

CS100M/CIS121/EAS121

Introduction to Computer Programming

Spring 2004
Lecture 2
Introduction to MATLAB
Problem Solving

2

Announcements

- CS100M sections start next week:
BLUE ROOM, ACCEL in Carpenter!
- TA office hours start this week
- Web announcements are happening! Readings, notes, staff, consulting hours, CMS info: <http://www.cs.cornell.edu/cs100m/2004sp>
- Consulting reminder: B101 Carpenter
- Advance warning: Prelim 2

Objectives

- See some MATLAB
- Basic MATLAB operation
- Some MATLAB language
- Problem solving process
- Algorithms
- Pseudocode
- Advice

3

Programming?

- Demonstration!
- Problems and Solutions
- Need *code*! What is code?
- Let's see some MATLAB!
- How did they do that?

4

MATLAB

- Running: how?
- Help? Tutorials? Command-line Help?

5

MATLAB Windows

- Command Window
 - primary window
 - execute commands
- Other windows
 - Workspace
 - Command History
 - Current Directory
 - Launch Pad

6

Some MATLAB

- Appearance, control
- Values
- Arithmetic
- Logic
- Arrays
- Functions
- Plotting

7

Some MATLAB

- Notes:

8

MATLAB Programs

- **M-File**
 - MATLAB Program
 - Two kinds: *scripts*, *functions*
- **Scripts**
 - collection of sequential MATLAB commands (top to bottom)
 - To create script: File→New→M-File
- **Functions**
 - special kind of script that has a name and can be called

9

Directory and Path

- MATLAB needs to “know” where to find files
- Directories/Folders/Files/Path:
 - **Current directory**: directory on computer where MATLAB looks for file
 - **Search Path**: all directories that MATLAB looks for file
- Your intended program must be in current directory or search path!

10

Problem Solving Process

- What you now know:
 - Have problem? Find solution!
 - Use language “building-blocks” for code!
 - Problem → ? → Code
- What is ?
 - Thinking
 - Writing
 - more?

11

Writing Process

- brainstorm
- research
- outline
- draft
- rewrite
- polish
- review
- repeat

12

Example

- Suppose a device randomly generates a random number (1 to 5, inclusive) of *bytes* worth of *bits*.
- Note: *bits* = 0 or 1. One *byte* has 8 bits. Assume each bit has equal probability of being generated.
- Write a program that performs one simulation and reports largest number of successively generated 1s.
- e.g., if program generates 001111001011110, it would report ____.
- I have a question....

13

Brainstorm

- Do NOT start writing the program!
- Think about the problem, jot ideas down, and let your mind wander...
- Skim problem and see what strikes you
- Sooner to code, the longer it takes!

14

Research

- Find **nouns**
 - input? values?
 - output? other *specifications*?
- Find **verbs**
 - what does program do?
 - specific actions to solve problem?

15

Outline

- Formulate a solution
- Think recipe!
 - ingredients: nouns
 - steps: verbs
- Is result “edible”?
 - test solution
 - might need to iterate the design

16

Algorithm

- **Algorithm:** sequence of steps to solve problem
 - must be instructive, correct, and efficient
 - must consider level of detail
- **Stepwise refinement:**
 - break big problems into smaller chunks
 - why? small problems easier than big problems!

17

Pseudocode

- How to write algorithm?
 - Use specific language with verbs that resemble code
 - **pseudocode:** human-language representation of code
- Common terms: *set, update, increment, for, if, else, otherwise, do, repeat, report, print, get, put, while, unless, until, stop*, and more!
- Examples?

18

Example Outline

```
Set currentMaxLength to zero
Generate one bit
Increment count of bits
If not finished (exceed total), continue
  If bit not 0, increment currentLength
    Swap maxLength with currentLength
  Else bit is 0
    Reset currentLength to zero
  Get next bit and repeat
Else finished
  report answer
```

19

Draft, Rewrite, Polish, Repeat

- Need to write program...
- Need to document code...
- Need to test and debug it...
- Need to submit to client for review...
- Might need to iterate solution...
- No coding for now! (must learn more concepts)
- But, what are these terms?

20

Some Programming Lingo

- **compile**: check program for correct language
- **run**: to execute a program
- **bug**: mistake in program
- **debug**: attempt to fix program
- **hack**: lots of meanings...
- see Programming on website

21

Collective Wisdom

- When doubt, check it (the website) out.
- The sooner to computer, the longer it takes.
- Practice makes perfect. Programming is a skill!
- Programming is also an art and theory!
- Tinker! What's the worst that can happen?
- Dig for answers.
- Start early! Otherwise, save now, but suffer later.

22