L01: Welcome to E7!

Class Overview and Syllabus

Lucas A. J. Bastien

E7 Spring 2017, University of California at Berkeley

January 18, 2017

Version: release

Topics for today

- Presenting instructor and GSIs; Say hello to your neighbors
- What is E7?
- What is computer programming? What is a numerical method?
- Examples of algorithms
- Class logistics: discussing key points of the syllabus

Instructor and head GSI

Instructor: Lucas Bastien

- Email: lucas.aj.bastien@berkeley.edu
- Office hours (535 Davis Hall):
  - Mondays: 9:30–11 am
  - Wednesdays: 4–5:30 pm
- Education: Ph.D. from UC Berkeley (Air Quality Modeling)

Head GSI: Matthew Vannucci

- Email: vannucci@berkeley.edu
- Education: Ph.D. candidate at UC Berkeley (Indoor Air Quality)

E7 GSIs this semester

- Hira Bakhsh
- Ninh Do
- Bradley (Brad) Harken
- Abdul (Aboudy) Kreidieh
- Jiang Jun (JJ) Lee
- Tiange (Tina) Li
- James Neher
- Ekaterina (Katya) Rakhmatulina
- Pearl Ranchal
- Andrew (A.J.) Santa Maria
- Jason Simon
- Demetra (Dema) Tzamaras
- Ethan Yen
- Lydia Yiu

“Hello world!”

Turn to your neighbors (forward, backward, left, right) and introduce yourself *e.g.*,

- Name
- Major
- Where you are from
- Previous programming experience?
- etc.

What is E7?

E7 is a computer programming course for engineers

First half of the course:

- Elements of programming needed to build your own programs

Second half of the course:

- Numerical methods to solve engineering problems

Programming language:

- Matlab. **Knowledge is transferable to other languages**

Prerequisite classes:

- Math 1A **and** 1B (1B may be taken concurrently)

Required programming experience:

- **None! E7 is an introductory class!**

What is: computer programming? a numerical method?

Computer programming consists of writing instructions that a computer can interpret, in order to perform a specific task

Examples of applications:

- Operating systems
- Web browsers, web sites
- Communication technologies
- **Science and Engineering**, for example:
  - Design mechanical systems *e.g.*, cars
  - Design structures *e.g.*, bridges; earthquake safety
  - Weather and climate forecasting
  - Develop renewable energy; Smart cities
  - Bio-engineering *e.g.*, prosthetic, model blood circulation

Numerical methods are used to calculate approximate solutions to [engineering] problems that are difficult to solve analytically

Algorithms

Algorithms are "recipes" used to accomplish specific tasks

For example, let us count the number of students in our lecture hall

Algorithm 1: the instructor counts the students one by one

Algorithm 2:

1. Stand up and assign yourself the number 1
2. Pair with someone who is standing, add your numbers together, adopt the result as your new number
3. One of you sits down, the other goes back to step 2
4. Stop when only one person is standing

Textbook

An Introduction to MATLAB Programming and Numerical Methods for Engineers by Timmy Siauw and Alexandre M. Bayen, Elsevier Inc, 2015.

- Written specifically for this class, in sync with lectures and labs through most of the semester
- Includes explanation of key concepts, examples, and useful tips
- Available online for free from the UC Berkeley campus
  - Instructions on how to access it will be posted on bCourses shortly

"To the students of UC Berkeley's E7 class: past, present, and future."

Software

Matlab is installed and available for you to use in some computer labs:

- 1109 Etcheverry Hall (dedicated to E7 this semester)
  - Open hours on **Fridays 8 am – 5 pm** (9 am – 12 pm with GSIs)
  - Also open **Monday/Wednesday 3 pm – 4 pm**
- 1535 Tolman Hall
  - Depending on schedule, more information on this topic later

Matlab is available to UC Berkeley students for free

- Instructions at http://software-central.berkeley.edu

Matlab is also available in other computer labs

- Check with the computing staff in your department

Lecture and discussion

Monday/Wednesday/Friday 2–3 pm, 155 Dwinelle

The teaching style incorporates lecture and discussion together

We will use **active learning** through class participation and interaction. This approach is chosen to **benefit you**!

