CS – 220
Computer Systems II

Instructor: Tom Bartenstein

Course Web Page:
http://www.cs.binghamton.edu/~tbartens/CS220_Spring_2019/

Teaching Staff

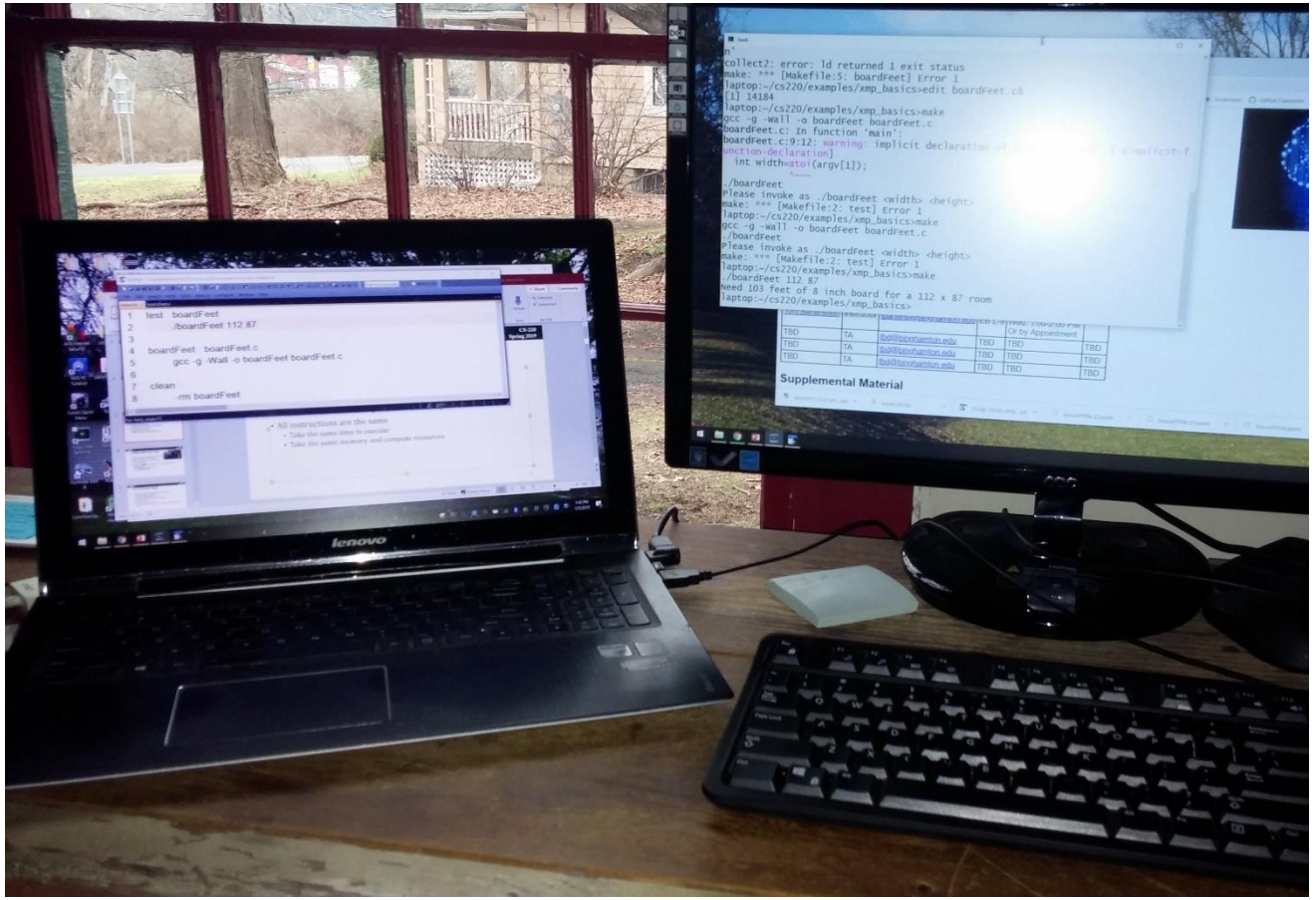
- Prof: Tom Bartenstein (tbartens@binghamton.edu)
  - Office Hours: Posted on Class Web Page
- Teaching Assistants
  - TBD

Programmer's View of Computer

- Executes abstract instructions in a programming language

  if (x>3) y=sqrt(x);

- All variables are the same
  - Same access time, same space, same availability
- All instructions are the same
  - Take the same time to execute
  - Take the same memory and compute resources



Looking Under the Hood

- The programmer's view hides (abstracts) low-level implementation details
- We can write software without knowing all those details, but
- We can write ***better*** software if we know how things work!
  - abstractions leak
  - For example, look at array access
    See examples/xmp_cache

CS-220 Course Goals

- Goal: Learn how to write good application software
  - What is “application software?”
  - What is “good” application software?
- Goal: Teach yourself new computer languages
  - C, x86 assembler
- Goal: Motivate other CS topics
  - Languages, Operating Systems, Algorithms, Data Structures, Architecture, etc.

CS-220 Outline

- Whirlwind Tour of C
  - Assumes some experience with C
  - Assumes some self-learning
- What makes C work?
  - Number Representation: Internals of numbers
  - Machine architecture: X86
  - X86 Assembler
  - Translation from C to X86
  - Function Call mechanisms
  - Link Edit and Load
  - Cache Memory and Virtual Memory
  - Heap Management
  - IO and Interrupt Mechanisms

Abstraction

"The act of considering something as a general quality or characteristic, apart from concrete realities, specific objects, or actual instances."

