C S C  3 3 3

Discrete Structures for Computer Science
Spring 2016, MW 2:30 p.m.-3:45 p.m. ED 324
Dr. Robert Strader 304e McKibben rstrader@sfasu.edu (936) 468-2508

PREREQUISITE: CSC 202; MTH 144 or MTH 233 with a C or better in each.

OFFICE HOURS: 10:50 – 11:50 a.m. MW; 1:30 – 2:30 p.m. MW.
10:00 – 11:00 a.m.; 12:30 p.m. – 1:30 p.m. TTh.
By appointment M-F.

REQUIRED MATERIALS FOR CSC 333:


OTHER MATERIALS:

Calculator – any non-programmable (“four function” plus exp and ln).

EXAMINATIONS: (67% of the course grade)

2 Class Examinations (100 points each)
Final Examination -- Comprehensive (200 points)
Note: There are no exemptions for the final examination. If you do not take the Final Exam you will receive an F in the course. If the final exam time (5/13/16 F 10:30 a.m.-12:30 p.m.) is a problem, you need to drop this course.

ASSIGNMENTS: (33% of the course grade)

Homework assignments (200 pts)
Attendance and class participation - expected

LITERATURE:
IEEE/CS Computer TK 7885 A1 I6
ACM Communications QA 76 A772
References (as indicated)

GRADING: <60 F, 60-69 D, 70-79 C, 80-89 B, >89 A; adjusted for difficulty
Attendance & Class Behavior: Roll will be taken regularly. Attendance will not be taken into consideration for your final grade. If you are absent from class, please do not come by my office and ask me to repeat the class lecture. There will be no smoking, no chewing of tobacco, no eating or drinking, no bare feet, and no wearing of hats during class. Please keep your feet off of the seat backs and seats. No disruptive behavior including offensive language will be tolerated in a computer science facility or related activity. Such behavior may result in administrative removal from class. Cell phones and pagers should be turned off for the duration of class. Surfing, texting, or other distracting behavior should not be initiated. Only students officially registered for the course and approved assistants may attend class.

Examination Policy: All class examinations are considered to be a major part of the course work upon which a large part of the course grade depends. There are NO make-up exams! Class examinations will be announced at least two classes prior to the examination. If you have a conflict with another university event, you must contact me well in advance of the examination. In case of an extreme emergency, contact me before the scheduled examination. Failure to do so may result in an examination grade of zero.

Assignment Policy: All assignments are due at the BEGINNING of class (or by an announced time) on the specified due date. That means any assignment given to me after I have collected the assignments from the class is considered to be late. There will be NO late assignments accepted without an appropriate reason (doctor's note, university event - see above, etc.). Under NO circumstances will any assignment be accepted for credit after the collected class assignments have been graded and returned. DO NOT place assignments in my box or under my office door during class. If you have a conflict, please contact me in advance. PLEASE NOTE: You may be given assignments during the last five class days of the semester.

Software Policy: Disciplinary action will be taken against individuals who perform unauthorized duplication of software or who are involved in the unauthorized use of duplicated software. Such action may make it impossible for you to successfully complete this course.

Computer Laboratory Usage: Students utilizing equipment in university computing laboratories are expected to read and abide by all posted policies for the laboratories. Please note that no children and no pets are permitted in university computing laboratories.

Drop Policy (Univ.): The date (03/23/16) is the last day to drop this course with a W.

Special Accommodation Requests: Students with special accommodation requests have the responsibility to immediately initiate a meeting with the instructor to discuss how the special accommodations will be provided. Students who are aware of these special needs at the beginning of the semester must inform the instructor in person before the twelfth class day about any class activity, which will require special accommodations.

Computer Account Policy: All assignments that require the use of the University Computer must be done under the computer account that is assigned to you in this class. You should NOT do other class assignments in this account, and you should NOT do assignments from this class in other accounts. Failure to abide by university and departmental computer account policies could mean receiving a grade of F in this course.

Cheating Policy: If in my judgment a student is found cheating on an examination, a grade of zero will be assigned as the examination grade and a minimum of one (1) letter grade will be lost in the course grade. A course grade of F may be assigned depending on the situation. A student found cheating on an examination may not drop the course. All class assignments are to be done INDEPENDENTLY unless part of a team project. If in my judgment two or more people hand in (non-team) assignments that I judge to be the same, a grade of zero may be awarded to all involved assignments and a minimum of one letter grade may be lost in the course grade. A recurrence of this by any individual will result in a grade of F in the course. Students should save all developmental copies of their programs so that individual program development can be verified by me if I think it is necessary. DO YOUR OWN WORK!!!! Do NOT show other students your code!!!
Identification: Valid student I.D. cards must be presented on each examination day. (No I.D...No exam...Grade of zero)

The following web pages contain pertinent information:

