

CSCI/CMPE 1370
Engineering Computer Science I
(for CSCI/CMPE majors, minors)

Emmett Tomai

Office: ENGR 3.2100

Phone: 665-7229

Email: emmett.tomai@utrgv.edu

Question #1

Why are you here?

Question #1

Why are you here?

Where do you see yourself five years from now?

How is this class going to help that happen?

Question #2

What do you imagine you're going to be doing in this class?

Make Computers Solve Problems!

- Goal #1:
 - Introduce you to the ideas behind computer science
 - What can be solved? How? How hard is it?
- Goal #2: Learn two practical skills:
 - How to design computer programs to solve problems
 - How to create those programs

Computer Science I

- This requires a little knowledge and a **lot of practice**
 - Part fundamentals, repetition
 - Part creative exploration (the point!)



Who am I?

- IT professional
 - 3 years during college, mostly Unix and network administration
- Consultant
 - 5 years, mostly web development, with a firm and as an independent
- Professor of Computer Science at UTPA
 - 7th year
 - Teach CS I, CS II, Internet (web) Programming, Game Development
- Researcher
 - Artificial Intelligence
 - Computational language and narrative
 - Stories and learning in virtual environments

Who am I?

- Family Guy
 - Married 17 years
 - 3 kids (10, 7 and 4), who take up most of my time
- Christian Guy
 - InterVarsity Christian Fellowship in college
 - BT McAllen here
- Hawaii/Chicago Guy
 - Grew up in Hawaii, college, work and grad school in Chicago
- And more!
 - Video games, basketball, drumming, board games...

Who am I?

- Computer Engineer, Computer Scientist
 - I make computers solve problems!

Experimental Pilot Section

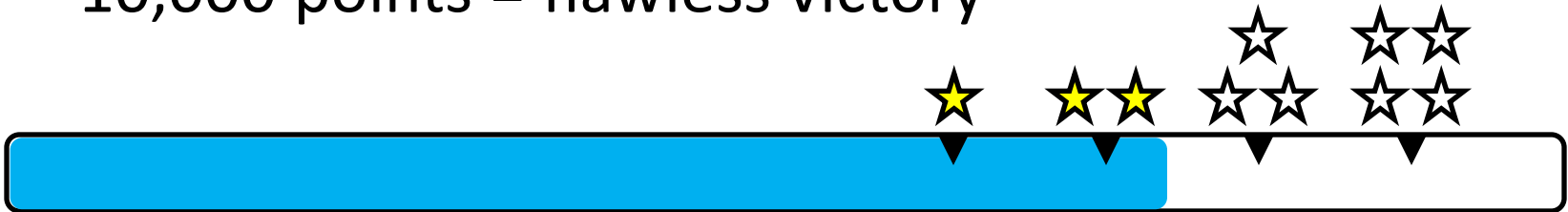
- This section of 1370 will be taught in *Python*
 - The other sections are taught in C++
 - More focus on computational problem-solving
 - Python allows us to work on more interesting problems
- CS II (2380) is taught in C++
 - If you are in this section, you *must* also take my 2380 section in the Spring!
 - Other sections will assume you already know C++
 - My section will assume you know more about problem-solving
 - We will catch up on C++
 - Will be offered at the same time slot in the Spring

Computer Science I

- Broken up into 5 modules (highly cumulative!)
 - Each module has it's own exam!
- In class we will:
 - Consider, analyze and discuss problems
 - Discover relevant concepts, show examples
 - Design solutions
 - Implement our solutions in the lab
- Outside class, you will:
 - Research and explore solutions
 - Practice fundamentals and test yourself
- **Expect to spend at least as much time outside of class as you spend in class**

Grading

- 10,000 points = flawless victory



- How do you get points?
 - In-class activities, at-home follow-up exercises (10%)
 - Weekly lab assignments (30%)
 - Larger homework assignments (20%)
 - Module exams, final exam (40%)

