

Department of Chemical Engineering  
ChE-101: Approaches to Chemical Engineering Problem Solving  
**MATLAB Tutorial I**

**MATLAB Basics**

(last updated 3/29/05 by GGB)

Objectives:

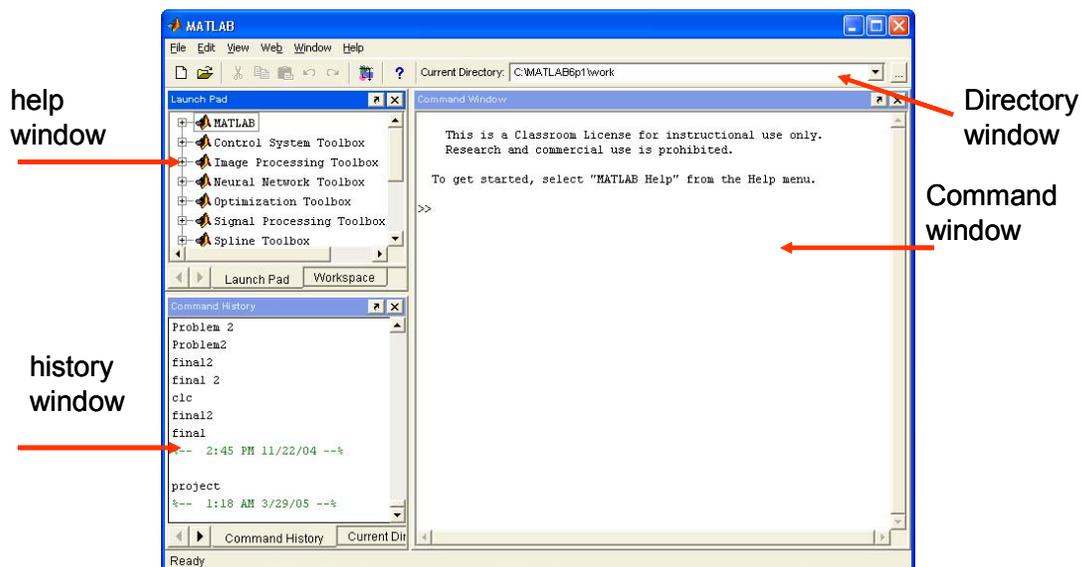
These tutorials are designed to show the introductory elements for any of the topics discussed. In almost all cases there are other ways to accomplish the same objective, or higher level features that can be added to the commands below.

Any text below appearing after the double prompt (`>>`) can be entered in the Command Window directly or in an m-file.

The following topics are covered in this tutorial;

**Starting MATLAB****Exiting MATLAB****Using the MATLAB Command Window****Matlab's Interaction with the Computer Memory****Online Help****Saving and Retrieving Data****Number Display Formats****Variable Names****Commenting Code****Simple Math Operations****Scientific Features / Common Math Functions****Solved Problems (guided tour)****Proposed Problems****Starting MATLAB**

Double click on the MATLAB icon. The MATLAB Command Window will appear and the MATLAB prompt `>>` is displayed. This is what you will see when you do this. Notice the different available windows: command window, help window, command history window, and directory window.



## Exiting MATLAB

You can exit MATLAB in three ways:

- ! by clicking the X in the upper right hand corner of the MATLAB window
- ! by clicking on FILE / EXIT MATLAB
- ! by typing quit at the MATLAB prompt

---

## Using The MATLAB Command Window

This is the window with the MATLAB double prompt, >>. All MATLAB commands are executed from this window. When an m-file is executed from the Command Window, it is the same as if the individual commands had been typed at the double prompt, >>.

### Suppressing Display

Putting a semi-colon at the end of any line of MATLAB code will cause the calculation to be performed, but the answer will not be displayed. The answer is still stored in memory. When we write m-files we use this after virtually all lines, then display the output at the end of the program.

### Retrieving Previous Commands

While in the MATLAB Command Window, you can move back and forth through the list of commands you have used previously, by using the up and down arrow keys. The left and right arrow keys move within a command. This makes for an easy way to repeat a command or edit a previous command. Hitting return executes the command again. To view previous portions of the Command Window use the sliding button or arrows on the right side of the Command Window screen, not the arrow keys. You cannot move up the Command Window and edit the previous display.