- The program learning outcomes for this course can be found at: [http://cs.sfasu.edu/cs/plo/](http://cs.sfasu.edu/cs/plo/)
- General student policies and information can be found at:
  - University Academic Integrity policy: [http://www.sfasu.edu/policies/academic_integrity.asp](http://www.sfasu.edu/policies/academic_integrity.asp)
  - University Withheld Grades policy: [http://www.sfasu.edu/policies/course-grades.pdf](http://www.sfasu.edu/policies/course-grades.pdf)
  - Students with disabilities information: [http://www.sfasu.edu/policies/academic-accommodation-for-students-with-disabilities.pdf](http://www.sfasu.edu/policies/academic-accommodation-for-students-with-disabilities.pdf)

Acceptable Student Behavior: Classroom behavior should not interfere with the instructor’s ability to conduct the class or the ability of other students to learn from the instructional program (see the Student Conduct Code, policy D-34.1). Unacceptable or disruptive behavior will not be tolerated. Students who disrupt the learning environment may be asked to leave class and may be subject to judicial, academic or other penalties. This prohibition applies to all instructional forums, including electronic, classroom, labs, discussion groups, field trips, etc. The instructor shall have full discretion over what behavior is appropriate/inappropriate in the classroom. Students who do not attend class regularly or who perform poorly on class projects/exams may be referred to the Early Alert Program. This program provides students with recommendations for resources or other assistance that is available to help SFA students succeed.  

CSC 333 - DISCRETE STRUCTURES FOR COMPUTER SCIENCE

CREDIT HOURS: 3
PREREQUISITES: CSC 202; MTH 233 or 144
GRADE REMINDER: Must have a grade of C or better in each prerequisite course.

CATALOG DESCRIPTION

Mathematical structures for describing data, algorithms, and computing machines. Theory and application of sets, relations, functions, combinatorics, matrices, graphs, and algebraic structures which are pertinent to computer science.

PURPOSE OF COURSE

To develop logical and mathematical concepts necessary to understand and analyze computational systems. Introduce concepts, techniques, and skills necessary to comprehend the underlying structure of problems encountered in designing and implementing computer systems and software. Provide the foundations for understanding computer science topics that rely upon the comprehension of formal abstract concepts.

EDUCATIONAL OBJECTIVES

Upon successful completion of the course, students should be able to:

1. Use formal notation for prepositional and predicate logic.

2. Construct formal proofs in prepositional and predicate logic and use such proofs to determine the validity of English language arguments.
3. Prove conjectures using the techniques of direct proof, proof by contraposition, proof by contradiction, and proof by induction.

4. Prove the correctness of programs that contain looping constructs.

5. Demonstrate an understanding of recursive definitions and to write recursive definitions for certain sequences and collections of objects.

6. Describe how recursive algorithms execute.

7. Use set notation and set operations to prove/disprove set identities.

8. Use the Principle of Inclusion and Exclusion to find the number of elements in the union of sets.

9. Solve permutation and combination problems for a set of \( n \) distinct objects.

10. Use relations and functions and apply these concepts to ordering problems.

11. Use graphs, directed graphs, and trees as representation tools in a wide variety of contexts.

**CONTENT**

**Hours**

Introduction to Formal Logic with applications: ..........................................................6
  - Logic as a model for computation e.g. the Prolog programming language
  - Using logic as a tool to design computer components
  - Logical expressions and circuits
  - Design and simplification of logical expressions using Karnaugh maps
  - Implicants and coverings of Karnaugh maps truth tables, tautologies, and the
    evaluation of conditional expressions in programming languages
  - The satisfiability problem and NP-completeness
  - Truth and provability
  - Inference using resolution

Introduction to Proofs and Recurrence Relations with applications:............................8
  - Repetition, recursion, induction in programs
  - Use of inductive definitions for data models
  - Use of inductive proofs for describing error-detecting codes
  - Proofs of program correctness
  - Recursive definitions, e.g. parenthesized expressions
  - Analysis of algorithms
  - Running time of programs with iterative structures, programs with procedure calls,
    programs with recursive procedure calls, conditional statements
  - Use of recurrence relations to describe running time of programs
  - Solving recurrence relations
Proving properties of programs
Properties of programs that cannot be proved (unsolvable problems)

Sets and Combinatorics with applications: ................................................................. 8
The set data model
Operations on sets: insert, delete, lookup
Enumerative algorithms and complexity classes
Algorithms for union, intersection, and difference

Relations, Functions, and Matrices with applications: ................................................. 8
Operations on functions: insert, delete, lookup
Partial orders and task system precedence constraints
Total orders and execution sequencing
Transitivity (pitfalls), symmetry, computing closures of relations
The relational data model, keys and indexes, operations
Projection and join operations
Matrices as a data structure for representing relations

Graphs and Trees with applications: ........................................................................... 8
Representations of graphs using lists and matrices
Paths and circuits in graphs
Algorithms for detecting connected components
Warshall's algorithm for reachability and transitive closure in a network
Depth-first search, network broadcast patterns
Planarity in circuit layout
Use of cycle detection in file systems and concurrent systems
Applications of graph coloring: cliques, conflict scheduling