Course Information

<http://faculty.utpa.edu/tomaie/courses/1370>

- All course materials and announcements will be available on this web site
 - Syllabus
 - Instructor and TA contact information
 - Lecture materials
 - Assignments, labs, review questions
 - Due dates, exam schedule, announcements
- Course announcements and other updates will be handled via email
 - You are expected to check your email every weekday and at least once on weekends

Lab Section

- You must be registered for the co-requisite lab course CSCI/CMPE 1170.02
 - Fridays at 1:20 in the ACSB building next door
 - Hands-on time to implement and test
 - Time to work together with peers and TAs there for support
 - Partly self-paced, won't generally take 3 hours
- You will receive the same grade for 1370 and 1170, based on all the work done

Textbook

- Zyante *Programming in Python 3* online interactive text
 - Contains interactive activities and practice
 - Allows me to follow your progress
 - ~\$50, online access good through end of semester
 - Can be downloaded and saved
 - Limited internet access? Online payment issues? See me!
 - Link to subscribe will be on the Course Info page of the course website
- Want a physical textbook?
 - Go for it, any one will be fine
 - I'd recommend one with lots of examples

Tools

- We will be using Python 3
 - Comes with the simple IDLE environment
 - Enough to start, can add better tools as you go
 - Free download at *python.org*
 - See the course website for details
 - If you're not regularly writing code, at a computer, outside of class, **you're doing this course wrong.**
- The textbook has it's own web-based programming environment
 - Convenient for some exercises and some labs

Lab Space

- ENGR 2nd Floor Computer Science lab (2.212)
 - Open around 8am to 9:45pm every day
 - This is *by far* the best place to do you out-of-class work
 - Meet other majors, older students
 - Students with community do far, far better throughout college
 - Lab consultants available for class help

Etc.

- Students are expected to attend all lectures and labs
 - You miss things, you miss the points
 - No make-ups, late turn-ins, etc.
- Got a legit excuse? Contact me *as soon as possible*
 - Happy to accommodate
- Know you're going to miss something?
 - Talk to me *beforehand*, we'll see what we can do

Office Hours

- By appointment, email one day in advance
 - Best way to make sure I'm there
- Scheduled drop-in (“office”) hours
 - See my schedule on the course website
 - Open drop in, first-come first-served
- Drop-in whenever
 - I may be there, I may not, check the calendar on the course website
 - I may be busy, if so I'll just tell you to come back later

Teamwork and Academic Honesty

- Students are encouraged to assist one another, but each student must still do their own work
- **Giving and receiving sections of code is cheating**
 - Both people are equally culpable
 - Discussing the problem? Good! Look at the same code? Cheating.
 - Never, ever, ever email someone else your code, they **will** copy it
- All students should be familiar with University policies on academic dishonesty
- Some homework assignments will allow teamwork in randomly assigned pairs
 - Partners are there to learn teamwork and motivate each other
 - Your partner is not there to carry you or teach you
 - You are responsible for the assignment, regardless of how helpful or not your partner is

Student Accessibility Services

- Services such as note takers, extended test time or separate accommodations for testing are available
 - Contact the Student Accessibility Services office
 - <http://www.utrgv.edu/en-us/student-experience/student-academic-success/student-accessibility-services/>
 - University Center 108
 - sas@utrgv.edu
 - 956-665-7005
- Verification of disability and processing is required
- Completely confidential

Closing Rants

- Don't waste your time
 - Sitting here in this classroom gets you nothing
 - Engage, or go do something more interesting to you
- Don't give up the easy points
 - The course is 60% activities, exercises, labs, and homework
 - Mostly about time management and effort
- Trying Harder Is Not a Plan!
 - Set up a realistic schedule for class and out-of-class time
 - Practice what you hear in class, test yourself, and repeat
 - Know when to use your resources (book, TAs, mentors, consultants, friends, instructors)
 - Don't let people tell you the answers! That's not helpful.

Questions?

- All course details and policies, as well as a tentative schedule are in the course syllabus, available on the web site