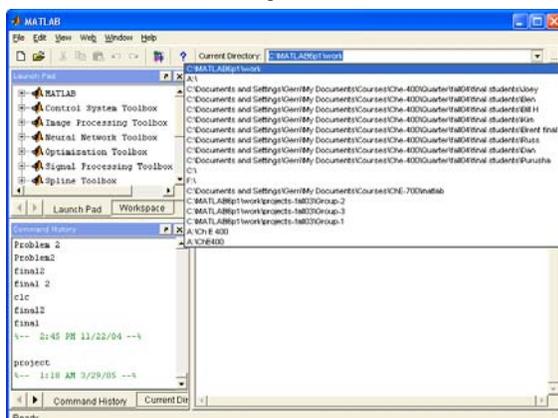
### Clearing the Screen

The MATLAB Command Window screen can be cleared by clicking on EDIT / CLEAR SESSION, or by typing 'clc' at the Command Window prompt. This does not clear the variables in memory.

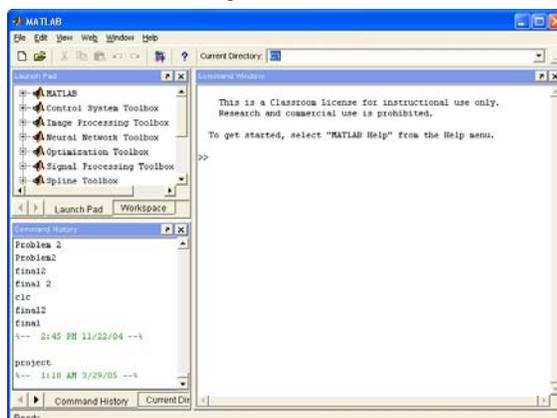
### Changing the Current Active Directory

To change the current active directory use the directory window as shown (example shows how to change to the "c" directory:

#### Step 1

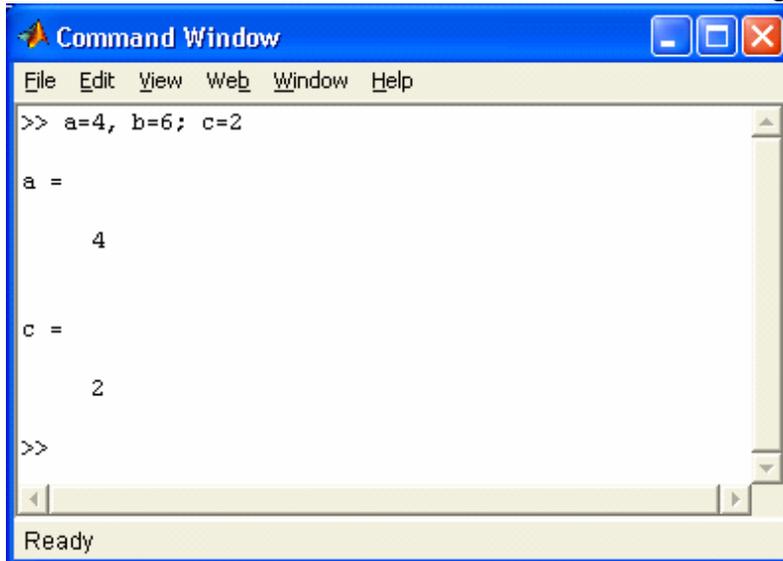


#### Step 2



### Multiple Commands

Multiple commands can be placed on one line if they are separated by commas or semicolons. This is not recommended because it is harder to read and debug.



```
Command Window
File Edit View Web Window Help
>> a=4, b=6; c=2
a =
    4
c =
    2
>>
Ready
```

Note that 'b' is not displayed because of the semi-colon.

