CSCE 2100 Syllabus
Spring, 2016

Instructor: Fahmida Hamid
Office: NTDP F214
Phone: 940-369-7223
Office Hours: Monday 10:00 AM – 11:00 AM, Tuesday 1:00 PM – 2:00 PM
Email: Fahmida.Hamid@unt.edu

Class:
Section 001: MWF (8:30 AM – 9:20 AM), B185
Section 002: MWF (1:30 PM – 2:20 PM), B140

Teaching Assistant(s):
Yang Zhou
  Email: YangZhou2@my.unt.edu (recitation section: 202, 203)
A B M Rezbaul Islam
  Email: AbmRezbaulIslam@my.unt.edu (recitation section: 204, 205)

Grader:
Neha Kasala,
  Email: nehakasala@my.unt.edu
  Office Hours:
    Monday (10:00 AM – 12:00 PM),
    Tuesday (3:00 PM – 5:00 PM)

Textbook: (Available on line – for free !!!)
  Foundations of Computer Science, Aho and Ullman

Course Description:
Introduces students to both data structures and formalisms used in computer science, such as
asymptotic behavior of algorithms. Learn about 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.

Prerequisite(s): CSCE 1040

Course Outcomes:
Course outcomes are measurable achievements to be accomplished [by the student] by the
completion of a course. Outcomes are evaluated in ABET’s accreditation process.

- Use abstraction in design and implementation of algorithms.
- Use induction in proofs of algorithms and programs.
- Use C++ classes and inheritance in implementation of algorithms using list and tree data
  structures.
- Estimate the running time of programs using big-Oh notation.
- Use recurrence relations to evaluate the running time of recursive programs.
- Apply principles of combinatorics and probability to solve real-world problems.
Policies:

All the exams, the midterm and the final, will be held in Discovery Park (your class room).

All of your programs (three/four of them) will be written in C++. In addition, you should use a C++ that is compatible with that compiled on the “CSE” machines within the CSE department. **If your program does not compile with the CSE gcc or g++ and without any special flags, it will be counted as a program that has compile errors (which it does).**

**EXCEPTION:** You can, if you choose, submit a makefile with your program, in which case your program will be “compiled” using the makefile. It still needs to be written in C++ but it can use whatever compile flags you like, as long as it works on the CSE machines. But be warned. **If your makefile doesn’t work properly with whatever code you submit, your program will be considered to have compiler error(s).**

You will NOT use the Standard Template Library in your programs this term in 2100. Failure to follow this “guideline” will result in a 0 for the program. The one exception will be that you CAN use the String class, which doesn’t belong in the STL, anyway!

Late Submission Policy:

Most programs and homework assignments are due at 11:59 pm on the due date. All programs will be submitted to Blackboard. **A program or an assignment can only be late for at most 48 hours from the due date with a 20% penalty per day. Any submission after that time will not be considered.**

Attendance:

Attendance will not be taken in lecture. However, your attendance is strongly recommended to improve your opportunity to meet course outcomes. In addition, part of your grade (see grade break down below) will be based upon work done in class. Of course, if you are not in class when this work is done your grade will suffer.

Often students will ask “Is recitation attendance required?” Of course, in one sense attendance is never “required”, but a significant portion of your grade (and even more of your learning) is based upon full participation in recitation.

Grading:

Your grade will be weighted as follows:

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recitation Quizzes</td>
<td>10%</td>
</tr>
<tr>
<td>Homework Assignments</td>
<td>10%</td>
</tr>
<tr>
<td>Programming Assignments</td>
<td>25%</td>
</tr>
<tr>
<td>Class Test</td>
<td>10%</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>20%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>25%</td>
</tr>
</tbody>
</table>
Make-Up Policy:

There will be NO make-up exams, recitations, or programs given in this class. However, for documented *excused absences* or *emergencies* on a day of an exam or a recitation, the exam and/or recitation grades will be replaced by an average of the other exams or recitation scores. There is one exception to this rule.

Under NO circumstances will more than one exam or two recitation scores be replaced by an average of the other scores. For a second (and subsequent) missed exam or third recitation, even if all are excused, students will receive a 0 for the missed work.

Excused Absences

Students are expected to schedule routine appointments and activities that do not conflict with attending class. However, some absences cannot be prevented. In the event of a medical *emergency* or family death, students must request an excused absence as quickly as feasible following the emergency.

Students must provide documentation to verify the emergency.

Emergencies

By definition, emergencies cannot be planned for. Your instructor attempts to make accommodations in these instances that allow for making up missed work and completion of the course in a timely manner. Among these emergencies are

- A death in your *immediate* family
- An accident or illness requiring immediate medical treatment and where a doctor has indicated attending class is impossible or inadvisable.
- Employees on call 24/7 must document that they were called during class time.

Collaboration and Cheating:

Programming:

Do NOT work with other students on shared program solutions.

Do NOT get help with algorithms or coding from anyone other than the course instructor, the 2100 TA(s), grader, or peer mentor(s).

