CS 2401 Elementary Data Structures and Algorithms
Fall 2018 Syllabus

Class information of two lecture sections:
    CRN: 13629
    Class time: Mondays and Wednesdays 10:30 am-11:50 am
    Class location: PSYC 308

    CRN: 15766
    Class time: Mondays and Wednesdays 12:00 pm-1:20 pm
    Class location: PSYC 308

Instructor information:
    Instructor: Dr. Mahmud Shahriar Hossain
    Email: mhossain@utep.edu
    Office location: CCSB 3.0504
    Office hours: MW 1:45 PM – 2:45 PM and by appointment.

Teaching Assistants (TAs):
    Teaching Assistant: Ismael Villanueva Miranda
    Email: ivillanueva5@miners.utep.edu
    Office hours: Monday and Wednesday: 9 am - 10:30 am
                Tuesday: 9:30 am - 10:30 am.

    Teaching Assistant: Mahdokht Afravi
    Email: mafravi@miners.utep.edu
    Office location: TBA
    Office hours: Tuesdays, 9:30a-10:20a
                Wednesdays, 10a-10:50a
                Thursdays, 9:30a-10:20a
                Fridays, 10a-10:50a

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.

CRN 15767: MW 7:30 am - 8:50 am   Chemistry Computer Sci Bldg 1.0410
CRN 15849: MW 4:30 pm - 5:50 pm   Chemistry Computer Sci Bldg 1.0704
CRN 11556: MW 3:00 pm - 4:20 pm   Chemistry Computer Sci Bldg 1.0410
CRN 11557: TR 10:30 am - 11:50 am   Chemistry Computer Sci Bldg 1.0410

Final Exams:
    CRN 13629 (Morning section): TBA

CRN 15766 (Afternoon section): TBA

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 1301 and CS 1101, with a grade of C or better in both.

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*, 10ᵗʰ 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.

Software: Software used in this course will be available in the instructional labs. Students can install the necessary software in their home machines or laptops. Necessary software packages should be publicly available.

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 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. Typically, the lab assignments must be demonstrated within one week of submission. The submitted version of the work will be used for the demonstration and grading.

Homework and lab assignments 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, not even "just so they can see how you did it."** If you need help, consult a TA or the professor. **Taking pictures of monitors of other students (inside or outside labs) and loud discussions during lab and lecture sessions are strictly prohibited.**

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.

Late Policy: Lab assignments up to one week late will receive up to 65% percent of full credit (5% deduction per late day). 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 SOLUTION IS NOT COMPLETELY WORKING YOU SHOULD CONSIDER TURNING IN WHAT YOU HAVE FOR PARTIAL CREDIT.

You must score 65% or better on the lab assignments to pass this course.

Quizzes/In-class exercises: 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. In-class 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 some online quizzes, which may take 30 minutes to an hour. There will be no make-up on missed quizzes.

Midterm Exams: The purpose of the midterm exams is to allow you to demonstrate mastery of the 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 midterm exams, each contributing 10% to your final grade, or 30% total to the overall course grade. Because the midterm exams contribute so heavily to your total grade, it is vital that you do well on them.

Midterm exams are planned for the following weeks (subject to change):
1. MIDTERM EXAM 1: In Week 5
2. MIDTERM EXAM 2: In Week 10
3. MIDTERM EXAM 3: In Week 14

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

Final Exam: The final exam will be comprehensive and will count 25% toward your course grade.

If you have a scheduling conflict or if you are scheduled for more than 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 thresholds are adjusted 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:
In the classroom and in all academic activities, students are expected to uphold the highest standards of academic integrity. Any form of scholastic dishonesty is an affront to the pursuit of knowledge and jeopardizes the quality of the degree awarded to all graduates of UTEP.

Any student who commits an act of scholastic dishonesty is subject to discipline. Scholastic dishonesty includes, but is not limited to, cheating, plagiarism, the submission for credit of any work or any materials that are attributable in whole or in part to another person, taking an examination for another person, an act designed to give unfair advantage to a student or the attempt to commit such acts. Proven violations of the detailed regulations, as printed in the Handbook of Operating Procedures may result in sanctions ranging from disciplinary probation to a failing grade in the course, to suspension or dismissal, among others. The Handbook of Operating Procedures: Student Conduct and Discipline can be accessed at the following link: http://admin.utep.edu/Default.aspx?tabid=73922.

Disabilities:
If you have a disability and need classroom accommodations, please contact The Center for Accommodations and Support Services (CASS) at 915-747-5148, 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.*

Use of cellphones and headphones is strictly prohibited in the class. Use of such devices in the classroom and labs must be approved by CASS.

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. **Week 1:** Course overview, methods and arrays
   Reading: Chapter 6 and 7 (skip 7.11 for now)
2. **Week 2:** Multi-dimensional arrays, search
   Reading: Chapter 8
3. **Week 3:** Classes, objects and references
   Reading: Chapter 9
4. **Week 4:** Linked Lists
   Reading: Materials provided by the instructor
5. **Week 5:** Review, MIDTERM EXAM 1
   Reading: none
6. **Week 6:** Recursion
   Reading: Chapter 18
7. **Week 7:** Recursion
   Reading: Chapter 18
8. **Week 8:** Complexity analysis, big-oh notation
   Reading: Sections 22.1, 22.2, 22.3, 22.4.1, 22.4.6, and 22.7
9. **Week 9:** Sorting
   Reading: Sections 7.11, Sections 23.1-23.5
10. **Week 10:** Review, MIDTERM EXAM 2
    Reading: none
11. **Week 11:** Abstract data types, generics, Lists
    Reading: Chapter 13, Section 24.4
12. **Week 12:** Stacks and Queues
    Reading: Sections 24.5 and 20.10
13. **Week 13:** Trees, Binary Search Trees, AVL trees
    Reading: Section 25.1-3, Chapter 26
14. **Week 14:** Review, MIDTERM EXAM 3
    Reading: none
15. **Week 15:** Review
    Reading: Section 25.5