Syllabus for CSCI 3333(-01,-02,-04):
  Algorithms & Data Structures
  Spring 2018

Course Information

Course description. A continuation of the topics covered in CSCI 2380 and CSCI 3310. Focuses on the analysis and design of algorithms (sorting, searching, dynamic programming) and data structures (priority queues, trees, hash tables, graphs). Also covers C++ implementation of algorithms and data structures discussed.

• Instructor: Andrew Winslow (andrew.winslow@utrgv.edu).
• Course webpage: http://andrewwinslow.com/3333/.
• Lecture:
  – Section 01: 10:50-12:05 MW in ELABN 109.
  – Section 02: 1:40-2:55 MW in EENGR 1.236.
  – Section 04: 10:50-12:05 TR in EHABE 1.124.
• Office hours: 3:00-4:00 MTWR, 10:00-11:00 F in EENGR 3.279.
• Final exam:
  – Section 01: 10:15-12:00 on Monday, May 7.
  – Section 02: 1:15-3:00 on Monday, May 7.
  – Section 04: 10:15-12:00 on Tuesday, May 8.
• Anonymous feedback: http://sayat.me/AndrewWinslow.

Prerequisites. Students must have completed CSCI/CMPE 2380 or CSCI/CMPE 2388 (Computer Science II). Students must have completed or be currently taking MATH 2346 (Math for EE & CE) or CSCI 3310 (Mathematical Foundations of Computer Science) or MATH 2305 (Discrete Mathematics).

Textbook. There is no required textbook for the course. The recommended textbook is Data Structures and Algorithm Analysis in C++, 4th Edition by Mark A. Weiss. This book is available on Amazon and other places. The textbook is useful in three ways:

• A second presentation of all course material.
• A reference for C++ topics (classes, pointers, references, templates, STL).
• A reference for mathematics (Big-O, logarithms, summations).

Course schedule and topics. Below is a rough schedule of the course and topics covered; the exact schedule will be maintained and updated on the course webpage.

• Week 1: Administrivia.
• Week 2: Big-O, C++ tools.
• Weeks 3-4: Sorting algorithms.
• Weeks 5-7: Trees-based data structures.
• Weeks 8-10: Hash tables and heaps.
• Weeks 11-13: Graph algorithms.
• Weeks 14-15: Dynamic programming, NP-completeness.

Grading. The course grade is four parts:

• Homework: weekly individual programming assignments, each with two deadlines:
  – Compile: writing a program that compiles.
  – Test: writing a program that passes all tests.
• Exercises: twice-weekly in-class 10-minute paper-&-pencil group exercises.
• Quizzes: twice-weekly in-class 10-minute paper-&-pencil individual tests.
• Exams: two in-class 70-minute paper-&-pencil individual tests.

Each part determines a portion of the final grade as follows:

• Homework: 48% total (4% per homework).
  – 1% for each compile deadline.
  – 3% for each pass deadline.
• Exercises: 7% total (evenly divided across exercises).
• Quizzes: 25% total (evenly divided across quizzes).
• Exams: 20% total (10% each).

The final grade is determined by computing the weighted total (out of 100%) of all four parts and applying the following percentage-to-letter-grade function: 90%-100% → A, 80%-89% → B, 70%-79% → C, 60%-69% → D, 0%-59% → F. Grades may be curved to reflect the overall performance of the class.
**Slack.** Slack is a popular tool for communicating in software development teams. The course uses Slack as a way to quickly and effectively seek course help from other students and the instructor.

At the beginning of the semester, instructions to join the course Slack team will be sent to all enrolled students (via Blackboard). Joining the Slack is not mandatory, but is strongly encouraged.

**Feedback.** Constructive feedback about the course is welcome at any time. Anonymous feedback can be given using sayat.me.

### Course Policies

**Attendance.** Students are expected to attend all scheduled classes and may be dropped from the course for excessive absences. The UTRGV 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.

**Late work.** Late submissions for the *test deadline* of each homework (except hwBTC) are accepted up to one week after the original deadline for 70% credit. The late submission process is identical to the regular submission process, with the last late submission used for grading.

No other late submissions are accepted.

**Dropping classes.** 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.

- **6-drop rule:** 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.

- **3-peat rule:** additional fees are charged to students who take the same class for the third time.

**Scholastic integrity.** As members of a community dedicated to honesty, integrity, and mutual respect in all interactions and relationships, students, faculty, and administration of our university pledge to abide by the principles in the Vaquero Honor Code. For more information, see the Student Conduct and Discipline Code.
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 Module 1 February 14 - February 20
Spring 2018 Module 2 April 11 - April 17
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.

Student Learning and ABET Outcomes

Student learning outcomes. Upon successful completion of this course, students will be able to:

1. Understand basic data structures and abstract data types.
2. Gain an appreciation of the variety, theoretical nature, and practical uses of data structures.
3. Select appropriate data structures for uses in computer programs.
4. Understand the basic techniques of algorithm design and analysis.
5. Understand the basic concepts of computational complexity.
6. Design and implement efficient algorithms based on the selected data structures.

**ABET Outcomes for CSCI 3333**

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. Recognition of the need for and an ability to engage in continuing professional development.

5. An ability to use current techniques, skills, and tools necessary for computing practice.

6. 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.

7. An ability to apply design and development principles in the construction of software systems of varying complexity.

**ABET Outcomes for CMPE 3333**

1. An ability to apply knowledge of mathematics, science, and engineering.

2. An ability to design and conduct experiments, as well as to analyze and interpret data.

3. An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.

4. An ability to identify, formulate, and solve engineering problems.

5. A recognition of the need for, and an ability to engage in life-long learning.


7. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.