

Problem ID Conditional Lab: Static Friction Combined input Programmer Solution

## Set Up/ Planning

Type of Program:  Script  Function

## 1. Problem Statement (in your own words):

*Develop code for the user to select one of four cases and then calculate the static friction for that case. The four cases are: 1) metal on metal ( $\mu = 0.2$ ), 2) Wood on Wood ( $\mu = 0.35$ ), 3) Metal on wood ( $\mu = 0.4$ ) and 4) Rubber on concrete ( $\mu = 0.7$ ). The program should handle command line or interactive input based on the user call (command line if input arguments are provided and interactive if inputs are not provided).*

Inputs: (full name, variable to be used, units)

Variable Name	Description	Units or Values	Input Source*
$W$	<i>Weight of object</i>	<i>Pounds (lbs)</i>	<i>Command Line or Interactive</i>
$M$	<i>Number of material combination</i>	<i>1, 2, 3 or 4</i>	<i>Command Line or Interactive</i>

## 2. Output: (full name, variable to be used, units)

Variable Name	Description	Units or Values	Output type*
$F$	<i>Frictional Force</i>	<i>Lb-force</i>	<i>Command Line or Interactive</i>

## 3. Solution Steps (order of these two parts may be varied):

- (1) Perform calculation on test case(s)      (2) Identify the steps/equations to be used in code  
Include flowchart when appropriate

**Test Cases:**

1. Metal on metal,  $\mu = 0.2$ , *weight = 12 lbs*,       $F = (0.2)(12 \text{ lbs}) = 2.4 \text{ lbs-force}$   
 2. Wood on wood,  $\mu = 0.35$ , *weight = 12 lbs*,       $F = (0.35)(12 \text{ lbs}) = 4.2 \text{ lbs-force}$   
 3. Metal on wood,  $\mu = 0.4$ , *weight = 12 lbs*,       $F = (0.4)(12 \text{ lbs}) = 4.8 \text{ lbs-force}$   
 4. Rubber on concrete,  $\mu = 0.7$ , *weight = 12 lbs*,       $F = (0.7)(12 \text{ lbs}) = 8.4 \text{ lbs-force}$

Also test each I/O method:

- Weight & material number at the command line → no interaction, output at command line
- No input variables → interactive input, no requirements were specified in the problem statement. For this implementation output is in a message box for this case.

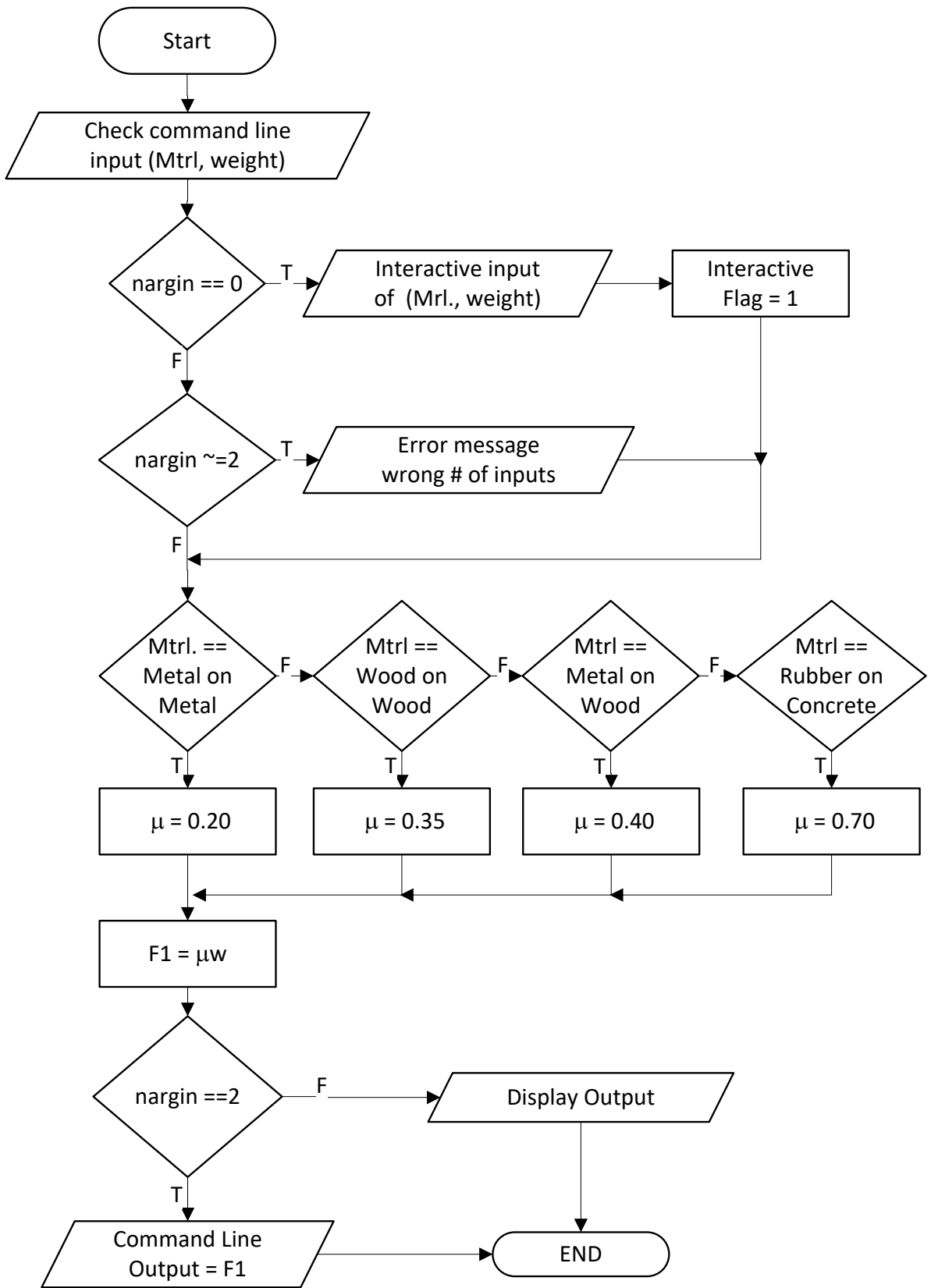
Give credit at most reasonable attempts at the above.

