# The University of Texas at Tyler
## Bachelor of Science in Computer Science

### Syllabus

<table>
<thead>
<tr>
<th>Course Number:</th>
<th>COSC 3325</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course Title:</td>
<td>Algorithm Design and Analysis</td>
</tr>
</tbody>
</table>

**Course Description:**
Introduction to formal techniques used to support the design and analysis of algorithms, focusing on both the underlying mathematical theory and practical considerations of efficiency. Topics include asymptotic complexity bounds, techniques of analysis, algorithmic strategies, basic computability, and complexity classes.

<table>
<thead>
<tr>
<th>Pre-requisites:</th>
<th>COSC 2336</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credits:</td>
<td>3</td>
</tr>
</tbody>
</table>
|                 | Anany Levitin  
|                 | Addison Wesley/Pearson Education, Inc. |
| Languages Used: | C, C++, Java |

<table>
<thead>
<tr>
<th>(if applicable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTLINE</td>
</tr>
<tr>
<td>HOURS</td>
</tr>
<tr>
<td>1. Proof techniques</td>
</tr>
<tr>
<td>2. Graphs and trees</td>
</tr>
<tr>
<td>3. Algorithms and problem-solving</td>
</tr>
<tr>
<td>4. Fundamental data structures</td>
</tr>
<tr>
<td>6. Basic algorithmic analysis</td>
</tr>
<tr>
<td>7. Algorithmic strategies</td>
</tr>
<tr>
<td>8. Fundamental computing algorithms</td>
</tr>
<tr>
<td>9. Basic computability</td>
</tr>
<tr>
<td>10. The complexity classes P and NP</td>
</tr>
<tr>
<td>TOTAL</td>
</tr>
</tbody>
</table>

**Additional Materials:**

**Evaluation Method:** (only items in dark print apply)

1. Examination/Quiz  
2. Homework  
3. Paper/Report  
4. Computer Program  
5. Project  
6. Presentation  
7. Class Participation  
8. Peer Review

**Course Objectives:** By the end of this course students are expected to:

1. Outline the basic structure of and give examples of proof techniques. [1,2,7]  
2. Discuss which type of proof is best for a given problem. [1,7]  
3. Relate the ideas of mathematical induction to recursion and recursively defined structures. [1,2,7]
4. Identify the difference between mathematical and strong induction and give examples of the appropriate use of each. [1,2,7]

5. Relate graphs and trees to data structures, algorithms, and counting. [1,2,7]

6. Explain the use of big O, omega, and theta notation to describe the amount of work done by an algorithm. [1,2,7]

7. Use big O, omega, and theta notation to give asymptotic upper, lower, and tight bounds on time and space complexity of algorithms. [1,2,7]

8. Determine the time and space complexity of simple algorithms. [1,2,7]

9. Deduce recurrence relations that describe the time complexity of recursively defined algorithms. [1,2,7]

10. Solve elementary recurrence relations. [1,2,7]

11. Describe the shortcoming of brute-force algorithms. [1,7]

12. For each of several kinds of algorithm (brute force, greedy, divide-and-conquer, backtracking, branch-and-bound, and heuristic), identify an example of everyday human behavior that exemplifies the basic concept. [1,7]

13. Implement a greedy algorithm to solve an appropriate problem. [1,2,4,7]

14. Implement a divide-and-conquer algorithm to solve an appropriate problem. [1,2,4,7]

15. Use backtracking to solve a problem such as navigating a maze. [1,2,4,7]

16. Describe various heuristic problem-solving methods. [1,7]

17. Use pattern matching to analyze substrings. [1,2,7]

18. Use numerical approximation to solve mathematical problems, such as finding the roots of a polynomial. [1,2,7]

19. Implement the most common quadratic and O(N log N) sorting algorithms. [1,2,4,7]

20. Design and implement an appropriate hashing function for an application. [1,2,4,7]

21. Design and implement a collision-resolution algorithm for a hash table. [1,2,4,7]

22. Discuss the computational efficiency of the principal algorithms for sorting, searching, and hashing. [1,7]

23. Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data. [1,7]

