

19. Provide Reasons for the Substantial Modifications or New Course:

    change advisory

20. a. Cross-Listed Course *(Enter Course Code)*: *N/A*
    b. Replacement Course *(Enter original Course Code)*: *N/A*

21. Grading Method *(choose one)*: Letter Grade Only

22. MIS Course Data Elements
    a. Course Control Number [CB00]: CCC000523775
    b. T.O.P. Code [CB03]: 170100.00 - Mathematics, General
    c. Credit Status [CB04]: D - Credit - Degree Applicable
    d. Course Transfer Status [CB05]: A = Transfer to UC, CSU
    e. Basic Skills Status [CB08]: 2N = Not basic skills course
    f. Vocational Status [CB09]: Not Occupational
    g. Course Classification [CB11]: Y - Credit Course
    h. Special Class Status [CB13]: N - Not Special
    i. Course CAN Code [CB14]: *N/A*
    j. Course Prior to College Level [CB21]: Y = Not Applicable
    k. Course Noncredit Category [CB22]: Y - Not Applicable
    l. Funding Agency Category [CB23]: Y = Not Applicable
    m. Program Status [CB24]: 1 = Program Applicable

    Name of Approved Program *(if program-applicable)*: COMPUTER SCIENCE
    Attach listings of Degree and/or Certificate Programs showing this course as a required or a restricted elective.)

23. Enrollment - Estimate Enrollment
    First Year: 15
    Third Year: 25

24. Resources - Faculty - Discipline and Other Qualifications:
    a. Sufficient Faculty Resources: Yes
    b. If No, list number of FTE needed to offer this course: *N/A*

25. Additional Equipment and/or Supplies Needed and Source of Funding.

    N/A

26. Additional Construction or Modification of Existing Classroom Space Needed. *(Explain:)*

    N/A

27. FOR NEW OR SUBSTANTIALLY MODIFIED COURSES
    Library and/or Learning Resources Present in the Collection are Sufficient to Meet the Need of the Students Enrolled in the Course: Yes

28. Originator John Learned Origination Date 10/20/17