CSC 342 Algorithm Analysis
Fall 2017, MWF 9:00-9:50 p.m., MTH 214

Instructor          Office Hours
Dr. Jeremy Becnel  T: 1-3
(936) 468-1582     R: 1-3
322 Math Building  by appointment
becneljj@sfasu.edu

Prerequisite: CSC 214 and CSC 241 with a C or better in each.

Materials
C/C++/Java Compiler and Development Environment.
Calculator

Exams: (70% of the course grade)
3 Class Examinations (50%)
Final Examination – Comprehensive (20%) Final Exam Schedule

Assignments: (30% of the course grade)
Several individual or group assignments will be given.

Grading Scale: <60 F, 60-69 D, 70-79 C, 80-89 B, >90 A; adjusted for difficulty

Reading: You are expected to prepare for each lecture by reading the sections expected to be covered during the lecture. See the schedule to know what to expect.

Homework: Suggested homework problems will be given. These will not be graded. However, exam questions will closely mimic problems from these exercises.

References
Analysis of the document:

**Attendance & Class Behavior:**
The policy states that attendance will not be taken into consideration for the final grade. Absentees are discouraged from visiting the professor’s office to repeat lectures. The behavior in the classroom is also regulated, with specific rules about feet on seats, disruptive behavior, and administrative removal from class. Only officially registered students are allowed to attend class.

**Examination Policy:**
Class examinations are major components of the course grade. There are no make-up exams. Exams are announced at least two classes before the exam day. In case of a conflict, students should contact the professor before the exam date. Failure to do so may result in an examination grade of zero.

**Assignment Policy:**
Assignments are due at the beginning of class or by an announced time on the specified due date. Late assignments are not accepted unless there is an appropriate reason (doctor's note, university event, etc.). Assignments cannot be submitted after the collected class assignments have been graded and returned. Students are not allowed to place assignments in the professor's box or under the office door during class.

**Software Policy:**
Unauthorized duplication or use of duplicated software will result in disciplinary action, making it impossible to complete the course.

**Computer Laboratory Usage:**
Students are expected to read and abide by all posted policies for the laboratory. Children and pets are not permitted in university computing laboratories.

**Drop Policy (Univ.):**
The last day to drop the course with a W is (10/25).

**Special Accommodation Requests:**
Students with special needs must initiate a meeting with the instructor to discuss how accommodations will be provided. Special needs must be addressed before the twelfth class day.

**Computer Account Policy:**
Assignments must be done under the assigned computer account. Students should not use the account for other class assignments.

**Cheating Policy:**
Cheating on an examination will result in a zero grade for the examination, a minimum of one letter grade may be lost in the course grade, and a grade of F may be assigned depending on the situation. Students should save all developmental copies of their programs.

Additional note:

- The word "independence" is used instead of "independently".
- "Not" is emphasized in the last sentence to discourage sharing code with others.

Conclusion:

The document outlines comprehensive guidelines for academic integrity and classroom expectations, emphasizing the importance of honesty and individual work. The policies are designed to create a fair and respectful learning environment. By following these guidelines, students can ensure a successful academic experience.
**Plagiarism:** You should do your own work (unless on a team) for all assignments. If you utilize other sources (Web, friends, etc.), you must cite the source appropriately.

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 342 - ALGORITHM ANALYSIS**

**CREDIT HOURS:** 3

**PREREQUISITES:** CSC 214 and 241

**GRADE REMINDER:** Must have a grade of C or better in each prerequisite course.

**CATALOG 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.

**EDUCATIONAL OBJECTIVES**
This course will provide students an opportunity to do the following:

- To develop the concept of an algorithm, and thereby distinguish between solvable and unsolvable problems.
- To 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.
- To apply formal analysis techniques, based on algorithm time and space requirements, to algorithms involving iteration and recursion.
- To develop the use of mathematical techniques, such as recurrence relations, as tools for analyzing the complexity of algorithms.
- To study, implement, and analyze the performance of algorithms for sorting, generalized searching, string matching, pattern matching, and data compression.
To develop and 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.

To discuss emerging trends in algorithm developments, including parallel and distributed processing.

**CONTENT**

<table>
<thead>
<tr>
<th>Introduction and Math Preliminaries</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation, recursion, recurrence, induction</td>
<td></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>
<tr>
<td>Sort and Searching Algorithms</td>
<td>5</td>
</tr>
<tr>
<td>Selection, insertion, exchange, special, binary, tree, hashing</td>
<td></td>
</tr>
<tr>
<td>String Algorithms</td>
<td>3</td>
</tr>
<tr>
<td>Matching (Knuth-Morris-Pratt, Boyer-Moore), parsing, compression</td>
<td></td>
</tr>
<tr>
<td>Graph Algorithms</td>
<td>9</td>
</tr>
<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></td>
</tr>
<tr>
<td>Set Algorithms</td>
<td>3</td>
</tr>
<tr>
<td>Union-find, dictionary, Boolean matrices</td>
<td></td>
</tr>
<tr>
<td>Algebraic Methods</td>
<td>3</td>
</tr>
<tr>
<td>Polynomials, matrix operations, random numbers</td>
<td></td>
</tr>
<tr>
<td>Complexity Classes</td>
<td>4</td>
</tr>
<tr>
<td>Definition, classes, examples</td>
<td></td>
</tr>
<tr>
<td>Branch and Bound Algorithms</td>
<td>3</td>
</tr>
<tr>
<td>Backtracking, greedy methods, traveling salesman problem</td>
<td></td>
</tr>
<tr>
<td>Dynamic Programming</td>
<td>3</td>
</tr>
<tr>
<td>Optimal search trees, all pairs shortest path</td>
<td></td>
</tr>
<tr>
<td>Advanced Topics</td>
<td>4</td>
</tr>
<tr>
<td>Computation models - computability, decidability, finite state machines, grammars, pushdown automata</td>
<td></td>
</tr>
<tr>
<td>Iterative Refinement</td>
<td></td>
</tr>
<tr>
<td>Parallel and Distributed algorithms</td>
<td></td>
</tr>
<tr>
<td>Geometric algorithms</td>
<td></td>
</tr>
<tr>
<td>Heuristics (including genetic and neural techniques)</td>
<td></td>
</tr>
<tr>
<td>Probabilistic and approximation algorithms</td>
<td></td>
</tr>
<tr>
<td>Exams</td>
<td>3</td>
</tr>
<tr>
<td>TOTAL</td>
<td>45</td>
</tr>
</tbody>
</table>