

CS100M Section (Lab) Exercise 2

When you have completed the lab, show this sheet and any associated programs to your lab instructor, who will record that you have completed the lab. If you do not finish this exercise during the lab, show the instructor what you have done at the end of the lab and be sure to complete it in the next few days.

If you have any questions, **ask** your lab instructor or a consultant immediately! They are in the lab to help you learn the material.

CMS: Course Management System

You will use CMS to submit homework, find your scores, view grading comments, and access the on-line book chapters. Try to log in into CMS now. *If you are not able to log in you must tell your section instructor now to add your information into the system.*

1 Minimum of a quadratic

<http://www.cs.cornell.edu/courses/cs100m/> → FVL Chapters & Software

Download and review **Eg1.2** in *FVL*.

[M1.2.4 from *FVL*] Modify **Eg1.2** to first check that **L** is less than or equal to **R**. Switch the values of **L** and **R** if necessary before computing the minimum of the quadratic.

Refer to fragment at the top of page 10 (above **Eg1.2**). Modify the **if**-condition so that it uses the logical or operator (**||**) instead of the logical and (**&&**) operator. You may need to make other changes to the fragment.

2 Triangle

The three interior angles of any triangle add up to 180° . Complete the program fragment below to print scalene, isocoles, or equilateral given three angles.

```
% Assume a, b, and c are positive integers that sum to 180

if (_____ )
    disp('Scalene triangle')
else
    if (_____ )
        disp('Equilateral triangle')
    else
        disp('Isocoles triangle')
    end
end
end
```

3 Which quadrant?

Write two different programs to determine in which quadrant a user-input value of A degrees belongs. Assume that the user may enter any non-negative number. For example, 725° is the same, and should be treated, as 5° . (Hint: the function **rem** that you saw last week might be useful.) To avoid ambiguity, we use the following convention:

$$\text{Quadrant is } \begin{cases} 1 & \text{if } 0 \leq A < 90 \\ 2 & \text{if } 90 \leq A < 180 \\ 3 & \text{if } 180 \leq A < 270 \\ 4 & \text{if } 270 \leq A < 360 \end{cases}$$

Print the result. In the first script use four *separate* **if** statements (4 separate **if-end** constructs) and call the program **angle1.m**. In the second script, use a *single* **if-elseif-else-end** construction for the evaluation and call it **angle2.m**. Pay close attention to the differences between the two programs—are the boolean expressions as concise as they can be?

4 Submitting text files in CMS

The files that you create using MATLAB's editor window are *plain text files*, i.e., there is no formatting information associated with any character in the file. For example, there is no *italic*, **bold**, or other character format. You can create plain text files using applications such as Notepad or Wordpad. MATLAB's Editor also produces plain text files. Be sure to submit plain text files whenever we ask you to submit MATLAB files with the filename extension `.m`.

To make sure that you understand the difference between plain text files and files with formatting, create two files, one using *Notepad* and the other using *Microsoft Word*, containing the same text. You can use any text, but here's a suggestion:

```
"There's nothing sillier in the world, I say, than being a devil in  
despair." (Mephistopheles)
```

```
Faust -- Johann Wolfgang Von Goethe.
```

Save both files in their default format. Use *Windows Explorer* to find out their sizes (in bytes). Which one is smaller? Now close both files and then try to open the *Microsoft Word* file using *Notepad*, a plain text editor. What do you see?

5 Challenge question*

*Topic for next lecture—not required in this section (but think about it if you have time!)

Nesting in programming refers to the inclusion, or placement, of one construct inside another. For example, we can *nest* `if`-statements:

```
if boolean expression 1  
    code to execute if expression 1 evaluates to true  
else  
    if boolean expression 2  
        code to execute if expr1 evaluates to false but expr2 evaluates to true  
    else  
        code to execute if expr1 evaluates to false and expr2 evaluates to false  
    end  
end
```

In the example above, we nested an `if`-statement inside the `else` branch of the first, or outer, `if`-statement. Use *nested if*-statements without `elseif` to solve the quadrant question above. Call the script `angle3.m`.

Please delete your files from the computer before leaving the lab!