CSCI/CMPE 1370 Engineering Computer Science I
CSCI/CMPE 1170 Engineering Computer Science I Laboratory

Section 02, Fall 2016, MW 9:25am – 10:40am
Lab Section 02, F 1:20pm-3:50pm

Instructor: Emmett Tomai
Office: ENGR 3.2100
Phone: 665-7229
Email: **e**mmett.tomaie@utrgv.edu
Office hours: MW 1pm – 2pm, TR 10:45am - 12pm, or by appointment

Course web sites:

All course information, materials and announcements: http://faculty.utpa.edu/tomaie/courses/1370/

Course Description (UTPA Undergraduate Catalog):

CSCI/CMPE 1370 Engineering Computer Science I: An introduction to computer science and computer engineering. The fundamentals of a high-level programming language will be introduced. Methods of problem solving, techniques of algorithmic development and concepts of procedural and object-oriented programming will be emphasized. Societal and social issues related to computer engineering will be introduced. Prerequisites: Computer Literacy experience and enrollment in or credit for CSCI 1101 or CMPE 1101, grade of "C" or better in MATH 1340 or placement in a higher level math course. Corequisite: CSCI/CMPE 1170. Cannot receive credit for both CSCI 1380 and CSCI/CMPE 1370. Will replace a grade received in CSCI 1380. Cannot receive credit for both CSCI 1370 and CMPE 1370.

CSCI/CMPE 1170 Engineering Computer Science I Laboratory: The course includes hands-on instruction and laboratory exercises in developing programs written in a high level object-oriented programming language applying the principles taught in the CMPE 1370 lecture course. Corequisite: CMPE 1370. Cannot receive credit for both CSCI 1170 and CMPE 1170.

Experimental Pilot Section:

This is the experimental pilot section in Python. The other sections are teaching the established C++ curriculum. If you take this section, you must also take the matching CS II (2380) section in the Spring, where we will transition to C++. Other 2380 sections will expect you to have prior C++ experience.

Textbook:

For this section, we will be using the Zyante Programming in Python 3 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:

This course is an introduction to Computer Science and the practice of programming for CSCI and CMPE majors and minors. It is part of the Component Area Option in the General Education Core, with alignment to the Mathematics Foundational Component Area. It focuses on techniques of problem solving and algorithmic design in the context of computer programming. Students will improve qualitative literacy in logic, patterns and relationships, as they learn to analyze problems in the formal mathematical concepts of computation, and design appropriate solutions. They will apply these tools to everyday problems through both theoretical and hands-on implementation of algorithms in the Python 3 programming language.

Students will learn:

1. How to analyze a problem as a formal mathematical system of quantitative data, equations and logical processes
    1. To quantify data in the problem
    2. To identify patterns and relationships between quantities
    3. To identify equations and logical processes that describe the domain
    4. To formalize those elements in computational terms of digital data and algorithms
2. How to design and write computer programs to solve problems
    a) How to implement algorithms by writing C++ code
    b) How to compile and link that code into a working program
    c) How to use testing and debugging strategies to identify and fix program faults
3. How *programming language*, *libraries* and *development environment* each impact the way programs are written
4. How a program can be written many different ways, and why some are better than others
    a) How different algorithms meet different requirements
    b) How to modularize code for clarity, testing and reuse
    c) How to evaluate, use and modify existing algorithms
5. How to write and document your code so that it is useful to other programmers (including yourself!)
6. How technical issues inform discussion of practical applications and ethical decisions.

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

(d) An ability to function effectively on teams to accomplish a common goal

(e) An understanding of professional, ethical, legal, security and social issues and responsibilities

(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: 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 exercises, exams or other activities.

In-class exercises: There will be a variety of in-class exercises during this course, including worksheets and small projects. They will be completed during class time and turned in at the end of class. You are expected to attend every lecture and take part in these exercises.

Out-of-class exercises: There will be regular exercises to be completed at home and either posted online or brought to class. You are expected to come to class prepared, having completed these exercises.

Failure to complete in-class and out-of-class exercises on a regular basis will be considered grounds for being dropped from the course.

Labs: There will be weekly lab assignments given out in the 1170 lab period. 1-3 labs will be due each week.

Assignments: There will be homework assignments involving writing and testing larger programs in Python 3. 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, each of the 5 content modules will have its own smaller exam, in place of larger midterm examinations. There will be a final exam as well.

Scoring and Grading:

This course is graded on a simple point scale. 10,000 points is a perfect score. Roughly, exams are worth 40% of the possible points, homeworks 20%, labs 30% and exercises 10%. See the course website for available points.

Final grade:

9000 points or more	A
8000 points or more	B
7000 points or more	C
6000 points or more	D
Less than 6000 points	F

CSCI/CMPE 1370 and 1170 are co-requisite courses. The purpose of the lab section is to increase your hands-on experience with the material, and to provide you with another avenue to demonstrate what you have learned. **You will receive the same grade for both courses.** That grade is calculated based on the total work performed in both CSCI/CMPE 1370 and 1170.

If you are enrolled in only one of CSCI/CMPE 1370 or 1170, **please contact the instructor immediately**.

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 1-2: Program organization and execution, data and variables, user input
Week 3-5: Conditional and iterative execution
Week 6-8: User-defined functions and parameters
Week 9-12: Arrays and structured data
Week 13-15: Classes, pointers and dynamic memory

Course Policies

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.

Drop Class Policy:

It is the student’s responsibility to Drop the class if desired, and be aware of the drop deadline. The student must drop both CSCI/CMPE 1370 and 1170.

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.utrgv.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.

Computer use policies:

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