Abstraction in Computer Science

- Concrete – This soccer ball  Abstract – "ball"

- Abstraction is at the heart of all computer science!

Numbers	Data Structures	Operations
Languages	Algorithms	Objects
Functions	Memory	Disk I/O
Operating Systems	Graphical User Interfaces	Keyboards
Et cetera	Et cetera	Et cetera

"Leaky" Abstractions

- Sometimes we need to know concrete details about abstractions
- For instance, numbers are abstract:
  - Properties: Infinite, associative, commutative, distributive, well ordered
- Computer Arithmetic sometimes "leaks"
  - (x+1) < x (see xmp_counting)

- The idea of leaky abstraction was stolen from Joel Spolsky…
  http://www.joelonsoftware.com/articles/LeakyAbstractions.html
  Co-founder of https://stackoverflow.com/

Course Mechanics

- Lectures : Mon/Wed/Fri
  - Attendance Expected
  - Added incentive: Pop Quizzes (up to 3 or 4 over the semester)
- Labs: Tuesday (5 sections) in G-7
  - Attendance Expected – Should be able to finish during lab period
  - Practice coding, testing, debugging
  - Lab submissions due Tuesday at Midnight (11:59 PM)
- Homework
  - Reading
  - Practice – problem solving and coding, typically due Sunday at Midnight
- Projects
  - 4 Larger Coding projects: Lowest grade will be dropped

Lab (G-7) Access

- G-7 is Key Card access controlled
- All CS-220 students should have G-7 access (if not let us know)
- Use G-7 during your lab period
- You may use G-7 whenever there are no labs
  - Schedule posted outside the room and online

Class Web Page

- PDF of lecture slides
- Lab Instructions
- Homework Instructions
- Project Instructions
- Example C Programs
- Reference Material

Attendance Policy

- Attendance will not be taken at lectures or labs
- If you miss a lab or lecture, please send an e-mail BEFORE the lab or lecture starts to tbartens@binghamton.edu
  - Not looking for an excuse, just notification
- Occasional pop quizzes
  - No make-up.
  - If you are excused (sent an e-mail) Quiz doesn't count
  - If you are not excused and miss the quiz, you get a zero on the quiz
- If this policy is abused, it will have to be changed!

Textbooks

- Randal E. Bryant and David R. O'Hallaron,
  - "Computer Systems: A Programmer's Perspective, Third Edition" Prentice Hall, 2015
  - http://csapp.cs.cmu.edu

- Brian Kernighan and Dennis Ritchie,
  - "The C Programming Language, Second Edition", Prentice Hall, 1988

Grading

Quizzes, Attendance, Participation*	10%
Homework	10%
Labs	10%
Projects	30%
Tests	20%
Final Exam	20%

*Your final average will include components that are not on blackboard.
The mapping from a numeric average to a letter grade is not pre-defined, but will be determined based on class performance and comparison to previous CS-220 classes.

Academic Honesty

For Yourself!	For your peers	For me	For the institution
College education is not cheap, don't waste it!	One person cheating hurts all other students	Cheating wastes time and effort	Our reputation depends on you!

Academic Honesty Policy

- The Watson School has an academic honesty code.
- It is your responsibility to read and understand it.
- When in doubt, ask beforehand!

https://www.binghamton.edu/watson/student-services/advising/undergrad-forms/honesty-policy.pdf

Academic Honesty: HW, Labs, & Projects

- Unless otherwise noted, assignments are individual assignments. This means that all work submitted will have been done by you.
- You may (and are encouraged to) seek help from others, including the instructor, TAs, and classmates. Help includes assistance with:
  - using the software tools needed to complete an assignment
  - understanding the specifications or requirements of an assignment
  - evaluating strategies for solving a problem
  - debugging code that you have written
  - interpreting compiler and run-time error messages
- You may NOT copy code!

Academic Honesty Pledge

I pledge to uphold the Student Academic Honesty Code for all work completed for CS-220, Computer Systems II, in the Spring Semester, 2019. This includes, but is not limited to:

1. I pledge not to present the work of another person as my own work, including computer code, words, ideas, information, data, or style of presentation taken from other students, the internet, books, periodicals, or other sources. This includes:
   - Quoting, paraphrasing, or summarizing even a few phrases or lines of code without acknowledgement.
   - Relying on another person's data, evidence, or methods without credit or permission.
   - Submitting another person's work as my own.
   - Using unacknowledged material gathered by someone else.
2. I pledge not to give or receive unauthorized help before, during, or after tests or examinations. Examples include:
   - Collaboration of any sort during an exam.
   - Reading an exam before it has been given.
   - Unauthorized use of notes, books, tapes, computers, cell phones, or other aids during an examination.
   - Looking at someone else's examination during the exam period.
   - Allowing another person to look at my exam during the examination period.
3. I pledge not to collaborate on projects, assignments, computer programs, or labs unless explicitly authorized by the instructor.
4. I pledge not to misrepresent facts related to my academic performance, such as misrepresenting or fabricating material, or facts related to the lateness of assignments or absences.
5. I pledge not to deliberately destroy the work of other students, such as destroying another student's computer program or project, or interfering with the operation of a computer system so that it has a negative effect on the academic performance of others.