24. Solve problems using the fundamental graph algorithms, including depth-first and breadth-first search, single-source and all-pairs shortest paths, transitive closure, topological sort, and at least one minimum spanning tree algorithm. [1,2,7]

25. Demonstrate the following capabilities: to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in programming context. [1,2,4,7]

26. Explain how some problems have no algorithmic solution. [1,2,7]

27. Provide examples that illustrate the concept of uncomputability. [1,2,7]

28. Define the classes P and NP. [1,2,7]

29. Explain the significance of NP-completeness. [1,2,7]

30. Prove that a problem is NP-complete by reducing a classic known NP-complete problem to it. [1,2,7]

31. Prove that a language is in a specified class and that it is not in the next lower class. [1,2,7]

1 Numbers in brackets refer to method(s) used to evaluate the course objective.
Relationship to Program Outcomes: (only items in dark print apply)†

This course supports the following Computer Science Program Outcomes, which state that our students at the time of graduation are expected to:

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Posses knowledge of the fundamentals of mathematics, science, and technology. [1-31]</td>
</tr>
<tr>
<td>2.</td>
<td>Be able to use modern computational tools and techniques in the practice of computer science. [1-9,13-15,17-21,24-31]</td>
</tr>
<tr>
<td>3.</td>
<td>Be able to develop logically sound and efficient algorithms. [13-15,19-21,25]</td>
</tr>
<tr>
<td>4.</td>
<td>Be prepared to implement algorithms in multiple programming languages, on multiple hardware platforms, and in multiple operating system environments. [13-15,19-21,25]</td>
</tr>
<tr>
<td>5.</td>
<td>Be able to perform analysis, design, implementation, testing, and maintenance of computer-based systems, stressing software engineering principles.</td>
</tr>
<tr>
<td>6.</td>
<td>Be prepared to seek continuing professional development, graduate studies, or professional certifications related to computer science.</td>
</tr>
<tr>
<td>7.</td>
<td>Demonstrate effective written, visual and oral communication skills. [1-31]</td>
</tr>
</tbody>
</table>
| 8.     | Posses an educational background to understand the global context in which computer science is practiced, including:  
  a. Knowledge of contemporary issues related to computer science;  
  b. The impact of computers on society;  
  c. The role of ethics in the practice of computer science. |
| 9.     | Be able to contribute effectively as members of a project development team. |
| 10.    | Recognize the need to pursue continued learning throughout their professional careers. |

†Numbers in brackets refer to course objective(s) that address the Program Outcome.

Prepared By: Stephen B. Rainwater  
Date: 8/18/06  
Revised: 8/21/07
COSC 3325.001  Algorithm Design and Analysis

Course Description:  Introduction to formal techniques in the design and analysis of algorithms, focusing on both the underlying mathematical theory and practical considerations of efficiency. Topics include asymptotic complexity bounds, techniques of analysis, algorithmic strategies, basic computability, and complexity classes.

Course Prerequisites:  Data Structures and Algorithms (COSC 2336)

Fall, 2010  
August 26 - December 14  
TR  8:00 AM - 9:15 PM  RBN 3040

Instructor:  Dr. Stephen B. Rainwater  
Office:  RBN 3002  
PHONE: (903)566-7235  FAX: (903)566-5607  
E-MAIL: srainwater@uttyler.edu

Office Hours:  TWR  9:30 AM - 10:30 AM  
Others by appointment only

Anany Levitin  
Addison-Wesley/Pearson Education, Inc.

Grade Determination:  
Course Homework and Lab Assignments  35%  
Examination I  20%  
Examination II  20%  
Final Examination (Comprehensive)  25%

Grading Scale  
A = 100% - 90%  
B = 89% - 80%  
C = 79% - 70%  
D = 69% - 60%  
F = 59% - 0%

Course Outline:  
1. Introduction to Algorithm Analysis  
2. Fundamentals of Analysis of Algorithm Efficiency  
3. Analysis of Brute Force Algorithms  
4. Analysis of Divide-and-Conquer Algorithms  
5. Analysis of Decrease-and-Conquer Algorithms  
6. Analysis of Transform-and-Conquer Algorithms  
7. Space and Time Tradeoffs  
8. Dynamic Programming  
9. Greedy Technique  
10. Iterative Improvement  
11. Limitations of Algorithm Power  
12. Coping with Limitations of Algorithm Power

