

CS 2401 Elementary Data Structures and Algorithms

Fall 2014 Syllabus

Instructor:

Mahmud Shahriar Hossain

Email: mhossain@utep.edu

Class hours: Monday and Wednesday 10:30 am - 11:50 am

Class room: Chemistry Computer Sci Bldg G.0208

Website: <http://www.cs.utep.edu/mhossain/>

Office: CCSB 3.0504

Office hours: MW 1:00-2:30 and by appointment

Lab Sections:

You should be enrolled in one of the following lab sections. You must attend the lecture section and a lab section that you are enrolled in.

3:00 pm - 4:20 pm MW Chemistry Computer Sci Bldg 1.0410

9:00 am - 10:20 am TR Chemistry Computer Sci Bldg 1.0410

10:30 am - 11:50 am TR Chemistry Computer Sci Bldg 1.0410

Teaching Assistants (TAs):

Lauren Waldrop (lewaldrop@miners.utep.edu)

Office location: Chemistry and Computer Sci Bldg G.0512

Office hours:

Monday: 1:30PM-3:00PM

Wednesday: 1:30PM-3:00PM

Thursday: 3:00PM-5PM

Balakrishna Dasam (bdasam@miners.utep.edu)

Office location: Chemistry and Computer Sci Bldg G.0512

Office hours:

Monday: 9:00AM-10:30AM

Wednesday: 9:00AM-10:30AM

Final Exam:

- Friday, Dec 12th 10:00 am – 12:45 pm (in the regular classroom: Computer Sci Bldg G.0208)

NOTE: The next computer science class CS 2302 (Data Structures) has two prerequisites: CS2401 and MATH 2300 (Discrete Math). To avoid delaying your progress through the program, if you are not currently taking MATH 2300 and have not already passed it, you should seriously consider adjusting your schedule.

Course Objectives: This is the second course for students majoring in Computer Science. Students will learn about fundamental computing algorithms, including searching and sorting; recursion; elementary abstract data types including linked lists, stacks, queues and trees; and elementary algorithm analysis. Prerequisite: CS 1401, with a grade of C or better.

Knowledge and Abilities Required Before the Students Enter the Course: Students are assumed to be comfortable programming in Java. Students should be able to code basic arithmetic expressions, define simple classes, use strings, code loops and conditional statements, write methods, create objects from classes, invoke methods on an object, perform basic text file input and output, and use arrays.

Learning Outcomes

Level 3: Synthesis and Evaluation:

Level 3 outcomes are those in which the student can apply the material in new situations. This is the highest level of mastery. On successful completion of this course, students will be able to identify, implement and use the following data structures as appropriate for a given problem:

1. Design and implement solutions to computational problems using the following data structures:
 - a. multi-dimensional arrays;
 - b. lists implemented as arrays or linked lists;
 - c. stacks;
 - d. queues;
 - e. binary trees and binary search trees.

Level 2: Application and Analysis:

Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details. Upon successful completion of this course, students will be able to:

1. Describe, implement, and use the following concepts:
 - a. classes, subclasses, and inheritance
 - b. encapsulation and information hiding
2. Describe, implement, and use the following algorithms:
 - a. sequential and binary search
 - b. quadratic and $O(n \log n)$ sorting
 - c. string manipulation and parsing
3. Describe and trace computer representation and memory allocation of:
 - a. integers, real numbers, arrays and objects
 - b. methods, including recursive methods and the use of activation records
4. Use basic notions of algorithm complexity:
 - a. use Big-O notation to express the best-, average- and worst-case behaviors of an algorithm

- b. determine the best, average and worst-case behaviors of a simple algorithm
 - c. assess basic time and space trade-offs in algorithms
5. Use recursion and iteration as problem solving techniques

Level 1: Knowledge and Comprehension

Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. On successful completion of this course, students will be able to:

1. Explain the concept of polymorphism

Textbook: Reading and laboratory assignments will be drawn from *Introduction to Java Programming*, 9th edition, COMPREHENSIVE VERSION, by Y. Daniel Liang. If necessary, this book is available at the bookstore and through major online book retailers, and you are expected to acquire a copy for your use in this course. Photocopied textbooks are illegal and their use will not be tolerated.

Supplemental Materials: The instructor will provide the students with OpenDSA (<http://algoviz.org/OpenDSA/>) access as an additional resource. A number of reading assignments and practices will be based on OpenDSA.

Software: Software used in this course will be available in the instructional labs. For those who wish to use the course software on your home computer, instructions will be given in the labs.

Assignments: Reading and homework assignments will be handed out or announced in class and in labs. If you miss a class or lab session, it is your responsibility to find out what you missed. You should expect to spend at least 10-12 hours/week outside of class and lab on reading and homework.

Grading: Your semester grade will be based on a combination of homework and lab assignments, weekly quizzes, lab attendance, exams, and a final exam. The approximate percentages are as follows:

- 30% Homework and lab assignments
- 5% Lecture attendance, lab attendance and activities
- 10% In-class exercises and quizzes
- 30% Midterm Exams (3 exams, 10% each)
- 25% Final comprehensive exam

Each of these is explained in more detail below.

Homework and Lab Assignments: Homework and lab assignments are designed to allow you to practice using the concepts presented in lecture and in your reading. Homework and lab assignments may include written problems, tutorial exercises, and

programming problems. In addition to turning in your code, you will be required to demonstrate lab assignments to your TA.

Homework must be done individually unless specifically instructed to work in groups. While you may discuss the problem in general terms with other people, your answers and your code should be written and tested by you alone. **Do not exchange programs or let someone look at your code, even "just so they can see how you did it."** If you need help, consult a TA or the professor.

Laboratory Sessions: Laboratory sessions are designed to give you guidance in getting your homework assignment started well. In a typical lab session, the Teaching Assistant will present additional material that will help you complete the assignment and answer your questions as you begin work.