### Interrupting a Program

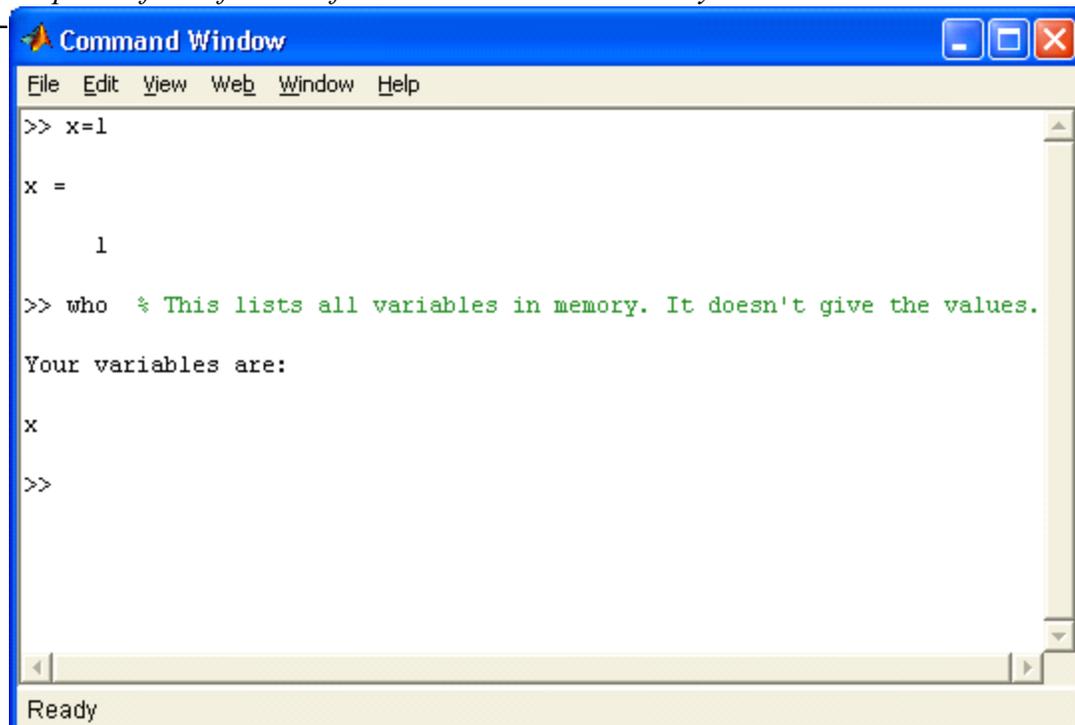
A MATLAB program that is caught in a loop can be interrupted by pressing Ctrl C

---

### MATLAB's Interaction With the Computer Memory

MATLAB interacts directly with the computer memory. During any MATLAB session any variable you use is stored in memory, and can be recalled and used until you clear the memory.

*Note: Function files work differently. See the tutorial "Matlab Function m-files" for a description of how function files interact with the memory.*

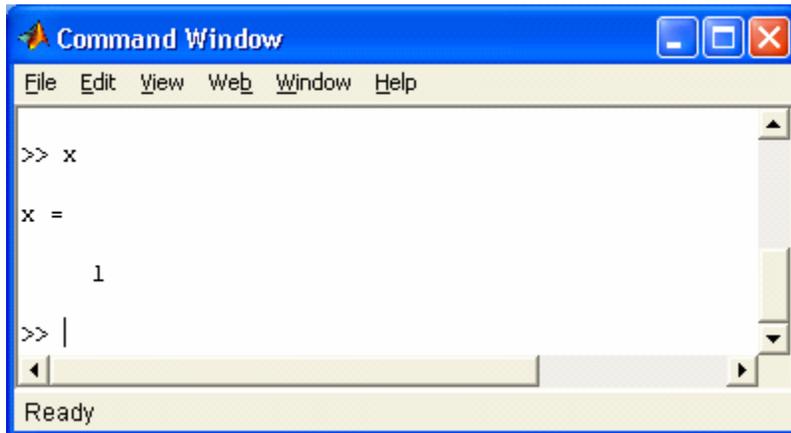


```
Command Window
File Edit View Web Window Help
>> x=1
x =
    1
>> who % This lists all variables in memory. It doesn't give the values.
Your variables are:
x
>>
Ready
```

---

*The value of a variable is displayed by simply typing the variable name*

---



---

*The 'clear' command deletes all memory.*

---

```
>> clear      % This clears the variables in memory
>> who       % Nothing will be displayed.  The memory is cleared.
```

---

**Be careful, there is no getting them back without re-executing the commands that created them.**

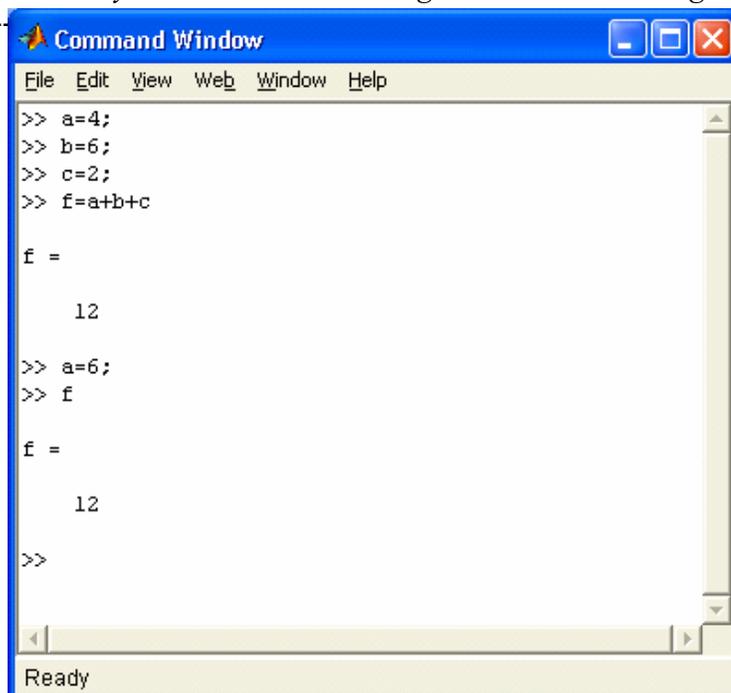
*Variables can be deleted, either individually or altogether.*

