Computer Science I

CSCE 155E

Dr. Chris Bourke

Are you in the right class?

- CSCE 155E: Computer Science, Computer Engineering, Electrical Engineering
- CSCE 155T (Humanities focus)
- CSCE 155N (General Engineering)
- CSCE 155H: Honors (Java *and* C)
- CSCE 156: Computer Science II (for those with extensive experience)

Instructor, GTAs:

Learning Assistants:

What is Computer Science?

Course Mantras

"If you really want to understand something, the best way is to try and explain it to someone else. That forces you to sort it out in your own mind… that's really the essence of programming. By the time you've sorted out a complicated idea into little steps that even a stupid machine can deal with, you've certainly learned something about it yourself." --Douglas Adams, Dirk Gently's Holistic Detective Agency

"In my experience, you assert control over a computer–show it who's the boss– by making it do something unique. That means programming it… If you devote a couple of hours to programming a new machine, you'll feel better about it ever afterwards" --Michael Crichton, Electronic Life

What is Computer Science?

- At its core: Problem Solving
- Computers do not solve problems, *you* do
- Computers only automate solutions
- A discipline that is becoming ancillary to *all* other disciplines

"The world of A.D. 2014 will have few routine jobs that cannot be done better by some machine than by any human being. Mankind will therefore have become largely a race of machine tenders. Schools will have to be oriented in this direction. All the high-school students will be taught the fundamentals of computer technology, will become proficient in binary arithmetic and will be trained to perfection in the use of the computer languages that will have developed out of those like the contemporary Fortran." --Isaac Asimov, 1964

What is Computer Science?

What is CS1?

- Introduction to Computer Science
- Not *really* a programming course
- Programming is a *tool* we will use
- About: Problem Solving; Computational Thinking, Algorithms, data structures data processing, etc.
- The world is process; the world is computation (biological, physical, economic, engineering, etc.)

What is Computer Science?

Essential Topics:

- Mastery of the fundamentals of programming in a high-level language
- Problem solving methodology (specifications, design, refinement, testing, prototyping, best practices, etc.)
- Software Development principles
- Data: abstraction, modeling, processing
- Control structures: conditionals, repetition, error handling
- Algorithms: design & analysis, searching, sorting, recursion
- Exposure to Graphical User Interface & Event-driven programming
- Exposure to Databases and DB access

Course Structure

Modules

- Course is organized into *modules*, roughly 1 per week
- Each module covers a major topic/subtopics
- Each has:
    - Assigned Reading
    - Videos
    - Lectures
    - Lab
    - Hack

Course Structure

Readings

- Free textbook (mine): http://cse.unl.edu/~cbourke/ComputerScienceOne.pdf
- Read relevant sections *before* the module starts, *before* coming to lecture
- Many supplemental articles

Videos

- Videos: 59 videos, 9.5 hours of material
- 5 videos per module, 9 minutes each
- Watch before class
- Presents a different perspective/different exercises
- Watch as a review after lectures

Course Structure

Lectures

- Every Monday & Wednesday
- Attend *every* one!
- Lectures are "coded" in markdown and will be posted to piazza
- *You still need to take notes*
- Limit distractions:
    - Numerous studies indicate phones/social media are detrimental to learning
    - Even the mere presence of a smartphone makes you dumber
- Bring laptops, but use them wisely

Course Structure

Labs (15 * 10 = 15%)

- Weekly labs (Tuesdays)
- Peer programming: randomly assigned pairs
- Navigator: in charge of handout and discussing what needs to be done
- Driver: in charge of the keyboard
- Both: responsible for communicating and solving the problem
- Graded on completion, must be completed *during lab*
- Complete the code, worksheet and instructors will sign you out

Course Structure

Hacks (25 x 14 = 35%)

- Weekly hack sessions (Thursdays)
- Completely open collaboration: work alone, pair up, work in groups
- Each student turns in their own copy (with credit to all involved)
- Due following Monday evening, 1 second before midnight
- Turn it in *electronically* using the webhandin
- Semi-auto graded using the webgrader
- Graded on style, documentation, design, correctness
- Rubric in Canvas

Course Structure

Assignments (25%)

- 5 Assignments
- No collaboration allowed
- Hacks prep you for assignments
- About one every 3 weeks
- Due Friday evenings
- Turn it in *electronically* using the webhandin
- Semi-auto graded using the webgrader
- Graded on style, documentation, design, correctness
- Rubric in Canvas

Exams

- One midterm (10%), one final (15%)
- Coding exercises
- Laptop required

Collaboration

Item	**Collaboration**	**Explanation**
Labs	Required Collaboration	You are randomly assigned a partner and required to collaborate with them for that lab. You *may* get help from other groups and you *should* help other groups but you should primarily work with your assigned partner.
Hacks	Open Collaboration	You *may* work with any number of other students as long as you document whom you collaborated with in your source documentation.
Assignments	Closed Collaboration	You *may not* work with any other student or individual in any way. You may not communicate with anyone during the exam.
Exams	Closed Collaboration	You *may not* work with any other student or individual in any way. You may not communicate with anyone during the exam.

Policies

- Syllabus
- No late work accepted
- Academic Dishonesty
  - Abide by collaboration policies
  - Code Usage Policy
  - You *will* be caught: MOSS

Course Resources

- Canvas: http://canvas.unl.edu
- Piazza https://piazza.com/
- Handin: http://cse.unl.edu/handin
- Grader: http://cse.unl.edu/~cse155e/grade
- Textbook: http://cse.unl.edu/~cbourke/ComputerScienceOne.pdf
- Graduate Teaching Assistants
- Learning Assistants
- Student Resource Center: http://cse.unl.edu/src

How to Succeed in this Course

- Don't assume anything about yourself
  - We start with zero knowledge
  - Even if you have > zero knowledge, you will still learn something
  - Coding is a *skill* that needs *practice*
  - Compare yourself to yourself, not to others
  - All *may* struggle
  - All *can* succeed

Surveys & Studies

- Faculty Learning Community is:
  - Filming the first few weeks
  - Asking that you complete a (short) survey
- Other surveys may be administered

Setup and Demonstration

- Accounts: http://cse.unl.edu/claim
- BYOD Document
- Windows users: you need an SSH client; putty: http://www.putty.org/
- Mac users: use terminal
- Recommendation: https://atom.io/
- Connect to your "Z" drive: https://cse.unl.edu/faq

Demonstration

```c
#include<stdio.h>

int main(int argc, char **argv) {
  printf("Hello World!");
  return 0;
}
```

Compile:

```
gcc hello.c
```

Run:

```
./a.out
```

Why C?

- Lingua franca of programming languages
- Perfect to understand the fundamentals of how programs work
- You will *have* to learn *many* more languages
- 40+ years old, but still commonly used (and for good reason)
    - TIOBE
    - IEEE
    - PYPL
    - Aggregate Data