Transition to Matlab

CS1132 Fall 2012

Dr. K.-Y. Daisy Fan
Ankit Arora

http://www.cs.cornell.edu/courses/cs1132/2012fa/lec

Today's agenda

- Course goals
- Organization and requirements
- Logistics
- Matlab desktop
- Script vs. function

Course Goal

Learn how to program in Matlab!

Topics:

- Matlab basics (environment, built-in functions)
- Arrays (vector, matrix)
- Vectorized computation
- Control flow (if-else, loops)
- User-defined functions
- Strings and cell arrays
- Graphics
- Basic I/O (including file I/O)

Programming Fundamentals …

… that you should practice

- Top-down design
- Modular program development
  - Reduce redundancy
- Concise documentation
- Thorough testing

Must-know facts about CS1132 – Lecture course

- Course ends after 5 weeks: September 27th
- 1 credit, S/U
- Requires mastery of material
  - An average of B (85%) is required to pass the course
  - Average based on four items: Assignments 1 & 2 and Tests 1 & 2
- Assignments
  - If first submission is not perfect, one re-submission is allowed without penalty. Any additional, allowed, *re*-submission incurs a 10% deduction.
  - Late (re)submission allowed up to 24 hrs for a 10% penalty
  - Penalties accumulate from (re)submission to resubmission
- Tests
  - You may take each test a second time if you wish—a different version will be given
  - Final score is the most recent score for each test

Course staff and help

- See office hours listed on the syllabus
- Consulting available SMTWR 5-10pm in Green Rm of ACCEL (Carpenter Hall, former Engineering Library)
- Some of the lab sessions will be “help sessions” for assignments
- Review session will be given during class time before each test

Academic integrity

- Electronic submission does not alter the University standards on academic integrity
- Your individual work is required
  - Do not copy code from any source (friend, published work, Internet, …)
  - You can discuss general strategy with others, but do not share any code, whether written, electronic, or verbally
  - We use MOSS or similar software to check your submitted programs

What to do now (very soon)?

- Check course website
- Log on CMS (start from course website). If CS1132 LEC2 is not listed under "*My courses*," then you need to be added to the system. Email Ankit Arora <aa545@.cornell.edu> to request this and indicate your Cornell NetID.

Now demonstrate…

- Course website
- Matlab environment
- Example script
- CMS

Matlab Demo

- Command window, memory (Workspace) window
- Current folder box
- Built-in functions
  - `rand`, `floor`, `ceil`
- Example script diffArea
- Change script to function
- Variables are not declared: create a variable simply by assigning a value to a variable name. A number has the type `double` by default

Input & output

- *variable* = **`input`**('*prompt*')

- **`fprintf`**('*message to print*')

Input & output

- *variable* = **input**('*prompt* ')

```
r= input('Enter radius: ')
```

- **fprintf**('*message to print* ')

```
fprintf('Increase ')
fprintf('is %f inches\n', x)
fprintf('Position (%d,%d)\n', x,y)
```

Substitution sequences (conversion specifications)

%f	fixed point (or floating point)
%d	decimal—whole number
%e	exponential
%g	general—Matlab chooses a format
%c	character
%s	string

Examples: **%f** **%15.2f**

General form of a user-defined function

```
function [out1, out2, …]= functionName (in1, in2, …)
% 1-line comment to describe the function
% Additional description of function

    Executable code that at some point assigns
    values to output parameters out1, out2, …
```

- *in1*, *in2*, … are defined when the function begins execution. Variables *in1*, *in2*, … are called function *parameters* and they hold the function *arguments* used when the function is invoked (called).
- *out1*, *out2*, … are not defined until the executable code in the function assigns values to them.

18

Function header is the "contract" for how the function will be used (called)

You have this function:

```
function  [x, y] = polar2xy(r, theta)
% Convert polar coordinates (r, theta) to
% Cartesian coordinates (x,y).  Theta in degrees.
…
```

Code to call the above function:

```
% Convert polar (r1,t1) to Cartesian (x1,y1)
r1= 1;   t1= 30;
[x1, y1]= polar2xy(r1, t1);
plot(x1, y1, 'b*')
…
```