CSCI/CMPE 2380 Computer Science II

Section 03, Spring 2017, TR, 9:25am – 10:40am
Lecture meetings: ENGR 1.268

Instructor: Emmett Tomai
Office: ENGR 3.2100
Phone: 665-7229
Email: **emmett.tomai@utrgv.edu**
Office hours: TR 10:45-12:00, W 1:00-2:00, or by appointment

Course web site:

http://faculty.utrgv.edu/emmett.tomai/courses/2380/
All course materials are available on the course web site. Course announcements will be made in class and posted on the web site.

Course Description (UTPA Undergraduate Catalog):

CSCI/CMPE 2380 Computer Science II: A second programming course includes problem solving by structured design; provides an introduction to elementary data structures, including linked lists, stacks, queues, trees and graphs, and advanced programming techniques, including recursion, sorting and searching. Prerequisite: CSCI/CMPE 1370 or consent of instructor. Cannot receive credit for both CSCI 2380 and CMPE 2380.

Textbook:

For this section, we will be using the Zyante Programming in C++ online interactive text, which contains interactive examples and activities. The instructor will provide sign-up instructions in class and via email. Students who have limited internet access must speak directly to the instructor to arrange accommodation.

Course Objectives:

Students will:

1. Continue the development of problem solving skills in a context that emphasizes a structured, top-down approach.
2. Learn the application of software engineering principles in designing, coding and testing large programs.
3. Learn the object-oriented paradigm for developing software.
4. Learn to design and implement C++ classes that correctly and efficiently manage memory.
5. Learn essential data structures such as linked lists, stacks, queues, and trees. This introduction emphasizes the specification of each structure as an ADT.
6. Engage with a systematic approach to the study of algorithms that focuses first on the understanding of the algorithm and then on analyzing the algorithm from a time/space perspective. In particular searching, sorting, and recursive algorithms are covered.

7. Learn to evaluate the time and space tradeoffs in the design and implementation of ADT's.

Learning outcomes:

Throughout this course, students will begin to develop:

(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline

(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

(i) An ability to use current techniques, skills, and tools necessary for computing practice

(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices

(k) An ability to apply design and development principles in the construction of software systems of varying complexity

Course Requirements:

Lectures and labs: This course is scheduled for two meetings each week. Often, one of those meetings will be held in a computer lab for hands on work. You are expected to attend every lecture and lab period for this course. If you know you are going to be unable to attend, contact the instructor *beforehand* to arrange your absence. If you do not do so, you will not be given extensions or make-ups for any lab work, exercises, exams or other activities.

Labs and other in-class activities: Every week we will have hands-on assignments to work on in lab and at home. There will also be a variety of in-class activities, including worksheets and small projects. If you are not present in lecture or lab for these activities, you lose that credit.

Assignments: There will be 3-4 larger homework assignments involving writing, compiling and running your own programs in the Microsoft Visual C++ environment. These assignments will be announced in class and posted on the class website.

Exams: The material in this course is naturally cumulative, with each week's topics building on all the prior material. In order to provide consistent feedback at a reasonable pace, regular short exams will be given roughly every three weeks, in place of a single midterm examination. There will be a final exam as well.

Scoring and Grading:

Quizzes & exams	50%
Assignments	30%

Labs & participation	20%

Total possible score (max):	100%

Final grade:

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

Note: Grades on assignments and exams may be curved to reflect the overall performance of the class.

Course schedule

This is a rough course schedule to give you an idea of topics and pacing. The actual course schedule is kept up to date on the course website.

Week 1-3: Object-oriented programming program design in C++
Week 4-7: Memory management
Week 7-11: Composition and inheritance
Week 11-15: Data structures

Course Policies

Drop Class Policy:

It is the student's responsibility to Drop the class if desired, and be aware of the drop deadline.

Computer use policies:

Please read and be aware of University policies for computer use, which can be found at:
http://www.utpa.edu/policies/UTPAAcceptableUse.pdf

Late Work Policy:

No work in this course will be accepted late.

Make-up Policy:

No make-up exams or quizzes will be given except for university sanctioned excused absences. If you miss an exam (for a good reason), it is your responsibility to contact me before the exam, or as soon after the exam as possible.

Academic Integrity Policy:

The University expects a student to maintain a high standard of individual honor in his/her scholastic work. Unless otherwise required, each student is expected to complete his or her assignment individually and independently. Although study together is encouraged, the work handed in for grading by each student is expected to be his or her own. Any form of academic dishonesty will be strictly forbidden and will be punished to the maximum extent. Copying an assignment from another student in this class or obtaining a solution from some other source will lead to disciplinary action. Allowing another student to copy one's work will be treated as an act of academic dishonesty, leading to the same penalty as copying.

Note to students with disabilities:

Students with disabilities are encouraged to contact the Student Accessibility Services office for a confidential discussion of their individual needs for academic accommodation. It is the policy of the University of Texas Rio Grande Valley to provide flexible and individualized accommodation to students with documented disabilities that may affect their ability to fully participate in course activities or to meet course requirements. To receive accommodation services, contact SAS: University Center 108, sas@utrgv.edu, 956-665-7005. http://www.utrgv.edu/en-us/student-experience/student-academic-success/student-accessibility-services/

Mandatory course evaluation period:

Students are required to complete an ONLINE evaluation of this course, accessed through your UTPA account (https://my.utpa.edu/); you will be contacted through email with further instructions on the evaluation process. Students who complete their evaluations on time will have priority access to their grades.