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

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

- \textit{contains}(x, y)\quad \text{checks if the string } x \text{ contains a substring } y.
- \textit{count}(x, y)\quad \text{counts the number of occurrences of } y \text{ in the string } x.
- \textit{startsWith}(x, y)\quad \text{and } \textit{endsWith}(x, y)\quad \text{checks if the string } x \text{ starts with or ends with the substring } y.
- These functions treat } x \text{ as a single string and checks if it contains certain substrings.
(Check if the string \( s \) contains a letter from \( t \).)

1

\[
s = 'abc'; \quad t = 'cde';
\]

\( A : \) \texttt{contains}(s, \ t) \\
\( B : \) \texttt{ismember}(s, \ t) \\
\( C : \) \texttt{max}(\texttt{ismember}(s, \ t)) \\
\( D : \) \texttt{min}(\texttt{ismember}(s, \ t))
Input Validation, Check Combination

Quiz

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

0

1 $s = 'abc';$

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

B: $(\text{sum( isletter (s))} \geq 2) \mid (\text{sum(isstrprop(s, ' digit ')}) \geq 1)$

C: $(\text{sum( isletter (s))} \geq 2) \& (\text{sum(isstrprop(s, ' digit ')}) \geq 1)$
Input Validation, Check Permutation

**Quiz**

1. (Check if s is a permutation of t.)

- 0

- $s = 'aacc';\ t = 'abbc';$

- A: $\sum(s) == \sum(t)$

- B: $\sum(\text{ismember}(s, t)) == \text{length}(t)$

- C: $\sum(\text{sort}(s) == \text{sort}(t)) == \text{length}(t)$
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.

```plaintext
function f(x)
  arguments
  x (size) class {functions} = default value
end
...  % actual function
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 $n \times m$ 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.
Input Argument Validation Function

Code

- In `x (size) class { functions}`, the functions are special functions that raise an error when some conditions are not satisfied.
- For example, `x { mustBeGreaterThanOrEqual(x, l), mustBeLessThanOrEqual(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`.
A functional (noun.) is a function 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 outputs another function (or a scalar).
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

- $y = @(x) f(x)$ 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.
Another version of the max function can be defined to find the maximum of two functions $f$ and $g$ at a point $x$.

```matlab
function mfg = maxFun(f, g, x)
    mfg = max(f(x), g(x));
end
```

For example, `maxFun(@sin, @(x)(sin(2 * x)), 1)` finds the maximum between $\sin(1)$ and $\sin(2 \cdot 1)$. 
Another version of the max function can be defined to find the maximum of two functions $f$ and $g$ and return a function.

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

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

Quiz

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

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

Quiz

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

$h = noise(@sqrt, @(x)(x.^2), 0.5); h(4)$

$B : 18$
$C : 10$
$D : 6$
$E : Error$
Function Handle, Random Noise

Quiz

1. \texttt{function h = noise(f, g)}
2. \hspace{0.5cm} \texttt{h = @(x)(f(x) + rand() * g(x));}
3. \texttt{end}

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