Important Dates:  
October 29 - Last Day to Withdraw From Course!  
November 25 - No Class - Thanksgiving Holiday
Academic dishonesty: You are expected to do your own work. You may assist each other with general concepts, but direct assistance with a particular assignment or any attempts to gain an unfair academic advantage will not be tolerated. Cheating is considered a serious academic offense both by the department and the University. It may result in a failing grade from this course for all parties involved. If you have questions about the line between assistance and cheating, discuss it with your instructor. The instructor reserves the right to ask you to explain any assignment that you turn in to judge if the work is actually yours.

Disabilities: If you have a disability, including a learning disability, for which you request an accommodation, please contact Ida MacDonald in the Disability Support Services office so that the appropriate arrangements may be made. In accordance with federal law, a student requesting accommodation must provide documentation of his/her disability to the Disability Support Services counselor. For more information, call or visit the Student Services Center located in the University Center, Room 282. The telephone number is 566-7079 (TDD 565-5579).

Writing Center: Located in BUS 202, the UT-Tyler Writing Center provides professional writing tutoring for all students. If you wish to use the Writing Center, you should plan in advance for a minimum of two hour-long tutorials per assignment: the first to assess your needs, and the second to follow up. Be prepared to take an active role in your learning, as you will be asked to discuss your work. While Writing Center tutors are happy to give constructive criticism and teach effective writing techniques, they will under no circumstances write your paper for you. Appointments are strongly encouraged: call 903-565-5995. Visit www.uttyler.edu/writingcenter

Other policies related to your course enrollment can be found at:

http://www.uttyler.edu/academicaffairs/syllabuspolicies.pdf
Students Rights and Responsibilities
To know and understand the policies that affect your rights and responsibilities as a student at UT Tyler, please follow this link: http://www.utttyler.edu/wellness/StudentRightsandResponsibilities.html

Grade Replacement/Forgiveness
If you are repeating this course for a grade replacement, you must file an intent to receive grade forgiveness with the registrar by the 12th day of class. Failure to do so will result in both the original and repeated grade being used to calculate your overall grade point average. Undergraduates will receive grade forgiveness (grade replacement) for only three course repeats; graduates, for two course repeats during his/her career at UT Tyler.

State-Mandated Course Drop Policy
Texas law prohibits a student who began college for the first time in Fall 2007 or thereafter from dropping more than six courses during their entire undergraduate career. This includes courses dropped at another 2-year or 4-year Texas public college or university. For purposes of this rule, a dropped course is any course that is dropped after the 12th day of class (See Schedule of Classes for the specific date).

Exceptions to the 6-drop rule may be found in the catalog. Petitions for exemptions must be submitted to the Registrar's Office and must be accompanied by documentation of the extenuating circumstance. Please contact the Registrar's Office if you have any questions.

Disability Services
In accordance with federal law, a student requesting accommodation must provide documentation of his/her disability to the Disability Support Services counselor. If you have a disability, including a learning disability, for which you request an accommodation, please contact Ida MacDonald in the Disability Support Services office in UC 282, or call (903) 566-7079.

Student Absence due to Religious Observance
Students who anticipate being absent from class due to a religious observance are requested to inform the instructor of such absences by the second class meeting of the semester.

Student Absence for University-Sponsored Events and Activities
If you intend to be absent for a university-sponsored event or activity, you (or the event sponsor) must notify the instructor at least two weeks prior to the date of the planned absence. At that time the instructor will set a date and time when make-up assignments will be completed.

Social Security and FERPA Statement:
It is the policy of The University of Texas at Tyler to protect the confidential nature of social security numbers. The University has changed its computer programming so that all students have an identification number. The electronic transmission of grades (e.g., via e-mail) risks violation of the Family Educational Rights and Privacy Act; grades will not be transmitted electronically.

Emergency Exits and Evacuation:
Everyone is required to exit the building when a fire alarm goes off. Follow your instructor's directions regarding the appropriate exit. If you require assistance during an evacuation, inform your instructor in the first week of class. Do Not re-enter the building unless given permission by University Police, Fire department, or Fire Prevention Services.

Revised June 2010