



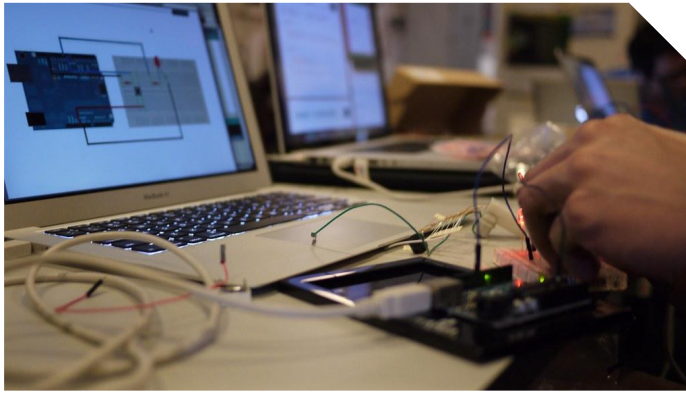
CoE 164

Computing Platforms

00: About the Course

HELLO!

Welcome to one of the laboratory components of your core CoE courses!



ICE BREAKING SESSION!

Check the polls!



WHAT YOU ALREADY KNOW...

- EEE 111
 - Basic programming
 - Computations
- EEE 137
 - Probability and statistics



WHAT YOU ALREADY KNOW...

- Math 40
 - Linear algebra
 - Matrix operations
- EEE 121
 - Basic programming
 - Data structures and algorithms
- EEE 153
 - Computer organization
 - Memory, cache, etc.



WHAT YOU ARE CURRENTLY STUDYING...

- CoE 161
 - Information theory
 - Computational complexity theory
- CoE 163
 - Computer algorithms and hardware



WHAT ARE GOOD TO KNOW...

- Knowledge of several programming languages
 - C/C++
 - Python
 - Matlab/Octave
 - Assembly (MIPS)



WHAT YOU'LL LEARN...

- How do we implement algorithms with efficiency in mind?
- Which tool(s) do I use to solve a computer engineering problem?
- Which tool is better in solving my problem?



COURSE SEGMENTS

- Rust programming
- [CoE 161] Introduction to Information and Complexity
 - Compression algorithms
 - Data transmission over different mediums
 - Turing machines
- [CoE 163] Computing Architectures and Algorithms
 - Program Profiling
 - Linear algebra operations
 - Parallel and GPU programming



RUST PROGRAMMING

- Rust is a programming language designed to write fast and safe code.
- It can be used to write systems
 - Websites
 - Embedded
 - Native applications



INFORMATION AND COMPLEXITY

- Information theory deals with quantization of information and ways to manipulate it.
- Computational complexity theory deals with ways to analyze program metrics and their general mode of operation.



COMPUTING ARCHITECTURES AND ALGORITHMS

- We can write more efficient programs and algorithms with the following knowledge
 - Complexity theory
 - Profiling
 - Concurrency
- We can exploit hardware to do the same
 - Parallel programming
 - Computer organization



SURPRISE REVIEW QUIZ!

Check the polls!



CLASS ARRANGEMENT

CoE 164 is **hybrid**, meaning sessions will be held both onsite and offsite (not fully hyflex).

Any software exercises will have **deadlines** usually a week after release.



LEARNING TOOLS

- Decent internet connection
 - Exercise submission
- Access to a desktop, laptop, or smartphone
 - Programming exercises
 - Capstone exercise
 - Will try our best to either:
 - Make exercises solvable on slow computers
 - Lend a remote server (VPS) or platform for your programs
- Better to have a camera and microphone for synchronous meetings, if any



CLASS MATERIALS

- Slides/Study guides
- Resource/reading links
 - UP Microlab wiki
- Forums
 - UVLe
 - Piazza

GRADING RUBRIC

Academic requirements:

- 50% Software exercise (SE)
- 20% Midterm problem (MidP)
- 30% Capstone problem (CP)

Completion requirements:

- Submit half of SEs
- Submit MidP
- Submit CP

We do not give 4.0s!

LATE SUBMISSIONS

We accept late submissions for everything until **Saturday, 01 June 2024** (end of classes), but...

- May have deductions
- May be evaluated late

INSTRUCTOR INFORMATION

Carl C. Dizon

Lecturer

ME Electrical Engineering

carl [dot] dizon [at] eeemail

... or forums via UVLe (or
Piazza)!

OTHER INSTRUCTORS

CoE 164 is co-taught with one other instructor from CARE:

- Carl Lester V. Fabian



OPEN FORUM

Enjoy and good luck with the course!





CoE 164

Computing Platforms

00: About the Course