CSCI/CMPE 2380-02
Engineering Computer Science II

Course Information

<table>
<thead>
<tr>
<th>Instructor</th>
<th>Dr. Tim Wylie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>Office: ENGR 3.287</td>
</tr>
<tr>
<td></td>
<td>Phone: 956-665-2577</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:timothy.wylie@utrgv.edu">timothy.wylie@utrgv.edu</a></td>
</tr>
<tr>
<td>Office Hours</td>
<td>TR 9:00 a.m. - 11:00 a.m.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Lecture: MW, 12:15 p.m. - 1:30 p.m., ENGR 1.274</td>
</tr>
<tr>
<td></td>
<td>Lab: R, ACSB 2.110</td>
</tr>
<tr>
<td>Final Exam</td>
<td>May 9, 2018, 10:15 a.m. - 12:00 p.m., ENGR 1.274</td>
</tr>
<tr>
<td>Course Website</td>
<td><a href="http://academic.timwylie.com/18CSCI2380">http://academic.timwylie.com/18CSCI2380</a></td>
</tr>
</tbody>
</table>

Course Description

CSCI/CMPE 2380 - Engineering 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. Prerequisites: CSCI/CMPE 1370 and CSCI/CMPE 1170 or consent of instructor.

Course Topics

This course will cover advanced programming techniques in C++ using basic principles of software engineering. Many fundamental concepts of object-oriented programming will be covered including classes, objects, abstraction, and inheritance. From this, students can begin to explore abstract data types such as stacks, queues, linear and circular linked lists, and binary tree data structures. These will be covered in detail with various implementations considered. Pointer data types and recursion will be explored as implementation techniques. Several sorting and search algorithms will be introduced and compared for efficiency of storage and computation. This includes covering basic methods and terminology used for analyzing algorithms. Other topics and data structures may be introduced depending on student interest and available time.

Course Objectives

1. To continue the development of problem solving skills in a context that emphasizes a structured, top-down approach
2. To demonstrate the application of software engineering principles in designing, coding and testing large programs
3. To introduce students to essential data structures such as linked lists, stacks, queues, and trees. This introduction emphasizes the specification of each structure as an abstract data type (ADT)
4. To provide 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
5. To evaluate the time and space trade-offs in the design and implementation of ADTs
6. To make students aware of the importance of object-oriented methods in developing software, particularly in the design and implementation of ADTs
Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Design, implement, test, and debug simple programs in an object-oriented programming language.
2. Write programs that use each of the following data structures: arrays, records, strings, linked lists, stacks, queues, and hash tables.
3. Choose the appropriate data structure for modeling a given problem.
4. Identify the base case and the general case of a recursively defined problem.
5. Implement, test, and debug simple recursive functions and procedures.
7. Demonstrate different traversal methods for trees and graphs.
8. Model problems in computer science using graphs and trees

ABET Outcomes

1. An ability to apply knowledge of computing and mathematics appropriate to the discipline
2. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
3. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
4. An ability to use current techniques, skills, and tools necessary for computing practice
5. 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
6. An ability to apply design and development principles in the construction of software systems of varying complexity

Course Overview

Labs. There will be a weekly lab during this course relevant to the topic being covered. These will be critical to understanding the material and will require substantial work. They will usually be submitted electronically by the specified due date and time. Late labs will not be accepted. Occasionally, a class exercise may be used as a lab.

Assignments. Students are expected to have assignments completed at the beginning of class (or when specified) on the due date. This may include selected readings from the textbook.

Exams. The material in this course is naturally cumulative, with each week’s topics building on all the prior material. Therefore, each exam will focus on the material covered since the previous exam, however, the student is expected to understand and apply all previous course material.

Scoring and Grading. Final grades are calculated based on the number of accepted labs, assignments, and the average of the exams. In order to get credit for the assignment or lab, it must be turned in on time and be correct. Whether credit is given or not is at the discretion of the instructor. If unacceptable but close, corrections may be possible. The table below lists the minimum number of accepted homeworks and labs and the minimum test average a student must have to get the corresponding final grade.

<table>
<thead>
<tr>
<th>Final Grade</th>
<th>Test Average (100%)</th>
<th>Homeworks (10)</th>
<th>Labs (10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90%</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>B</td>
<td>80%</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>C</td>
<td>70%</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>D</td>
<td>50%</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>F</td>
<td>failure to meet the minimum conditions for a D</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Grades 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 likely to change and will be kept up to date on the course website.

