CS368 MATLAB Programming
Lecture 11

Young Wu

Based on lecture slides by Michael O'Neill and Beck Hasti

April 13, 2022
Errors
Math

Syntax error is an error in spelling or grammar.

MATLAB displays red messages for syntax errors, so they are easy to find and fix.

Semantic error is an error in meaning or logic.

For small programs, compare the program outputs with expected outputs computed by hand to find and fix the semantic error.

For large programs, break into smaller programs and debug each one.
A debugger can set break points.

A break point stops the program so that the current variable values can be viewed in Workspace.

It is useful to check if loops and conditionals are written correctly.
User Input Validation

Code

- Use a while loop to keep asking for user input until the input is valid.

```plaintext
1. valid = 0;
2. while ~valid
3.   x = input (...)
4.   if ... % check valid
5.      valid = 1;
6.   end
7. end
```
Input Validation Membership Functions

Code

- `isletter (x)` or `isstrprop (x, 'alpha')` checks if the string `x` contains characters that are letters 'a' to 'z' or 'A' to 'Z'.
- `isstrprop(x, 'digit')` checks if the string `x` contains characters that are numbers '0' to '9'.
- `ismember(x, y)` checks if the string `x` contains characters that are characters in the other string `y`.
- These functions treat `x` as a vector and returns a Boolean vector specifying whether each character in `x` belongs to some set of characters.
Input Validation Substring Functions

Code

- `contains(x, y)` checks if the string $x$ contains a substring $y$.
- `count(x, y)` counts the number of occurrences of $y$ in the string $x$.
- `startsWith(x, y)` and `endsWith(x, y)` checks if the string $x$ starts with or ends with the substring $y$.

These functions treat $x$ as a single string and checks if it contains certain substrings.
Input Validation, Check Contain

Quiz

1. (Check if the string s contains a letter from t.)
   - \( s = 'abc'; t = 'cde'; \)
   - \( A: \text{contains}(s, t) \)
   - \( B: \text{ismember}(s, t) \)
   - \( C: \max(\text{ismember}(s, t)) \)
   - \( D: \min(\text{ismember}(s, t)) \)

   - Yes
   - No
   - One of \( s \) is member of \( t \)
   - All of \( s \) is member of \( t \)
Input Validation, Check Combination

Quiz

(Check if the string s contains at least two letters and a number.)

0 - false

1 s = 'abc';

A: \( \text{sum( isletter(s) + isstrprop(s, 'digit'))} \geq 3 \)

B:

\( (\text{sum( isletter(s))} \geq 2) \lor (\text{sum(isstrprop(s, 'digit'))} \geq 1) \)

C:

\( (\text{sum( isletter(s))} \geq 2) \land (\text{sum(isstrprop(s, 'digit'))} \geq 1) \)

[ or scalar \& \land ]
Input Validation, Check Permutation

Quiz

1. Check if s is a permutation of t.
2. \( s = \text{'aacc'}; \quad t = \text{'abbc'}; \)
3. \( A: \quad \text{sum}(s) == \text{sum}(t) \)
4. \( B: \quad \text{sum}(	ext{ismember}(s, t)) == \text{length}(t) \)
5. \( C: \quad \text{sum}(	ext{sort}(s)) == \text{sort}(t)) == \text{length}(t) \)

\( aacc == abbc \Rightarrow 1001 \quad \text{sum} \quad z \neq 4 \)
Input Argument Validation

Code

The inputs to a function can be validated so that an input that does not satisfy the conditions will cause an error instead of incorrect outputs.

```
1. function f(x)
2. arguments
3. x (size) class {functions} = default_value
4. end
5. ... % actual function
6. end
```
Input Argument Size

Code

- In `x (size) class { functions}`, the size is specified by a comma-separated list.
- `x (n, m) ...` requires `x` to be an `n x m` matrix.
- `x (n, :) ...` requires `x` to be a matrix with `n` rows or a vector with `n` elements.
- `x (:, m) ...` requires `x` to be a matrix with `m` columns.
Input Argument Class

Code

- In `x (size) class {functions}`, the class is specified by its class name.
- `x char ...` and `x string ...` require `x` to be a string.
- `x single ...` and `x double ...` require `x` to be a number.
- `x logical ...` requires `x` to be a Boolean variable.
In `x (size) class {functions}`, the functions are special functions that raise an error when some conditions are not satisfied.

