Course Description
Study of algorithm design, analysis tools, and techniques for selected problems including sorting, searching, graphs, branch and bound strategies, dynamic programming, algebraic methods, string matching, and sets. An introduction to order notation, timing routines, and complexity classes.

Purpose of Course
The purpose of this course is to provide the student with tools and techniques for analyzing problem solutions. Complexity theory and computability issues are introduced. Evaluation of algorithms used in solving representative problems will be emphasized.

Program Learning Outcomes
Program learning outcomes define the knowledge, skills, and abilities students are expected to demonstrate upon completion of an academic program. These learning outcomes are regularly assessed to determine student learning and to evaluate overall program effectiveness. You may access the program learning outcomes for your major and particular courses at http://cobweb.sfasu.edu/plo.html.

Prerequisites
CSC 214 and CSC 241. Must have a grade of C or better in each prerequisite course.

Office Hours
- **Monday:** 8:30AM – 10:00AM 1:30PM – 2:30PM
- **Tuesday:** 9:15AM – 10:15AM 1:30PM – 2:30PM
- **Wednesday:** 8:30AM – 10:00AM 1:30PM – 2:30PM
- **Thursday:** 9:15AM – 10:15AM 1:30PM – 2:30PM
- **Friday:** 8:30AM – 10:00AM
Required Material

Examinations: (80% of the course grade)
- 3 Class Examinations (20 % each) (See the class schedule for the dates and exam coverage.)
- Comprehensive Final Examination (20 %) (See the class schedule for the time and date)

*Note: There are no exemptions for the final examination and no changes in taking the final examination. All students must take the final exam. A zero on the final exam will result in an F in the course. Check the final examination time. If the final examination time is a problem, you need to drop this course.

Assignments and Quizzes: (20% of the course grade)
There will be a number of homework assignments and in-class quizzes of unequal weight. Some of the homework assignments will be written assignments while some will be programming assignments.

Tentative Course Calendar
Classes begin Monday, August 30, 2010

Topics for Exam I:
- Introduction to algorithms (chapter 1)
- Fundamentals of Analysis (chapter 2)
- Brute Force (chapter 3)

Exam I

Topics for Exam II:
- Divide-and-Conquer (chapter 4)
- Decrease-and-Conquer (chapter 5)
- Transform-and-Conquer (chapter 6)

Exam II

Topics for Exam III
- Space and Time Tradeoffs (chapter 7)
- Dynamic Programming (chapter 8)
- Greedy (chapter 9)

Exam III

Remaining Topics:
- Iterative Improvement (chapter 10)
- Limitations of Algorithm Power (chapter 11)

Final Examination December 15, 10:30AM – 12:30PM
Grading Policy
Grades will be distributed according to the following guidelines:

90 - 100     A
80 - 89      B
70 - 79      C
60 - 69      D
<60          F

*Note: I hold the right to give someone a higher grade than what is listed.

Other Class Information and Policies.

Attendance: Seating Assignments may be made and roll will be taken regularly. Attendance may be taken into consideration for your final grade. If you are absent from class please make sure to get notes from a classmate. Please remember there is no smoking, no chewing of tobacco, no eating or drinking, no bare feet, and no cell phone use during class. Cell phones and other electronic communication devices must be turned off during class. Possession of a cell phone or other electronic communication device during an exam will result in an examination grade of zero. Please keep your feet off of the seat backs and seats. Inappropriate student behavior and offensive language in class, computer science facility or other related activity will not be tolerated. Do not sleep in class, I will wake you up. 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 will result in an examination grade of zero. There are no exemptions for the final examination and no changes in taking the final examination. All students must take the final exam. A zero on the final exam will result in an F in the course. Check the final examination time. If the final examination time is a problem, you need to drop this course. Once a student leaves the room on the day of an examination, they will not be permitted to return. Once the first person has left the room on the day of an examination, no one else will be permitted to begin the exam.

Assignment Policy: All assignments are due at the announced time on the specified due date. Assignments will not be accepted late. If you have a conflict, please contact me in advance. Please Note: You will be given assignments and quizzes during the last five class days of the semester. You should turn in your homework assignments done neatly, clearly, and to the best of your ability. Follow all the instructions given. You will lose points for failure to follow instructions. Any work turned in to my box should be dated and timed by the CSC department staff. Please ask nicely. Do not slide any work under my door or under the door to the Computer Science Offices.

University Drop Policy The official university add/drop policy is located at:
http://www.sfasu.edu/policies/add_drop.asp If you have questions concerning registration,
add/drop or the withdraw process, contact the Registrar at (936) 468-2501 or E-mail: REGISTRAR@SFASU.EDU The Registrar is located on the 2nd floor of the Rusk building.

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 the above statements will mean that you will receive a grade of F in this course.

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

General Student Policies:

Academic Integrity (A-9.1)
Academic integrity is a responsibility of all university faculty and students. Faculty members promote academic integrity in multiple ways including instruction on the components of academic honesty, as well as abiding by university policy on penalties for cheating and plagiarism.