---

```
>> clear a;
>> clear      % This clears all variables AND THEY CAN'T BE RETRIEVED
```

*Every time a variable name is assigned either directly by you or through a calculation it is stored in memory. The value is not changed unless it is reassigned through a calculation.*

---



---

*Note that 'f' is not recalculated just because the value of 'a' is changed. To recalculate f with the new value of 'a', use the up arrow key until  $f = a + b + c$  then press enter.*

---

---

### Online Help

Help can be accessed in three different ways:

- ! By clicking on HELP from the menu bar.
  - ! By typing HELP at the MATLAB prompt. Typing HELP alone will result in a list of topics which can then be accessed for sublists. If you know the exact topic you want help on, type HELP TOPIC directly.
  - ! Type LOOKFOR KEYWORD causes MATLAB to search the help files for the keyword, even if it is not a MATLAB command itself.
- 

### Saving and Retrieving Data

It is not necessary to save variables, only the files used to create them.

Store all your m-files on diskettes (3.5" or zip). This is the A drive on the computer. If you don't tell it to use the A drive it will default to the hard drive in the directory C:\TEMP. This is not wise. We cannot promise security of your files. Also the temp directory or even the entire C drive will occasionally be wiped clean.

To save an m- file, click on FILE / SAVE AS in the Matlab Editor, not the Command Window. Change to the desired directory and give it a filename. The .m extension will automatically be added.

To open a saved file, click FILE / OPEN, locate the file then open. Note that opening or saving a file in the Editor does not change the current Command Window directory.

If you want to see the list of all files stored on the current active directory type DIR at the MATLAB prompt.

---

### Number Display Formats

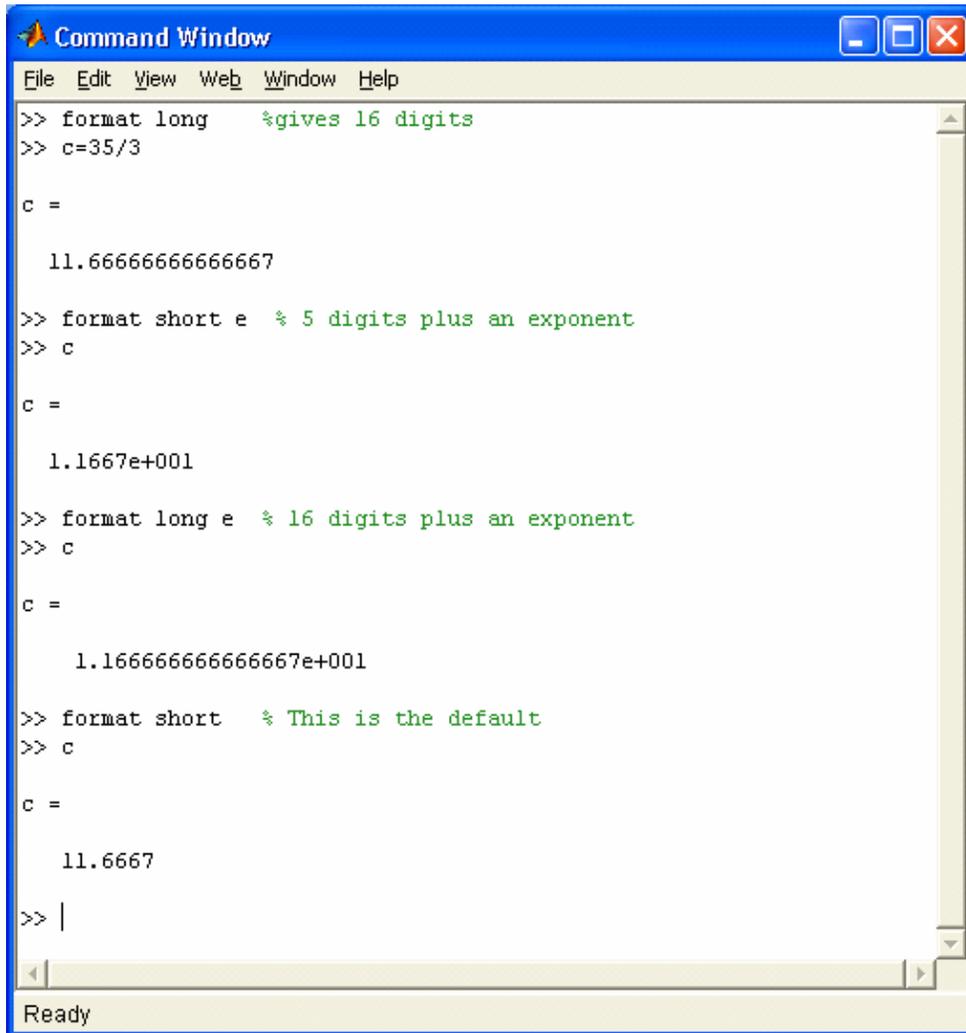
MATLAB allows you to use a number of different numeric displays on the screen. Using any of these does not change the precision of the number as it is stored in memory.