OK if students have put this information in the validation section

It is an issue is there are no comparison values

**Program logic** shown by flowchart on next page

Considerable variation in flowcharts is likely – as long as logic is reasonable and can be followed they should get credit.



Program Code (with comments):

```

function F = StaticFriction(W,M)
% S.Moor March 2005
% Calculates the static friction for one of four specific cases given the case and the weight.
%
% Input can be either at the command line or by interactive questions.
% Command Line: I/O is interactive if two inputs are provided
% Interactive: Interactive questions if no input arguments in the call.
% If an output variable is provided in the call the output will % be at the command line.
% Otherwise it will be displayed in the command window.
%
% Call Structure: F = StaticFriction(W,M)
% The four Cases (M): 1. Metal on metal
%                    2. Wood on wood
%                    3. Metal on wood
%                    4. Rubber on concrete
%
% Inputs W = weight of the object (lbm)
%        M = a number representing the material selections as shown above
% Output: F = force required to over come static friction (lbf)

% Intermediate variable: mu = the coefficient of static friction
%                       F1 = intermediate holder for output force

% Input section provides interactive input if no command line arguments are given.
if nargin ==0
    M = menu('Select Material Combination', 'Metal on Metal',...
            'Wood on Wood', 'Metal on Wood', 'Rubber on Concrete');
    W = input('Specify the weight of the moving object (lbs.): ');
elseif nargin ~= 2
    disp('Error: Program requires zero or two input arguments')
end

% switch structure selects the correct coefficient of static friction based on material input.
switch M
    case 1
        mu = 0.2;
    case 2
        mu =0.35;
    case 3
        mu=0.40;
    case 4
        mu =0.7;
    otherwise
        disp('ERROR: invalid materials choice')
        mu=NaN;
end

% Calculates force required to get object moving, temp. stored in F1
F1=mu*W;

% Sets output to be command line or interactive depending on user call
if nargin ==2 % Set to command line output if input variables were provided.
    F = F1;
else % Otherwise display output in command window
    disp(['The starting static friction force = ', num2str(F1), ' lbs-force'])
end

```

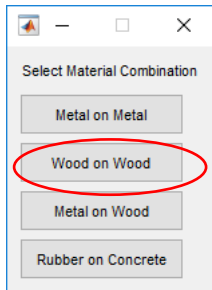
Validation: (prove that the program works by showing the execution and comparing to known results from another source, e.g. hand calculations from step 4).

Using four cases from step 4 and using each of both possible call formats the following was run

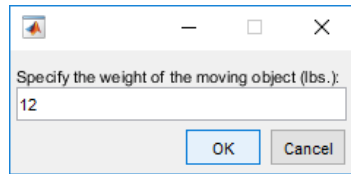
>> StaticFrictionNew(12, 1)                      *Runs as a pure command line function*  
ans =  
    2.4000                                      *Answer matches hand calculation #1*

>> StaticFrictionNew                              *Run with interactive input*

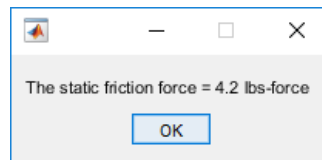
1. *Material entered via Popup Menu (wood on wood selected)*



2. *Entered 12 in dialog box*

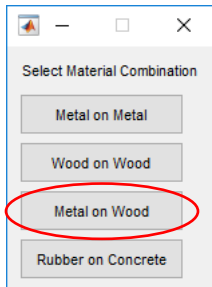


3. *Resulting Message box (Answer matches hand calculation #2)*

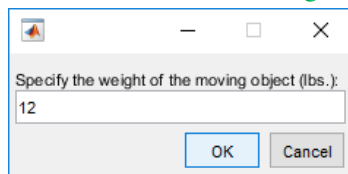


>> StaticFrictionNew                              *Run with interactive input*

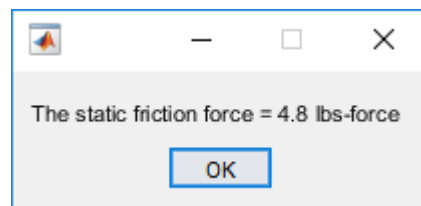
1. *Material entered via Popup Menu (Metal on wood selected)*



2. *Entered 12 in dialog box*



3. *Resulting Message box (Answer matches hand calculation #3)*



>> F = StaticFrictionNew(12, 4)                      *Runs as a pure command line function*

F =  
    8.4000                                      *Answer matches hand calculation #4*

Alternative:

The previous solution example is for the second option presented in the handout.  
For the first option students can use a main program that calls the two programs coded previously.

The main might look like this:

```
function F = StaticFriction(W,M)
% S.Moor      March 2020

% ... lots 'o' introductory comments

% Select subprogram based on input

if nargin ==0          % if no inputs call interactive program
    F = StaticInteractive
elseif nargin = 2     % if two inputs call command line program
    F = StaticCommandLine(W,M)
else                  % if other display error message
    disp('Error: Program requires zero or two input arguments')
end
```

Sub programs would include the appropriate parts of the code from the combined example.

For this case three flow charts are required – one for the main and one for each subprogram.