Use of laptop computers is restricted to the $n = 10$ front rows. n may vary throughout the semester

Lab sections and lab assignments

Monday/Wednesday **or** Tuesday/Thursday (2×2-hour per week)

- Where you will practice programming on a computer and where you can get help
- **Only go to the lab section into which you are enrolled**
  Attendance will be taken in lab sections

Every section has two graduate student instructors:

- **Primary GSI is your main contact person**
- Second GSI assists primary GSI

Submit labs electronically via bCourses, due on Fridays at 12 pm

- Late submissions accepted during 2-hour grace period
- First lab posted January 19 (tomorrow) and due on January 27
- You can upload partial work and update your submission before the deadline

Lab sections and lab assignments (continued)

Lab assignments will typically require 6-8 hours of work

- Start labs early
- Come to lab section early, and with questions
- Don't wait until your lab section to start the lab assignments

Lowest two lab scores will be dropped for everyone

- In case of illness, personal/family issue, travel
- No questions asked, no emails necessary
- Email us only about exceptional situations (month-long illness, accommodation for disability)

Grading of lab assignments

Labs will be auto-graded

- You submit Matlab code via bCourses
- We check to see if your code gives correct results
- Test cases are provided so you can check your code
  - It is your responsibility to ensure that your code works in general, not just for the published test cases
  - Additional undisclosed test cases will be used for grading

You have to write your own code. We will check for cases of copied or edited versions of someone else's code. DO NOT COPY SOMEONE ELSE'S CODE!

Programming project

Graduate programs and employers often ask questions such as

- What things have you built/designed?
- What projects have you worked on?
- What teamwork experience do you have?

In the second half of the semester, you will work in small teams on a programming project (more details will be provided later)

The project code will be due on the last day of classes (April 28)

Course grade

Option 1:

- Participation (15%)
- Lab assignments (35%)
- Project (10%)
- Midterm exam (10%)
- Final exam (30%)

Option 2:

- No points for participation
- Lab assignments (35%)
- Project (10%)
- Midterm exam (15%)
- Final exam (40%)

Participation: ask at least one E7-related question during one of your lab sections each week (see syllabus for details)

We will calculate overall scores for everyone using both options, and, for each student, use the higher grade

Need 60% overall course grade to pass the class

Communication

You are responsible for staying up to date with the information communicated via:

- bCourses announcements
- bCourses messages
- email

We will always use our @berkeley.edu address to communicate with you via email

Please always use your @berkeley.edu address to communicate with us via email

Getting help

The material covered during the first few weeks of class will be used throughout the rest of the semester

- Stay up to date with the material from the very beginning!
- Seek help early!

Getting help

- Asking questions in person to GSIs during your lab section
- Asking questions in person to GSIs in 1109 Etcheverry Hall on Fridays between 9 am and 12 pm
- Consulting the lists of frequently asked questions (FAQs) in the bCourses Pages
- bCourses "Discussions" (or in person) for student-to-student help
  - Keep in mind the plagiarism rules!
- My office hours, but only for questions not related to assignments

Schedule and Next Steps

Next Lecture

- Friday January 20, 2–3 pm in 155 Dwinelle Hall

This Week

- Read syllabus (posted on bCourses)
- First lab assignment posted January 19 (tomorrow) and due on Friday January 27

Exam Dates

- Midterm: Wednesday March 1, 2 – 3 pm in 155 Dwinelle Hall
- Final exam: Tuesday May 9, 11:30 am – 2:30 pm
- Check **now** that you don't have final exam conflicts or too many exams on one day

Feedback at the end of most lectures/discussions

I would like your help to make this class better and better throughout the semester. I will ask for your feedback during the last few minutes of most lectures/discussions:

- Pack first (30 seconds)
- What went well? (1 minute)
- What could be improved? (1 minute)

Methods (we will see what works best):

- A few voluntary students give feedback verbally out loud
- All students can write feedback on paper, and drop in drop box