Introduction

Prof. Eric Polizzi
Teaching Team

- **Lead Lecture + Discussions:** Prof. Eric Polizzi

- **Teaching Assistants (PhD students):**
  - Braegon Spring
  - Fubao Wu
  - Guoyi Zhao
  project graders, discussions support, office hours

- **Lead Supplement Instruction:** Francesca Maroney
  Weekly learning review sessions
Who Am I?

- Faculty in the ECE and Math departments
- Research:
  - Nanoelectronics – advanced quantum mechanics
  - Scientific High-Performance Computing – advanced applied mathematics and numerical parallel algorithms
- Personal:
  - My first computer
  - My current computer
  - 32 years of programming
  - 25 lines of code every day
- Others:
  - Je parle Français
Motivations

- 1- Learn programming language, 2- Learn programming practices
  - How to think about data and operations on data
  - How to design data structure for efficient use
  - How to determine the efficiency of an algorithm
  - Basic data structures and algorithms
  - More complex programming techniques

- Why 242?
  - Practice Programming: a key enabling technology
  - Learn both fundamentals concepts and practical strategies
  - Interview preparation (jobs, internships, etc.)
  - Become a billionaire
  - Useful and Fun!

<table>
<thead>
<tr>
<th></th>
<th>Mon</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sep</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>23</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>30</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Oct</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>Nov</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Dec</td>
<td>7</td>
<td></td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

- **HW (not graded):** Every Monday- Solution in discussion the following Thursday
- **Grading policy:** 6 Projects (36%), Mid-Term (28%), Final (36%)
Syllabus and Policies

- Lecture (Goessmann 20): M-W-F 10:10-11:05
- Discussions Thursday (ELAB 307) and TA section leader
  - 10-11:15 Fubao fubaowu@umass.edu
  - 11:30-12:45 Guoyi guoyi@umass.edu
  - 1:2:15 Braegan bspring@student.umass.edu
- Office hours (you can go to any TA for help) – starting next week
  - Fubao, Guoyi, Braegam – TBD
  - Prof. Polizzi - TBD
- Learning review sessions – Francesca - TBD
- E-mail: TA will respond to technical e-mails (project/HW questions, etc.)
- Projects:
  - By group of two (recommended/preferred) or alone
  - One project every two weeks- to be uploaded on Moodle in zip file
  - **Warning:** Two weeks really means two weeks worth of work
Syllabus and Policies

- **Project, HW, Class Notes** will be posted on-line on my website. Important Announcement will be sent by e-mail.

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

- Any questions regarding the grading of projects/Mid-term should be raised to respective TA or Instructor within one week after the grades are announced. Otherwise the grades will become final.

- **Intellectual Responsibility**
  - Encourage discussion on class materials among each other.
  - Project/program has to be your own or your group. **No sharing of code** (automated checking of submissions). Otherwise, you will get a “F” for this course. Please read Academic Honesty Policy of the University carefully
  - Late submission is not accepted.
  - **Read Syllabus entirely** - understand expectation for this class
Syllabus and Policies

- 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
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 algorithms act on database
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?
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 algorithms act on database

**High-Performance Computing**

- Complex algorithms
- Basic data structure
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
Next steps

- **Upcoming Topics:** Arrays, Simple Sorting, Analysis, Queues/Stacks, Linked-List, Recursion, Advanced Sorting, Basic Numerical Algorithms, Matrix, Trees, Graphs, etc.
- HW1 posted (solution tomorrow)
- Project 1 will be presented and discussed tomorrow

Further Reading:
- Textbook – Chapter 1
- Review of Java