Introduction

Prof. Eric Polizzi
Teaching Team and Logistics

- **Lead Lecture + Discussions/Projects:**
  - Prof. Eric Polizzi (polizzi@ecs.umass.edu)
- **Teaching Assistants (PhD students):** project graders, discussions support
  - Fubao Wu (fubaowu@umass.edu)
  - Guoyi Zhao (guoyi@umass.edu)
  - Xuzhi Zhang (xuzhizhang@umass.edu)
- **Discussions (3 sections)** – start next week
  - Thursday 10-11:15; 11:30-12:45; 2:30-3:45pm
- **Office hours** – start next week
  - **TA:** Mon. 4-5:30, Marcus 228; Tue. 10-11:30, 1-2:30pm, Marcus 209
  - **E. Polizzi:** Fri.11:10-11:40 and 1-1:30pm, Marcus 201C
- All technical e-mails regarding projects and HW, to be sent to TA
- TA can answer your project questions during office hours and discussions (come prepare with specific questions)
- My office hours and email are for any non-technical concerns (I will never look at your code)
- Short questions on lecture only welcome after class
Who Am I?

- **Professor in ECE & Math:** I am a ‘Computational Scientist’ working on:
  - **Computational Nanoelectronics** – advanced quantum mechanics
  - **High-Performance Scientific Computing** – advanced applied mathematics and numerical parallel algorithms
- **34 years** of programming...

**Others:** Je parle Français
Motivations

- **What will you learn in 242?**
  - How to design data structures for efficient use (from basic to complex)
  - How to think about data and operations on data
  - Basic algorithms and operations on database
  - How to determine the efficiency of an algorithm

- **Why does it matter?**
  - Practice Programming: a key enabling technology
  - Learn both fundamentals concepts and practical strategies
  - Interview preparation (jobs, internships, etc.)
  - Become a billionaire
  - Useful and Fun!

### Planning

**Grading:** 6 Projects (33%), Mid-Term (28%), Final (36%), Participation (3%)
Projects, HW, Class Notes, Textbook pages will be posted on-line on my website. Important Announcement will be sent by e-mail.

Exams:
- Class notes + HW + projects; Final is cumulative

Projects:
- To do it alone- No sharing of code (automated checking). Otherwise, you will get a “F” for this course. Please read Academic Honesty Policy of the University carefully
- One project every two weeks- to be uploaded on Moodle in zip file
- Deadline is 11:59pm on the due date- Late submission accepted until the next day at 10am (-10 points penalty). After that, system is locked, late submission will never be accepted.

Warning: No Extension, No Exception policy.

Warning: Projects are time consuming, start day 1

Read Syllabus entirely- understand expectation for this class
ECE 242 – Lessons learned

- Grades (previous terms):
  - Mid-term averages: ~56.5  Final average: ~58.5
  - Projects average: ~77  Total average: ~64
  - (grading on a transient loop) https://www.youtube.com/watch?v=Y0K_cf49gCM
    A at ~85/80, A- at ~75, C in the ~55

- Word of Wisdom from your colleagues:
  - Do the homework, and practice writing your own code for every concept covered in class before the exam.
  - don't skip class half the semester
  - Study hard for the exams and get started on the projects early
  - Attend every lecture
  - Start practicing sooner and study harder
  - Read over lecture notes after each lecture.
  - Don't start the projects so late.
  - Have more confidence in yourself, this class isn't nearly as hard as people say it is.
  - The exam will be hard, study more
  - Probably to start the projects earlier and start studying for the first exam earlier
  - Start working on the projects earlier.
  - Don't wait to start the projects
  - Do the homework, go to class more, and start the projects earlier.
  - Study extremely hard for the midterm.
  - Make sure to keep yourself on track. And don't worry as much.
  - I would do the homeworks more often and read the book as we did the material in class
  - get started on the projects a lot earlier because they are very tedious and time consuming
Textbook

- Textbook – Robert Lafore
  - Basic Java programming
  - Comprehensive and clearly written
- Include Java Applets
  - 1-Download “WorkshopApplets.ZIP”
  - 2- Test it (open using a “browser” or appletviewer software)- follow textbook instructions
- 3- Bring your laptop to class
Data Structures and Algorithms are independent of programming language

No such thing as “best programming language”; Particular language more convenient to use in particular situation/application

Main criteria: Portability, Sustainability, Performance/Optimization, Scripting or not, etc.

Two programming approaches:

1- Procedural (with or without data objects) – codes divided into methods (procedure, subroutines)- intuitive programming - Price to pay: flexibility

2- OOP – (objects contains both data and methods)- offers more flexibility for some real world applications – Price to pay: some abstraction, planned-ahead programming, performance
How to use Java

- Use an integrated development environment (IDE): Eclipse, Dr Java
- Use any editors (emacs, vim, etc.) and command line:
  - `javac HelloWorld.java`
  - `Java HelloWorld`
Every computer program uses **data structures and algorithms**

### Data Structures

**Why?** organize your data in computer's memory to efficiently store and retrieve **information**

**How?** using arrays, linked lists, stacks, queues, trees, matrix, etc.

### Algorithms

**Why?** manipulate the data in various way to perform **computation**

**How?** using strategy/method, operations and analysis

---

**Data-centric view of the world**
- Complex data structures
- Basic ‘non-numerical’ algorithms act on database

**High-Performance Computing**
- Basic data structure
- Complex ‘numerical’ algorithms
Data-centric view of the world

- **Data management to store and retrieve information**
- **Applications:** customer profiles, page rank, network traffic, DNA sequences, etc.
- **Making use of Non-numerical Algorithms:** Insert, Delete, Searching, Sorting, etc.
- **Challenges:** Data sets can be very large (Big Data)
- **Example:** Million song database
  
  labrosa.ee.columbia.edu/millionsong/

Data set of songs: title, artist, recording years, etc.

How would you figure out:
- Which artist has recorded most songs?
- Which song has been covered the most times?
- What are the most common words in a title?
High-Performance Computing

- **Modeling/simulation as a primary tool for scientific discovery and innovation in engineering**
- **Applications:** signal processing, modeling of electronic devices, climate, etc.
- Making use of **Numerical Algorithms** (solving mathematical equations – floating arithmetics): Root finding, numerical integration, Fourier transform, numerical linear algebra, etc.
- **Challenges:** scalability and parallelism - towards exascale computing - (Big Computing)
- **Example:** device modeling
DATA STRUCTURES

Array
(1d,2d,etc.)

Linked - List
(simple,doubly)

Unsorted list

Sorted list

Trees
(BST,CBT,RBT)

Hash-Tables

Stacks

Queues

Heap
(Priority Q)

Graphs

DATA STRUCTURES

General-purpose

Specialized
(ADT)

ALGORITHMS

Insert

Remove

Search

Sort

Traverse

Pop, Push, dequeue, enqueue, etc.
Next steps

- 1st lecture is Friday
- No discussion tomorrow
- Project 1 will be presented next week during discussion
- One week left to get up to speed with Java, and get ready for projects (how to compile/run/read from files/etc.).

Further Reading:
- Textbook – Chapter 1
- Comprehensive review of Java
  http://www.ecs.umass.edu/ece/ece122/