ENGR 15100: Software Tools for Engineers
Spring 2015

Computer Assignment #11

Due: Tuesday, April 28, 2015, 9:00am CST

Departments of Engineering
School of Engineering, Mathematics, & Sciences
Purdue University Calumet

PURDUE
UNIVERSITY
CALUMET

1. OBJECTIVE

Continue working with user-defined functions, iterative statements, and conditional statements.

2. PROCEDURE

Task I: User-Defined Functions – Computing Factorials [30 points]

(a) **[15 points]** The factorial of a natural number (i.e. a nonnegative integer) `n` is defined by $n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 2 \cdot 1$, where $0! = 1$. Write a user defined function named **`myFactorial()`** that computes the factorial of a natural number `n`. Function **`myFactorial()`** accepts one input parameter named `n` and returns one output parameter named `factOut`. Parameter `n` represents the natural number while `factOut` represents the result of performing the factorial operation on parameter `n`. *You may assume* `n` *is a natural number comprising less than or equal to* 20 *digits.* Utilize a **`for-end`** statement to implement **`myFactorial()`**. *You may not use the built-in MATLAB function* **`factorial()`** *in the body of* **`myFactorial()`**.

(b) **[15 points]** In a script file named `LASTNAME_LAB11_TASK1.m`, write a program that exercises the user-defined function **`myFactorial()`**. Before starting, clear all MATLAB Workspace variables and clear the MATLAB Command Window contents. Your program should perform the following steps:

i. Continuously prompt the user until the user enters a natural number. *You may assume the user will always enter a numerical scalar*.

ii. Call user-defined function **`myFactorial()`** to compute the factorial of the natural number entered by the user.

iii. Use built-in function **`fprintf()`** to display the result of computing the factorial of the natural number entered by the user. Display the result with 20 significant digits.

The above steps should be repeated as long as the user enters a nonnegative numerical scalar. Below is a sample of the Command Window after executing the script for *some* input combinations.

```
Enter a natural number (negative to quit): 0
0! = 00000000000000000001
Enter a natural number (negative to quit): 1
1! = 00000000000000000001
Enter a natural number (negative to quit): 3
3! = 00000000000000000006
Enter a natural number (negative to quit): 20
20! = 02432902008176640000
Enter a natural number (negative to quit): 2.3
The number is not an integer. Please try again.
Enter a natural number (negative to quit): -1
```

Test your program for at least the following program usage scenarios:

- User enters a negative number at the start of the program.
- User enters a negative number immediately after entering a positive real number by mistake.
- User enters either a negative real number or a negative integer to end the program after computing at least one valid factorial result.

Task II: User-Defined Functions – Cylinder Geometry [70 points]

(a) **[15 points]** In a function file named `getCylinderInfo.m`, write a user-defined function named **`getCylinderInfo()`** that prompts the user for the height and radius of a right circular cylinder. The user should be continuously prompted until the user enters a non-negative radius and a non-negative height, respectively. User-defined function **`getCylinderInfo()`** does not accept any input parameters but returns two output parameters named **`heightOut`** and **`radiusOut`**, respectively. Output parameter **`heightOut`** represents the height of the circular cylinder while output parameter **`radiusOut`** represents the radius of the top and bottom parts of the circular cylinder.

(b) **[15 points]** In a function file named `calcAreaAndVolume.m`, write a user-defined function named **`calcAreaAndVolume()`** that calculates the area and volume of a right circular cylinder having a particular height and radius. User defined function **`calcAreaAndVolume()`** accepts two input parameters named **`heightIn`** and **`radiusIn`**, respectively. Input parameter **`heightIn`** represents the height of a circular cylinder while input parameter **`radiusIn`** represents the radius of the top and bottom parts of the same circular cylinder. User defined function **`calcAreaAndVolume()`** also has two output parameters named **`surfAreaOut`** and **`volOut`**, respectively. Output parameter **`surfAreaOut`** represents the surface area of the right circular cylinder having a height and radius of **`heightIn`** and **`radiusIn`**, respectively. Output parameter **`volOut`** represents the volume of a right circular cylinder with having a height and a radius of **`heightIn`** and **`radiusIn`**, respectively.

(c) **[10 points]** In a function file named `dispAreaAndVolume.m`, write a user-defined function named **`dispAreaAndVolume()`** that displays the area and volume of a right circular cylinder using multiple instances of the **`fprintf()`** function. User defined function **`dispAreaAndVolume()`** accepts two input parameters named **`surfAreaIn`** and **`volIn`**, respectively. Input parameter **`surfAreaIn`** represents the surface area of a right circular cylinder while input parameter **`volIn`** represents the volume of the same right circular cylinder. User defined function **`dispAreaAndVolume()`** does not generate/return have any output parameters.

(d) **[30 points]** In a script file named `LASTNAME_LAB11_TASK2.m`, write a program that utilizes the user-defined functions created in parts (a) through (c). Before starting, clear all MATLAB Workspace variables and clear the MATLAB Command Window contents. Within one **`while-end`** statement, the program operates according to the following specifications:

1) **[1 point]** Clear the contents of the Command Window.
2) **[1 points]** Increment the current cylinder count (initialize the current cylinder count externally).
3) **[2 points]** Display in the Command Window the current cylinder count using **`fprintf()`**. A sample of the Command Window after performing steps 1) through 3) is given below.

```
Cylinder #1
---------------------------------
```

4) **[3 points]** Get the current right circular cylinder's height and radius from the user. Utilize the user-defined **`getCylinderInfo()`** function.
5) **[5 points]** Calculate the current right circular cylinder's surface area and volume. Utilize the user-defined **`calcAreaAndVolume()`** function.

6) [3 points] Display the current right circular cylinder's surface area and volume. Utilize the user-defined **dispAreaAndVolume()** function.
7) [1 point] Prompt the user if he/she wants to continue using the prompt string `'Want to calculate info for another cylinder (y = YES, n = NO)?: '`.

[13 points] Repeat steps 1) through 7) as long as the user enters a single `'y'` or a single `'Y'` when prompted as part of step 7). When the user enters a single `'n'` or a single `'N'` when prompted as part of step 7), the **while-end** statement should terminate. Otherwise, the user should be continuously prompted.

[1 point] Then, a message is displayed in the Command Window giving the number of cylinders for which information was calculated.

Below is a sample of the Command Window after performing steps 1) through 7) for three iterations (passes). ***Test your program for several cylinder heights and radii, including positive and negative values thereof.***

```
Cylinder #3
--------------------------------
Enter the height of a cylinder: 4
Enter the radius of a cylinder: 5
The surface area of the cylinder is 282.743
The volume of the cylinder is 314.159
Calculate info for another cylinder (y = YES, n = NO)?: N
You calculated information for 3 cylinders.
```

Task III

Upload the following MATLAB files to Blackboard Learn:

(a) `myFactorial.m`
(b) `LASTNAME_LAB11_TASK1.m`
(c) `getCylinderInfo.m`
(d) `calcAreaAndVolume.m`
(e) `dispAreaAndVolume.m`
(f) `LASTNAME_LAB11_TASK2.m`