You are required to sign up for and attend one of the lab sections associated with this course. Attendance will be taken. To earn full credit for attendance, you must show up on time, stay for the entire session, and work only on your assignment. You may be excused from lab with full credit if your work has been completed and turned in.

In addition to turning in your lab assignment, you must schedule a one-on-one lab demo session with your TA in which you will explain how your program works and he/she will ask questions to test your understanding of the program being submitted. The TA will then assign a grade based on this session. Your assignment is not completed and will not receive a grade until the demo is completed.

Late Policy: Lab assignments up to one week late will receive up to 75% percent of full credit. In extraordinary circumstances, you may be allowed to turn in assignments that are more than one week late for a maximum of 50% credit. However, to turn in an assignment more than one week late you **MUST** discuss the issue with the instructor to receive permission to turn it in more than one week late.

EVEN IF YOUR LAB IS NOT COMPLETELY WORKING YOU SHOULD CONSIDER TURNING IN WHAT YOU HAVE FOR PARTIAL CREDIT.

You must score 60% or better on the lab assignments to pass this course.

Quizzes: The purpose of a quiz is to ensure that you have read the weekly reading assignment and to verify that you have mastered the major concepts of recent lectures. Quizzes typically will be about 10-15 minutes in length and will cover the material assigned to be read for the upcoming lecture plus selected concepts from previous lectures. There will be no make-up on missed quizzes.

Exams: The purpose of the exams is to allow you to demonstrate mastery of course concepts. Each exam will focus on the material from the previous three or four weeks. Exams will take place during the regular class session. There will be three exams, each contributing 10% to your final grade, or 30% total to the overall course grade.

Because the exams contribute so heavily to your total grade, it is vital that you do well on them.

Exams are planned for the following dates (subject to change):

1. Monday, Sep 22 or Wednesday Sep 24
2. Monday, Oct 27 or Wednesday Oct 29
3. Monday, Nov 24 or Wednesday Nov 26

Make-up exams will be given only in extremely unusual circumstances. If you must miss an exam, please meet with an instructor, BEFORE the exam if at all possible.

Final Exam: The final exam will be comprehensive and will count 25% toward your course grade. If you cannot demonstrate your understanding of the concepts and skills needed for success in CS 2402, you cannot pass this course even if you have enough points to do so.

If you have a scheduling conflict or if you are scheduled for three final exams in one day, see the instructor IN ADVANCE.

Grading: The nominal percentage-score-to-letter-grade conversion is as follows:

- 90% or higher is an A
- 80-89% is a B
- 70-79% is a C
- 60-69% is a D
- below 60% is an F

These threshold downward in some cases, e.g., so that 88% or higher represents an A, but the thresholds will not be adjusted upward. You must earn a C or better to continue to the next course in this sequence.

Standards of Conduct: You are expected to conduct yourself in a professional and courteous manner, as prescribed by the UTEP Standards of Conduct (<http://sa.utep.edu/osccr/student-conduct/> , <http://studentaffairs.utep.edu>).

Graded work, e.g., homework and tests, is to be completed independently and should be unmistakably your own work (or, in the case of group work, your team's work), although you may discuss your project with other students in a general way. You may not represent as your own work material that is transcribed or copied from another person, book, or any other source, e.g., a web page.

Academic dishonesty includes but is not limited to cheating, plagiarism and collusion.

- Cheating may involve copying from or providing information to another student, possessing unauthorized materials during a test, or falsifying data (for example program outputs) in laboratory reports.
- Plagiarism occurs when someone represents the work or ideas of another person as his/her own.
- Collusion involves collaborating with another person to commit an academically dishonest act.

Professors are required to -- and will -- report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students (<http://sa.utep.edu/dean/>).

Disabilities:

If you have a disability and need classroom accommodations, please contact The Center for Accommodations and Support Services (CASS) at 747-5148, or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at www.sa.utep.edu/cass. *CASS' Staffs are the only individuals who can validate and if needed, authorize accommodations for students with disabilities.*

Weekly Schedule: (tentative and subject to change) When learning a new topic, often, we will see how it relates to the previously learned topics and thus, we will use this as an opportunity to recall the previous topics.

1. **Aug 25 - Aug 31:** Course overview, methods and arrays
Reading: Chapter 5 and 6 (skip 6.11)
2. **Sep 1-7:** Multi-dimensional arrays, search
Reading: Chapter 7
3. **Sep 8-14:** Classes, objects and references
Reading: Chapter 8
4. **Sep 15-21:** Linked Lists
Reading: none
5. **Sep 22-28:** Review, MIDTERM EXAM 1
Reading: none
6. **Sep 29 - Oct 5:** Recursion
Reading: Chapter 20
7. **Oct 6-12:** Recursion
Reading: Chapter 20
8. **Oct 13-19:** Complexity analysis, big-oh notation
Reading: Sections 24.1, 24.2, 24.3, 24.4.1, 24.4.6, and 24.7
9. **Oct 20-26:** Sorting
Reading: Sections 6.11, Sections 25.1-25.4
10. **Oct 27 - Nov 2:** Review, MIDTERM EXAM 2
Reading: none
11. **Nov 3-9:** Abstract data types, generics, Lists
Reading: Chapter 15, Sections 26.4

12. **Nov 10-16:** Stacks and Queues
Reading: Sections 26.5 and 22.10
13. **Nov 17-23:** Trees, Binary Search Trees, AVL trees
Reading: Section 27.1-3, Chapter 29
14. **Nov 24 - Nov 30:** Review, MIDTERM EXAM 3 (Thanksgiving)
Reading: none
15. **Dec 1-7:** Heaps and heapsort, review for final
Reading: Section 25.5