Do NOT use even partial program solutions from the internet unless those partial solutions are provided to you by the instructor as part of the assignment description. Failure to adhere to these strict standards will be cause for disciplinary action that could be as severe as expulsion from the university.
Do NOT use solutions posted at someone’s GitHub account. Do NOT post your solution in a public domain. Students in my class who place their solutions on GitHub or any other unprotected repository will be held equally responsible as any student who submits a copy of that program.

Those who are deemed to have cheated on program will receive a penalty that exceeds getting a 0 score for the program. Of course students who are deemed to have cheated have an appeal process of any penalty I invoke. If and when the time comes, that appeal process will be described to the student(s) in question.

It IS permissible to obtain help from whoever you wish to fix syntax errors. But remember, for anything but syntax errors, getting programming assistance from any source other than the instructor, or the 2100 TAs or peer mentor(s) will be considered cheating and will be penalized.

**Homework:**

You are expected to solve the problems by yourself, write it clearly, explain each intermediate step of your calculation, and turn it in the Blackboard.

You can submit pdf/doc/txt file. If you wish to write it by hand instead of typing it on a computer, make sure the scanned copies (pdf/image files) are clear and readable by the grader/TA.

**Recitations and Lectures:**

- In the first recitation class, talk to the TA and make sure you know how to connect to the CSE machines remotely from your laptop/home-computers.
- Bring LAPTOPs in the recitation classes.
- The recitation classes will be a blend of short programming tutorials and quizzes, and narrative questions.
- The TAs will grade the recitation quizzes, and the Grader will grade the Programming Problems and Homework Assignments

**Exams:**

All the exams will be closed-book unless I decide it to be otherwise.

**SPOT**

The Student of Teaching (SPOT) survey is a requirement for all organized undergraduate classes at UNT. This short survey will be made available to you at the end of the semester, providing you a chance to comment on how this class is taught. I am very interested in the feedback I get from students, as I work to continually improve my teaching. I consider the SPOT to be an important part of your participation in this class.

**ADA**

UNT complies with all federal and state laws and regulations regarding discrimination, including the Americans with Disability Act of 1990 (ADA). If you have a disability and need a reasonable accommodation for equal access to education or services, please contact the Office of Disability Accommodation.
<table>
<thead>
<tr>
<th>Tentative Schedule (Week)</th>
<th>Topic</th>
<th>Reading</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 01 01/17/17</td>
<td>Abstraction, Data Models, Data Structure, Algorithms</td>
<td>Chapter 01</td>
<td></td>
</tr>
<tr>
<td>Week 02 01/24/17</td>
<td>Iteration, Induction, Recursion</td>
<td>Chapter 02</td>
<td>Programming Assignment 01</td>
</tr>
<tr>
<td>Week 03 01/31/17</td>
<td>Iteration, Induction, Recursion</td>
<td>Chapter 02</td>
<td>Homework 01</td>
</tr>
<tr>
<td>Week 04 02/07/17</td>
<td>Run-time, Big-Oh</td>
<td>Chapter 03</td>
<td></td>
</tr>
<tr>
<td>Week 05 02/14/17</td>
<td>Big-Oh</td>
<td>Chapter 03</td>
<td>Homework 02</td>
</tr>
<tr>
<td>Week 06 02/21/17</td>
<td>Combinatorics and Probability</td>
<td>Chapter 04</td>
<td></td>
</tr>
<tr>
<td>Week 07 02/28/17</td>
<td>Combinatorics and Probability</td>
<td>Chapter 04</td>
<td>Programming Assignment 02</td>
</tr>
<tr>
<td>Week 08 03/07/17</td>
<td>Combinatorics and Probability</td>
<td>Chapter 04</td>
<td>Midterm</td>
</tr>
<tr>
<td>Week 09 03/14/17</td>
<td></td>
<td></td>
<td>SPRING BREAK 03/13/17 – 03/19/17</td>
</tr>
<tr>
<td>Week 10 03/21/17</td>
<td>List Data Model</td>
<td>Chapter 06</td>
<td>Homework 03</td>
</tr>
<tr>
<td>Week 11 03/28/17</td>
<td>List Data Model</td>
<td>Chapter 06</td>
<td></td>
</tr>
<tr>
<td>Week 12 04/04/17</td>
<td>Tree Data Model</td>
<td>Chapter 05</td>
<td>Programming Assignment 03</td>
</tr>
<tr>
<td>Week 13 04/11/17</td>
<td>Tree Data Model</td>
<td>Chapter 05</td>
<td></td>
</tr>
<tr>
<td>Week 14 04/18/17</td>
<td>Tree Data Model</td>
<td>Chapter 05</td>
<td>Homework 04</td>
</tr>
<tr>
<td>Week 15 04/25/17</td>
<td>Predicate Logic</td>
<td>Chapter 14</td>
<td>Homework 05</td>
</tr>
<tr>
<td>Week 16 05/02/17</td>
<td>Review Class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>05/06/17 – 05/11/17</td>
<td></td>
<td></td>
<td>FINAL EXAM</td>
</tr>
</tbody>
</table>

This syllabus may be modified as the course progresses. Notice of such changes will be announced in class or through Blackboard.