# CS100M/CIS121/EAS121

### Introduction to Computer Programming

Spring 2004 Lecture 11 MATLAB Advice

### Announcements

- Prelim 1 (from Syllabus):
  - retrieve in Section (lab)
  - will go over in lab as part of lab
  - leftovers in Carpenter B101 around Thurs
  - regrades? see Syllabus, deadline next Thurs
- Reminder about grades and grading
- A3 due 3/10 start soon!

2

## Advice

- The course
- Using the notes and solutions
- Using labs
- Doing homework/working with partners
- How to study
- Taking an exam

# **The Course**

- Are you keeping up with e-mail and on-line announcements?
- Are you attending office hours? Asking consultants for help? Making appointments?
- Have you checked out **Additional Help** on the course website?

1

# **Using Notes and Solutions**

- Are you going to lecture?
- Are you looking at the notes and posted examples?
- Are you reading the textbook and trying those Quizzes? (see pg 451)
- Are you asking for help if you get stuck on a particular problem?
- Are you using the on-line (extensive) version of MATLAB Help? (section programming?)

#### 5

# **Using Labs**

- Are you going?
- Are you trying to solve the problems before raising your hand?
- Are you reading the error messages and looking up syntax/semantics in MATLAB Help?
- Are you reviewing the posted solutions?

6

# Homework/Partners

- Are you actually doing the work independently? Are you relying too much on your partner?
- Are you spending time thinking about the design and writing algorithms before programming?
- Are you solving the problem by breaking it down into smaller parts and testing those parts?

# Studying

- Have you reviewed the notes, solutions, and review questions?
- Have you identified what you don't understand early?
- Are you practicing simple problems before tackling the longer, more challenging ones?
- Actually, are you practicing?
  - programming is a *skill*
  - so, you need to practice!

8

# **Studying (continued)**

- Some advice
  - write study sheet of language elements, syntax, example
  - memorize terms, rules
  - memorize simple examples ("templates")
- Drill small examples by principle
  - identify principles before solving, then solve
  - keep drilling until you can solve quickly, then increase difficulty
  - always solve by principles!
- Simulate test conditions!
  - 20-30 min/problem and drill!

9

# **Test Taking**

- Read instructions (in advance if possible)
- DIS's successful trick: Write down memorized examples *after* test begins (is this legal?)
- Skim entire test (sometimes hints for other problems)
- Problems sometimes give away algorithms
- Read comments, hints, specifications
- Try to calm down (breathe in and out slowly, think of something funny, ask yourself if the fate of Humankind depends on your performance....)

10