Week 1-2: Review of CSCI/CMPE 1370, introduction to object-oriented programming
Week 3-4: Object-oriented programming, pointers
Week 5-6: Advanced object-oriented techniques, recursion
Week 7-9: Linked lists
Week 10-12: Stacks, queues, searching and sorting
Week 13-14: Binary search trees, graphs
Week 15: More advanced data structures, topics of interest

Late Work Policy.  No class work will be accepted late unless the instructor Assignments turned in early will receive additional credit.

Make-up Policy.  No make-up exams will be given except for university sanctioned excused absences. If you need to miss an exam, it is your responsibility to contact me before the exam, or as soon after the exam as possible. Missing an exam without an approved (by the university or me) excuse will result in a zero.

UTRGV Course Policies

Attendance.  Students are expected to attend all scheduled classes and may be dropped from the course for excessive absences. UTRGVs attendance policy excuses students from attending class if they are participating in officially sponsored university activities, such as athletics; for observance of religious holy days; or for military service. Students should contact the instructor in advance of the excused absence and arrange to make up missed work or examinations.

Drop Class Policy.  According to UTRGV policy, students may drop any class without penalty earning a grade of DR until the official drop date. Following that date, students must be assigned a letter grade and can no longer drop the class. Students considering dropping the class should be aware of the 3-peat rule and the 6-drop rule so they can recognize how dropped classes may affect their academic success. The 6-drop rule refers to Texas law that dictates that undergraduate students may not drop more than six courses during their undergraduate career. Courses dropped at other Texas public higher education institutions will count toward the six-course drop limit. The 3-peat rule refers to additional fees charged to students who take the same class for the third time. The census date is January 31st, which is the last day to drop the class without it appearing on your transcript, and the last date to drop is April 12th.

Computer Use Policy.  Please read and be aware of University policies for computer use and data security, which can be found at: http://www.utrgv.edu/is/_files/documents/utrgv-aup.pdf

Scholastic Integrity Policy.  As members of a community dedicated to Honesty, Integrity and Respect, students are reminded that those who engage in scholastic dishonesty are subject to disciplinary penalties, including the possibility of failure in the course and expulsion from the University. Scholastic dishonesty includes but is not limited to: cheating, plagiarism, and collusion; submission for credit of any work or materials that are attributable in whole or in part to another person; taking an examination for another person; any act designed to give unfair advantage to a student; or the attempt to commit such acts. Since scholastic dishonesty harms the individual, all students and the integrity of the University, policies on scholastic dishonesty will be strictly enforced (Board of Regents Rules and Regulations and UTRGV Academic Integrity Guidelines). All scholastic dishonesty incidents will be reported to the Dean of Students.

Course Evaluation.  Students are required to complete an ONLINE evaluation of this course, accessed through your UTRGV account (http://my.utrgv.edu); you will be contacted through email with further instructions. Students who complete their evaluations will have priority access to their grades. Online evaluations will be available:
Spring 2018 (full semester) April 11 – May 2
Sexual Harassment, Discrimination, and Violence. In accordance with UT System regulations, your instructor is a “responsible employee” for reporting purposes under Title IX regulations and so must report any instance, occurring during a student’s time in college, of sexual assault, stalking, dating violence, domestic violence, or sexual harassment about which she/he becomes aware during this course through writing, discussion, or personal disclosure. More information can be found at www.utrgv.edu/equity, including confidential resources available on campus. The faculty and staff of UTRGV actively strive to provide a learning, working, and living environment that promotes personal integrity, civility, and mutual respect in an environment free from sexual misconduct and discrimination.

Students with Disabilities. Students with a documented disability (physical, psychological, learning, or other disability which affects academic performance) who would like to receive academic accommodations should contact Student Accessibility Services (SAS) as soon as possible to schedule an appointment to initiate services. Accommodations can be arranged through SAS at any time, but are not retroactive. Students who suffer a broken bone, severe injury or undergo surgery during the semester are eligible for temporary services.

- **Brownsville Campus:** Student Accessibility Services is located in Cortez Hall Room 129 and can be contacted by phone at (956) 882-7374 (Voice) or via email at ability@utrgv.edu.
- **Edinburg Campus:** Student Accessibility Services is located in 108 University Center and can be contacted by phone at (956) 665-7005 (Voice), (956) 665-3840 (Fax), or via email at ability@utrgv.edu.