Programming Exam 1 120 pts

Instructions

- Write a MATLAB script and function file that answers the questions below. Your submission for this part of the exam will be 1 .m script file
- Suppress all command output except the command that produces the final output requested for each question
- You may only access MATLAB, MATLAB's help interface, and this assignment while taking the exam; no notes, textbook, powerpoint, or previous example scripts etc. You may not confer with any other students. I reserve the right to observe your screens and end the exam for anyone violating this.
- The exam must be taken in the class meeting location unless prior arrangements and accommodations were made and agreed upon. I reserve the right to take attendance and not accept submissions from students without an attendance record or alternative arrangements agreed to in advance.
- The assignment must be submitted while in class. I reserve the right to reject exams submitted after the student has left the class.
- You have one hour fifteen minutes to complete this exam, unless accommodations have been granted in advance.
- Upload your completed script file as your submission for the assignment. The script file **MUST** be named in the format **YourlastnameYourFirstNamePE1.m**

Exam

1. (5 pts) Make the first line or couple lines of your script a comment with your name, the date, and a statement that this is your submission for programming exam 1. Credit for this also includes correctly naming your script file according to instructions above

2. (15 pts) Create a 3x5 matrix called M1 with the following methods for each row:
   a. Row 1: Should contain the values 2 4 6 8 10. Use the M:Q:N method (colon notation) to construct this row.
   b. Row 2: Should contain the values 3 6 9 12 15. Use the linspace function to construct this row.
   c. Row 1: Define the variables $d = \tan\frac{2}{3}\pi$, $f = \cos 60°$, $g = -3e^{1.5}$, $h = \ln(f - d)$, and $j = g * \frac{d}{3}$. Then use these variables to construct the third row [d f g h j]

   Final matrix M1 should show in the command window when script is run and should look like:

```
M1 =

    2.0000    4.0000    6.0000    8.0000   10.0000
    3.0000    6.0000    9.0000   12.0000   15.0000
   -1.7321    0.5000  -13.4451    0.8029    7.7625
```

3. (10 pts) Query the user to provide a row and column number, and return (assign to a variable and display in the command window) the element of M1 that matches that address.

4. (15 pts) The following table presents forces (F in Newtons) applied to a set of springs, and the spring rate (k, in N/m) of those springs.

F (N)	70	90	110	120	150
k (N/m)	87.5	100	98.5	105.5	110

According to Hook's law, spring force can be related to spring displacement (x, in m) by its spring rate k (N/m) such that

$$F = kx$$

Further, the energy stored in the spring is given by U (J) such that

$$U = \left(\frac{1}{2}kx^2\right)$$

Create vectors of F and k from the given data (may hard code by typing in values manually), use these vectors to find displacement x and energy stored U (using array operations), then create matrix **Sp** with four columns: the spring force, the spring rate, the displacement, and the stored energy. Allow Sp to be displayed in the command window.

5. (30 pts) Make the following two plots in subplots within the same figure window (5 pts for correct subplots).
    a. (15 pts) Plot F vs x and U vs x from the data in problem 4. Make the x data the horizontal axis data, F and U on the vertical axis. Have both lines on the same set of axes. Default line styling is fine, but feel free to customize if you like. Include the following:
        i. Title: **Force and Energy in Compressed Springs**
        ii. X axis label: **Spring Displacement (m)**
        iii. Y axis label **Force (N) and Energy (J)**
        iv. Legend: **Spring Force** and **Energy Stored** default location is fine
    b. (10 pts) Plot the function $2x^3 + 0.5x^2 - x + 3$ using fplot over the range $-10 \leq x \leq 10$

6. (20 pts) Query the user to enter a number. Display back to the user a statement repeating the value they entered, and what the sign of the value is. For example, if the user enters -2, display to the user something like "-2 is a negative number." Consider 0 to be positive.

7. (25 pts) Create a 4x4 matrix called ***Z*** where the value of each element is given by $z(r, c) = \sqrt{R * C}$ where C is the column number (starting at 1, ending at 4) and R is the row number (starting at 1, ending in 4). Start with a 4x4 matrix full of zeroes and use a loop structure to update the value of each element one by one. Display ***Z*** only once, after it is completed.