If in my judgment an instance of academic dishonesty on an exam has occurred, a grade of zero will be assigned and a minimum of one (1) letter grade will be lost in the course grade. Please note that being in possession of a cell phone or other electronic device during an exam will result in an examination grade of zero. A student found cheating on an examination may not drop the course. If in my judgment a student is found cheating on any part of a homework assignment or quiz, the student will receive negative points equal to the value of the entire homework/quiz. A negative grade will not be replaced by any possible bonus assignment. I consider the person who did the work (homework, quiz, test) and the person copying the work as both cheating. Do your own work. Do not share your work with others. A course grade of F may be assigned depending on the situation.

Definition of Academic Dishonesty
Academic dishonesty includes both cheating and plagiarism. Cheating includes but is not limited to (1) using or attempting to use unauthorized materials to aid in achieving a better grade on a component of a class; (2) the falsification or invention of any information, including citations, on an assigned exercise; and/or (3) helping or attempting to help another in an act of cheating or plagiarism. Plagiarism is presenting the words or ideas of another person as if they were your own. Examples of plagiarism are (1) submitting an assignment as if it were one's own work when, in fact, it is at least partly the work of another; (2) submitting a work that has been purchased or otherwise obtained from an Internet source or another source; and (3) incorporating the words or ideas of an author into one's paper without giving the author due credit. Please read the complete policy at http://www.sfasu.edu/policies/academic_integrity.asp.

Withheld Grades Semester Grades Policy (A-54)
Ordinarily, at the discretion of the instructor of record and with the approval of the academic
chair/director, a grade of WH will be assigned only if the student cannot complete the course work because of unavoidable circumstances. Students must complete the work within one calendar year from the end of the semester in which they receive a WH, or the grade automatically becomes an F. If students register for the same course in future terms the WH will automatically become an F and will be counted as a repeated course for the purpose of computing the grade point average.

Students with Disabilities
To obtain disability related accommodations, alternate formats and/or auxiliary aids, students with disabilities must contact the Office of Disability Services (ODS), Human Services Building, and Room 325, 468-3004 / 468-1004 (TDD) as early as possible in the semester. Once verified, ODS will notify the course instructor and outline the accommodation and/or auxiliary aids to be provided. Failure to request services in a timely manner may delay your accommodations. For additional information, go to http://www.sfasu.edu/disabilityservices/.

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.

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

Identification: Valid SFA student I.D. cards with CID (not SSN) must be presented on each exam day. (No I.D...No exam...Grade of zero)

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

1. Describe the concept of an algorithm, and thereby distinguish between solvable and unsolvable problems.
2. Present various complexity-levels of algorithms, and illustrate the concept with examples of algorithms that run in polynomial time as well as some that require exponential time.
3. Apply formal analysis techniques, based on algorithm time and space requirements, to algorithms involving iteration and recursion.
4. Demonstrate the use of mathematical techniques, such as recurrence relations, as tools for analyzing the complexity of algorithms.

5. Implement and analyze the performance of algorithms for sorting, generalized searching, string matching, pattern matching, and data compression.

6. Implement branch-and-bound algorithms for solving selected NP-complete problems, and present efficient heuristic methods for finding sub-optimal but practical solutions to such problems.

7. Discuss emerging trends in algorithm developments, including parallel and distributed processing.

8. Work as part of a team.

Content

<table>
<thead>
<tr>
<th>Introduction and Math Preliminaries</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation, recursion, recurrence, induction</td>
<td>5</td>
</tr>
<tr>
<td>Order notation</td>
<td></td>
</tr>
<tr>
<td>Analysis tools and timing routines</td>
<td></td>
</tr>
<tr>
<td>Overview of complexity classes (P, NP, exponential)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sorting and Searching Algorithms</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection, insertion, exchange, special, binary, tree, hashing</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String Algorithms</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching (Knuth-Morris-Pratt, Boyer-Moore), parsing, compression</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Graph Algorithms</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Representation, connectivity, reachability, traversal, shortest path, minimum spanning tree, transitive closure, topological sort, Steiner trees, networks, depth first search, breadth first search.</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set Algorithms</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Union-find, dictionary, Boolean matrices</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algebraic Methods</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polynomials, matrix operations, random numbers</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Complexity Classes</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition, classes, examples</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Branch and Bound Algorithms</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backtracking, greedy methods, traveling salesman problem</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic Programming</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal search trees, all pairs shortest path</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Advanced Topics</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation models - computability, decidability, finite state machines, grammars,</td>
<td>4</td>
</tr>
</tbody>
</table>
pushdown automata
Parallel and Distributed algorithms
Geometric algorithms
Heuristics (including genetic and neural techniques)
Probabilistic and approximation algorithms

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

TOTAL 45

REFERENCES