Introduction to Algebraic Structures, Languages, and Machines with applications: ........... 4
Encoding and decoding
Pattern specification and matching
Properties of regular expressions, UNIX specification of patterns
Equivalence of regular expressions and finite state automata
Minimization of automata
Graphs to represent state machines
Automata and their programs
Nondeterminism

Exams ............................................................................................................................. 3

TOTAL 45

REFERENCES


### CSC 333 - Spring 2016

**Tentative Schedule**

<table>
<thead>
<tr>
<th>Date</th>
<th>Day</th>
<th>Topic</th>
<th>Ch/Sec.</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>M</td>
<td>MLK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>W</td>
<td>Intro. Logic, Prop. Logic</td>
<td>Ch 1/1.1/1.2</td>
<td>Hw 1 (out)</td>
</tr>
<tr>
<td>25</td>
<td>M</td>
<td>Quant., Pred., Validity, Pred. Logic</td>
<td>Ch 1/1.3/1.4</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>W</td>
<td>Prolog, Correctness</td>
<td>Ch 1/1.5/1.6</td>
<td></td>
</tr>
<tr>
<td>2/01</td>
<td>M</td>
<td>Proofs – Intro., Classification</td>
<td>Ch 2/2.1</td>
<td>Hw 1</td>
</tr>
<tr>
<td>03</td>
<td>W</td>
<td>Proofs – Induction, Iteration</td>
<td>Ch 2/2.2/2.3</td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>M</td>
<td>Number Theory</td>
<td>Ch 2/2.4</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>W</td>
<td>Algorithm Analysis, Review</td>
<td>Ch 3/3.1</td>
<td>Hw 2</td>
</tr>
<tr>
<td>15</td>
<td>M</td>
<td>Test 1</td>
<td>Ch 1,2</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>W</td>
<td>Recurrence, Analysis</td>
<td>Ch 3/3.2/3.3</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>M</td>
<td>Sets – Intro., Counting, Inclusion, Excl.</td>
<td>Ch 4/4.1-4.3</td>
<td>Hw 3</td>
</tr>
<tr>
<td>24</td>
<td>W</td>
<td>Perm./Comb., Binom. Thm., Probability</td>
<td>Ch 4/4.4-4.6</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>M</td>
<td>Relations, TopoSort</td>
<td>Ch 5/5.1-5.2</td>
<td>Hw 4</td>
</tr>
<tr>
<td>3/02</td>
<td>W</td>
<td>Databases, Functions</td>
<td>Ch 5/5.3/5.4</td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>M</td>
<td>Order, MOD, Matrices</td>
<td>Ch 5/5.5-5.7</td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>W</td>
<td>Graphs/Trees</td>
<td>Ch 6/6.1/6.2</td>
<td>Hw 5</td>
</tr>
<tr>
<td>14/16</td>
<td>MW</td>
<td>Spring Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>M</td>
<td>Decision and Huffman Trees</td>
<td>Ch 6/6.3/6.4</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>W</td>
<td>Review</td>
<td>Ch 3,4,5,6</td>
<td>- Last day for W -</td>
</tr>
<tr>
<td>28</td>
<td>M</td>
<td>Easter Break ----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>W</td>
<td>Test 2</td>
<td>Ch 3,4,5,6</td>
<td></td>
</tr>
<tr>
<td>4/04</td>
<td>M</td>
<td>Graph Algo.: Reachability, Circuits</td>
<td>Ch 7/7.1/7.2</td>
<td>Hw 6</td>
</tr>
<tr>
<td>06</td>
<td>W</td>
<td>Graph Algo.: SP, MST, DFT/BFT, Art. Pts.</td>
<td>Ch 7/7.3-7.5</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>M</td>
<td>Boolean Alg., Dig. Logic, Networks</td>
<td>Ch 8/8.1/8.2</td>
<td>Hw 7</td>
</tr>
<tr>
<td>13</td>
<td>W</td>
<td>Min., Algebraic Structures</td>
<td>Ch 8/8.3, Ch9/9.1</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>M</td>
<td>Algebraic Structures</td>
<td>Ch 9/9.1</td>
<td>Hw 8</td>
</tr>
<tr>
<td>20</td>
<td>W</td>
<td>Algebraic Structures, Coding Theory</td>
<td>Ch 9/9.1/9.2</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>M</td>
<td>Coding Theory, FSM</td>
<td>Ch 9/9.2/9.3</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>W</td>
<td>TM</td>
<td>Ch 9/9.4</td>
<td></td>
</tr>
<tr>
<td>5/02</td>
<td>M</td>
<td>Formal Languages</td>
<td>Ch 9/9.5</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>W</td>
<td>Review</td>
<td>All</td>
<td>Hw 9</td>
</tr>
<tr>
<td>13</td>
<td>F</td>
<td>Final Exam (10:30 a.m. – 12:30 p.m.)</td>
<td>Ch 1-9</td>
<td></td>
</tr>
</tbody>
</table>