For example, `x {mustBeGreaterThanOrEqualTo(x, l), mustBeLessThanOrEqualTo(x, u)} ...` requires `x` to be between `l` and `u`, and raises an error when `x < l` or `x > u`.

For example, `x {mustBeMember(x, [u v w])}` requires `x` to be one of `u` or `v` or `w`.
Function of Functions

Math

- A functional (noun.) is a function that can take another function as an input, or returns another function as an output. Functionals are also called higher-order functions.

- The differential and integral operators are functionals: they take a function as input and output another function (or a scalar).

\[
\text{Functional } \left( \frac{d}{dx} \right) (f) \rightarrow \text{ another function }
\]

\[
\int (f, a, b) \rightarrow \text{ number}
\]
Function Handle

Code

- \( y = @f \) creates a variable \( y \) that represents the function \( f \). The variable \( y \) is a function handle.

- Function handles provide a way to pass a function as an input argument to another function.
Anonymous Functions

Code

\[ y = \frac{\sin(x) + 1}{\sin(x) + 1} \]

- \( y = @(x) f(x) \) creates an anonymous function and stores it in the variable named \( y \).

- Anonymous functions provide a way to write a function handle without defining a separate file for the function.

\( y(0) \to \sin(0) + 1 = 1 \)
Function Max at Value Example

Code

Another version of the max function can be defined to find the maximum of two functions $f$ and $g$ at a point $x$.

1. function $mfg = \text{maxFun}(f, g, x)$
2. $mfg = \max(f(x), g(x))$
3. end

For example, $\text{maxFun}(\text{@sin}, \text{@}(x)(\sin(2 \cdot x)), 1)$ finds the maximum between $\sin(1)$ and $\sin(2 \cdot 1)$. 
Function Max Example

Code

Another version of the max function can be defined to find the maximum of two functions $f$ and $g$ and return a function.

1. function $mfg = \text{maxFun}(f, g)$
2. $mfg = @(x)(\max(f(x), g(x)))$;
3. end

For example, if $h = \text{maxFun}(@\sin, @(x)(\sin(2 \cdot x)))$, then $h(1)$ finds the maximum between $\sin(1)$ and $\sin(2 \cdot 1)$.

$h(1) = \max(f(1), g(1))$

$h(1) = \max(f(1), g(1))$
Function Handle, Max

Quiz

1. function \( h = \text{maxFun}(f, x) \)
2. \( h = \max(f(x)); \)
3. end

- \( \text{maxFun(@x}(-x.^2, -2:2) \)
- \( B : -4 \)
- \( C : 0 \)
- \( D : 4 \)
- \( E : \text{Error} \)

Q5

\( -x^2 \)

\( \max \)

\( -x^2 \) at \([-2, -1, 0, 1, 2]\)

\( \max [-4, -1, 0, -1, -4] = 0 \)
Function Handle, Noise

Quiz

Q.6

1. function \( h = \text{noise}(f, g, v) \)
2. \( h = @(x)(f(x) + v * g(x)) \);
3. end

- \( h = \text{noise}(@\sqrt{}; @\(x\)(x^2), 0.5); h(4) \)
- \( B = 18 \)
- \( C = 10 \)
- \( D = 6 \)
- \( E : \text{Error} \)

\[ \sqrt{4 + 0.5 \cdot 16} \]

\[ 2 + 8 = 10 \]

Pizza post list of input validation functions

\( \rightarrow \) p1-p1, p4-p6 code next week.
\( \rightarrow \) debugger example next week.
Function Handle, Random Noise

Quiz

1. function \( h = \text{noise}(f, g) \)
2. \( h = @(x)(f(x) + \text{rand}() \times g(x)); \)
3. end

- \( h = \text{noise}(\text{@sqrt}, @(x)(x.^2)); \ h(4) == h(4) \)
- \( B : 0 \)
- \( C : 1 \)
- \( E : \text{Error} \)
Blank Slide