CSCE 2100 Computing Foundations I

Instructor:	Dr. Joseph Helsing, Joseph.Helsing@unt.edu
Office Hours:	2:30pm – 3:30pm, Monday/Wednesday, NTDP F203
	11:00am – 12:00pm, Tuesday/Thursday, NTDP F203
Class Room:	NTDP B185
Meeting Time:	4:00pm - 5:20pm, Monday/Wednesday

Teaching Assistant:	**F232**
Farzana Rashid	farzanarashid@my.unt.edu
Office Hours:	TBD
Zongze Li	ZongzeLi2@my.unt.edu
Office Hours:	TBD

Instructional Assistant:	**F232**
Augustin Tran	AugustinTran@my.unt.edu
Office Hours:	TBD

Course Objective:
Introduces students to both data structures and formalisms used in computer science, such as asymptotic behavior of algorithms. Data structures and the formalisms used to both describe and evaluate those data structures simultaneously. By the end of the two-semester sequence of which this course is the first part, each student will have a solid foundation in conceptual and formal models, efficiency, and levels of abstraction as used in the field of computer science.

Required Textbook:
We will be using an online textbook this semester through zyBooks. To access the text book you will need to:

1. Sign in or create an account at learn.zybooks.com
2. Enter zyBook code: UNTCSCE2100HelsingFall2018
3. Subscribe

Optional Reference Textbook:
The Foundations of Computer Science, by Alfred Aho & Jeffrey Ullman
http://infolab.stanford.edu/~ullman/focs.html

Expected Student Outcomes:
Student Outcomes are measurable achievements to be accomplished by the completion of the degree. These outcomes are evaluated as part of our ABET accreditation process.

Computer Engineering Students:

1. An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
2. An ability to acquire and apply new knowledge as needed, using appropriate learning strategies

Computer Science Students:

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. Apply computer science theory and software development fundamentals to produce computing-based solutions.

Information Technology Students:

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. Identify and analyze user needs and to take them into account in the selection, creation, integration, evaluation, and administration of computing-based systems.

Expected Course Outcomes:
Course Outcomes are measurable achievements to be accomplished by the completion of the course. These outcomes are evaluated as part of our ABET accreditation process.

1. Demonstrate a solid foundation in conceptual and formal models by describing loop structures in summation and/or product notation.
2. Use abstraction in the design and implementation of algorithms.
3. Design programming solutions to "simple" problems and implement those designs in C or C++.
4. Apply big-Oh notation to evaluate and compare algorithms and programs.
5. Use combinatorics and conditional probability in solving real-world problems.
6. Define the basic operations of sets, functions, relations, trees and graphs.
7. Demonstrate an introductory knowledge of formal languages by using regular expressions, deterministic finite automata and non-deterministic finite automata to describe patterns in strings.

Attendance Policy:
Students are encouraged to attend all lectures in order to gain the full benefit of the course. While I will be posting my slides before class, they will not contain all of the content discussed during class, nor the examples presented on the board. If you are not able to attend class or recitation, please email me as soon as possible.

Content Responsibility Policy:
Students are responsible for all content presented in class, required readings from the textbook, and problems presented during the recitations. While attendance will not be taken in class, and your individual progress in the textbook will not be monitored, you will be expected to know and understand the requisite topics and concepts. If you are confused or unsure about anything, please ask myself or the TA's.

Submission Policy:
All projects are expected to be submitted on time with all of the correct parts through the Canvas system. The project design report must be in PDF format, and no photographed or scanned content will be scored. Further, any content that is deemed illegible will not be scored. The program must be coded in C or C++ and contain ample comments and descriptions. All programs will be compiled and executed on the department's CSE servers, and any that fail to compile or execute on that system will lose points. Additionally, a README file with clear instructions on how to compile and execute your program must be included.

Make-up Work Policy:
For most situations there will be no make-up work for any assessment in this course. However, in the event of an unavoidable absence for one of the reasons below, email me as soon as possible so we can work out a solution. The following events are grounds for make-up work: being a participant in a conference in which you are presenting; being in an athletic or other UNT associated event in which you are an active participant; a family emergency; a severe illness; military duty; or in certain cases and with some restrictions a religious event. Additionally, in the case of a missed assignment due to illness, make-up work will only be allowed by providing the instructor with a physical copy of a signed doctor's note. See the UNT Attendance Policy for more information.

Late Work Submission Policy:
All projects may be submitted up to one day late at a 50% penalty. Any projects submitted two or more days after the due date will receive a 0. All recitation problems must be turned in by the end of the recitation.

Academic Integrity and Collaboration:
UNT policy 06.003 defines the following breaches of academic integrity:

A. **Cheating.** The use of unauthorized assistance in an academic exercise, including but not limited to:
   a. use of any unauthorized assistance to take exams, tests, quizzes or other assessments;
   b. usage of sources beyond those authorized by the instructor in writing papers, preparing reports, solving problems, or carrying out other assignments; usage without permission, of tests, notes, or other academic materials belonging to instructors, staff members, or other students of the university;
   c. dual submission of a paper or project, or resubmission of a paper or project to a different class without express permission from the instructor;
   d. any other act designed to give a student an unfair advantage on an academic assignment.

