EG 1011: Module 2
Introduction to Software Design and Programming

Lecture 1
Introduction

Meet Dr. Striegel

- Assistant Professor
  - Comp Sci & Engr. (CSE)
- Teaching
  - Networking, security, programming, embedded systems
- Research
  - Computer networks
  - Make YouTube fast
  - Computer security
    - Keep my network safe from l33t HaXX0RS!

383 Fitzpatrick Hall
Phone: 631-6896
striegel@nd.edu
aaron.striegel@gmail.com
http://www.cse.nd.edu/~striegel

Finding My Office

Office Hours:
T 10, W 11, stop on by

Fitzpatrick Hall
DeBartolo
Cushing
LAW
Construction Site

DeBartolo
You are here!
Mood Test

From Dr. Flynn of CSE
Which of these Seven Dwarves are you today?
1. Sleepy
2. Grumpy
3. Sneezy
4. Bashful
5. Happy

Note: nobody is Dopey, and I am Doc.

One slide motivation

- Engineers use tools to help people.
- Software (and, more generally, computing) is yet another tool

Computing

- A tool for manipulating representations (descriptions of things or ideas) efficiently and in detail
- Not just desktops / servers
- Tool for input (sensors)
  - Temperature, image,
- Tool for control (actuator)
  - Heat controllers, camera pan-tilt-zoom rigs, control rods
Computing and Engineering

• Computers are fundamental
  – Very hard to find systems without them
  – Roles can vary (using vs. programming)
• Programming skills are broad skills
  – Many are identical to engineering skills
  – But computer software/hardware is brittle; you can’t over-engineer it for a “safety margin”

What will you learn?

• Mapping a problem into a series of solvable steps
• Implementing software solutions in a platform (MATLAB)
• Effective software development techniques
  – Problem Decomposition
  – Programming Pragmatics
    • MATLAB syntax (and quirks)
    • Control structures: loops, conditionals
    • Subprograms: functions and scripts
    • Input/Output (I/O): graphical plots, files
    • Debugging and testing

Assessment

• Homework sets x4
• Quizzes x3
• Learning Center assignments x5
• A non-trivial project
  – Space station simulation
• Programming challenge
• Both Individual and Group work
The textbook

- Pretty good text for this course
- We won’t cover all of it
- We’ll jump around a bit
- Lectures will not be based completely on the book

How to do well

- COME TO CLASS
- Contribute effectively to Learning Centers
- Do the homework assignments
- Seek help when you need it… from the prof, the staff, the student assistants, your classmates
- Take your project seriously (you’ll see many more projects before you graduate… it’s what we do)
- Complete exam
  - Programming challenge
  - Written exam

The Honor Code

- Read the Honor Code from du Lac (it is also on the web)
- EG10111: Acceptable Cooperation on Assignments and Projects means
  - You are encouraged to learn together
  - You are encouraged to help each other to seek deeper understanding of the course material
  - You are encouraged to seek help for concepts you do not understand or for programming problems that are frustrating you
  - BUT . . .
• All work turned in under your name:
  – Must Be Your Work
  – Must Provide Citation For Assistance
  Obtained Or Any Collaborations That
  Contributed Directly To The Solution
  – Must Be Understood By You
  – Cannot In Whole Or In Part Be A Copy, Or
  Otherwise Duplicated Version, Of The Work
  Of Another Unless Proper Citation Is Provided
  Within The Body Of The Work

Danger!

• All students involved in improper duplication of a
  program will be considered equal contributors to the
duplication (whether they were original authors of the
program or copied the program from the original
author)
• Violations of these guidelines will be consider a
  violation of the Honor Code and we will follow its
  process scrupulously.

Danger!

• If your collaboration with other students has led to
  your being in a “gray area”, or if you don’t know
whether a proposed collaboration is acceptable…
ASK Prof. Striegel, Ms. Meyers, or Dr. McWilliams
(and nobody else)!
• We would rather deal with a situation proactively (and
  “reward” your ethical response) than be forced into a
reaction (Honesty committee…)


Overview of MATLAB and Programming

• Chapter 1 in text
• Looking ahead
  – Homework 1
  – Learning Center 1

What is MATLAB?

• A programming language?
• An application? (like Word?)
• As you know, you can do useful work with MATLAB without "writing programs".
• But it supports programming (including valuable/nontrivial programming) very nicely

MATLAB advantages

• High flexibility, medium performance
• We will use MATLAB here because:
  – It is an easy environment in which to learn some of the basics of programming
  – We do not need to compile, then run, programs
  – Good help structure
  – Good error reporting
Transferable concepts

• Software design (decompose, integrate, test…)
• Debugging
• The concepts of language vocabulary, syntax, and constructs
  – These concepts transfer regardless of language
  – Same in C, C++, Java, Perl, Python, …

Poll: What is your current level of experience with computer programming?

1. I am proficient at programming
2. I am good at programming
3. I have done some programming
4. I have done little or no programming

Let’s try some MATLAB

• Find it
• Start it
• Wait…
• Look at the “environment” (pp. 5-10)
  – Command window, menu bar, editor, help facility
• Do some math! (pp. 10-15)
  – MATLAB is a great text-oriented calculator
  – For simple math, it “just works”
  – *, /, +, -, ()^, precedence
  – Simple functions: sqrt(), sin(), exp(), abs(), log()
  – Semicolon ;
Variables

- A variable is a “container” you can create to hold a value
- It has a name (e.g., x) (Must start with a letter)
- It has a value (e.g., 14) (Not necessarily a number!)
- The value can be changed
  - x = 14;
  - x = x+1;
  - x = ’hello’;
- Type a variable’s name (no semicolon) to print its value

Examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>x = 14;</td>
<td>x = 14</td>
</tr>
<tr>
<td>x = x+1;</td>
<td>x = x+1</td>
</tr>
<tr>
<td>x = ’hello’;</td>
<td>x = ’hello’</td>
</tr>
<tr>
<td>x = 14;</td>
<td>x = 14</td>
</tr>
</tbody>
</table>

Variable housekeeping

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>who</td>
<td>clear</td>
</tr>
<tr>
<td>whos</td>
<td>clear x y</td>
</tr>
</tbody>
</table>

>> who
Your variables are:
  x
>> whos Name Size Bytes Class Attributes
  x 1x1 8 double
>> clear
>> who
>>
On-line help

- `help commandname` prints out information about the named command.
- `lookfor concept` looks through the MATLAB command descriptions for the word "concept" in the first line of the command description
- Great ways to figure out or find commands
- And, of course…. Google is your friend

Problem solving

- The "hacker's" way
  - Start with a vague idea of the task to be solved with software
  - Grind out code, and code, and more code
  - Rip out and rewrite the code that
    - Should not have been written
    - Was not relevant
    - Reflected poor choices
    - Was poorly organized for debugging
  - Repeat previous step… forever.

- The engineer's way
  - Understand the problem
  - Decompose the problem into manageable chunks and interactions (document)
  - Decompose chunks if needed (document)
  - Implement solutions to small problems (document)
  - Integrate the solutions into a larger solution (document)
  - Test and revise (document)

Think before coding!
• “The basic problem in programming is the management of complexity . . . Program development should begin by focusing attention on the problem to be solved, postponing considerations of architecture and language constructs.”


Questions?