B. **Plagiarism**. Use of another's thoughts or words without proper attribution in any academic exercise, regardless of the student's intent, including but not limited to:
   a. the knowing or negligent use by paraphrase or direct quotation of the published or unpublished work of another person without full and clear acknowledgement or citation.
   b. the knowing or negligent unacknowledged use of materials prepared by another person or by an agency engaged in selling term papers or other academic materials.

C. **Forgery**. Altering a score, grade or official academic university record; or forging the signature of an instructor or other student.

D. **Fabrication**. Falsifying or inventing any information, data or research as part of an academic exercise.

E. **Facilitating Academic Dishonesty**. Helping or assisting another in the commission of academic dishonesty.

F. **Sabotage**. Acting to prevent others from completing their work or willfully disrupting the academic work of others.

Cheating of any sort will not be tolerated in this course. All submissions must be your own original work. Taking information or code from the internet or other students is considered a breach of academic integrity. Failure to adhere to these strict standards will be cause for disciplinary action that could be as severe as expulsion from the university. If it is determined a student cheated on any assignment in this course they will receive an F for their final course grade and an academic integrity report will be filed with the Office of Academic Integrity. Further, UNT is now maintaining a database recording any acts of academic dishonesty that is available to employers.

Additionally, due to problems in the past do not post any of your code on public code repositories. We have had instanced where students completed the course, and then made their code publicly available via GitHub. Later students then discovered this code, downloaded it, and submitted it. If it is determined that code you wrote is used by another student at a future date, and that they obtained it from your repository, you will be held accountable as well for facilitating academic dishonesty.

For more information see the UNT Student Academic Integrity Policy.

Collaboration Policy:
For each project and exam, all work is expected to be your own. You are not to collaborate with other students outside of your group for projects, search for solutions on the internet, or provide solutions to others. Doing so will be deemed a breach of academic integrity. However, for the practice problems during recitations and any non-graded, practice assignments students are encouraged to work together to solve problems.

ODA:
The University of North Texas makes reasonable academic accommodation for students with disabilities. Students seeking reasonable accommodation must first register with the Office of Disability Accommodation (ODA) to verify their eligibility. If a disability is verified, the ODA will provide you with a reasonable accommodation letter to be delivered to faculty to begin a private discussion regarding your specific needs in a course. You may request reasonable accommodations at any time, however, ODA notices of reasonable accommodation should be provided as early as possible in the semester to avoid any delay in implementation. Note that students must obtain a new letter of reasonable accommodation for every semester and must meet with each faculty member prior to implementation in each class. Students are strongly encouraged to deliver letters of reasonable accommodation during faculty office hours or by appointment. Faculty members have the authority to ask students to discuss such letters during their designated office hours to protect the privacy of the student. For additional information see the Office of Disability Accommodation website at *http://www.unt.edu/oda*. You may also contact them by phone at **940.565.4323**.

Syllabus Revisions:
This syllabus may be modified as the course progresses should the instructor deem it necessary. Notice of changes to the syllabus shall be made through Canvas and/or in-class announcements.

Grading Policy:
By the end of the course you must have earned at least a 60% average from the projects. Failure to do so will result in a final grade of an F, despite having a potentially passing course average. Additionally, if there are questions about posted grades, they must be discussed with the instructor within one week of the grades being posted. After one week, barring an exceptional circumstance, grades will not be altered.

Grading Breakdown:

Assignments & Examinations	Total Percentage
Recitation Problems	**12.5%**
Project 1	**10%**
Project 2	**10%**
Project 3	**10%**
Exam 1	**12.5%**
Exam 2	**12.5%**
Exam 3	**12.5%**
Final Exam	**20%**

Important Dates:

Aug 27	First Class Day
Sept 3	Labor Day (NO CLASS)
Nov 5	Last day to drop a course
Nov 21	Last day to withdraw
Dec 12	Final Exam 1:30pm-3:30pm

Tentative Course Schedule:

Week	Dates	Topics	Exams	Projects
1	Aug 27,29	Introduction; Abstraction & Data Models; Sets		
2	Sept 5	Sets & Relations		
3	Sept 10,12	Sets & Relations		
4	Sept 17,19	Patterns, Automata, & Regular Expressions	Exam 1	
5	Sept 24,26	Patterns, Automata, & Regular Expressions		
6	Oct 1,3	Patterns, Automata, & Regular Expressions ;Iteration & Induction		Project 1 Due
7	Oct 8,10	Iteration & Induction		
8	Oct 15,17	Running Time of Programs	Exam 2	
9	Oct 22,24	Running Time of Programs		
10	Oct 29,31	Running Time of Programs; Graphs		
11	Nov 5,7	Graphs		Project 2 Due
12	Nov 12,14	Graphs	Exam 3	
13	Nov 19,21	Trees		
14	Nov 26,28	Trees; Combinatorics and Counting		
15	Dec 3,5	Combinatorics and Counting		Project 3 Due
16	Dec 12	Final Exam: 1:30-3:30pm	Final Exam	