To change the line spacing on the screen, click on FILE / PREFERENCES / NUMERIC FORMAT, then locate the desired choices on the pull down menus. This will change the line spacing and the number of digits displayed.

An alternate method is to use the 'format' command as a line of code.

```
>> format short e
>> format loose
```

For numeric formats the default for MATLAB is to display 4 digits to the right of the decimal point. The most common alternate choices are:



```
Command Window
File Edit View Web Window Help
>> format long %gives 16 digits
>> c=35/3

c =

    11.666666666666667

>> format short e % 5 digits plus an exponent
>> c

c =

    1.1667e+001

>> format long e % 16 digits plus an exponent
>> c

c =

    1.1666666666666667e+001

>> format short % This is the default
>> c

c =

    11.6667

>> |
Ready
```

---

## Variable Names

There are a number of rules as to the variable names you can use.

- ! Variable names must be continuous. It is a good idea to use variable names that make sense. Underscores can be used to make names continuous. ex. (average\_cost)
- ! Variables are case sensitive. FLOW, flow, Flow, FIOw are all different variables
- ! Variables must start with a letter, followed by letters, digits, or underscores.  
X12, rate\_const, Flow\_rate are all acceptable variable names
- ! Punctuation characters are not allowed.

There are a number of special built-in variable names. The most common are:

- ! ans = the default variable
- ! pi = 3.141592....
- ! eps = the smallest number that when added to one results in a number greater than one

! Inf = infinity

```
>> 1/0
ans =
    Inf
```

! NaN = not a number

```
>> 0/0
ans =
    NaN
```

Special variables can be redefined by you. It's better not to do this.

### Commenting Code

Any text on a line after a % sign is taken as a comment statement. This is very useful add a note to the user or yourself as to what a value is or what its units are. The comment can appear as a separate line or be on the same line as an executable command, It is the best way to specify the units on a number.

```
>> x = 4           % x = the distance to Grandma's house in miles
x =
    4
```

### Simple Math Operations

*As with most programming languages the numbers or variables with assigned values on the right side of the equal sign are manipulated and the result is assigned to the variable name on the left.*

```
>> b = 5 + 8
b =
    13
>> 5 + 8           % If no output variable is defined the default is "ans"
ans =
    13
```

*The known information and operation must be on the right side of the equal sign. Matlab cannot make sense of the following.*

```
>> 5 + 8 = b
??? 5 + 8 =
    Missing operator, comma or semicolon
```

*A variable can be reassigned.*

```
-----  
>> b = b + 5  
b =  
    18  
-----
```

### Order of Precedence

The spaces in mathematical expressions make no difference

Calculations are performed left to right with the following order of preference, starting with the inner parenthesis and working outward.

powers ( symbol ^ )

multiplication ( symbol \* ) and division ( symbol / or \ )

addition ( symbol + ) and subtraction (symbol - )

```
-----  
>> c = 20/5  
c =  
    4  
>> c = 20\5  
c =  
    0.2500  
>> c = 3*(4+2)  
c =  
    18  
>> c = 3 * 4 + 2  
c =  
    14  
>> c = 3 ^ (2+1)  
c =  
    27  
>> 2^3-4-8/4*2          <<-----  
ans =  
    0  
>> 2^3-4-8/(4*2)  
ans =  
    3
```

You may want to add parentheses or spaces for readability, even when they are not needed. For example the command indicated by the arrow above is more readable when written as;

```
>> (2^3) - 4 - ((8/4) * 2)  
ans =  
    0
```

---

### **Scientific Features, Common Mathematical Functions**

MATLAB contains a large number of mathematical features. These include logarithms, trig functions, and a number of methods for rounding.

**Printing the command window**

Go to File and select Print from the menu.

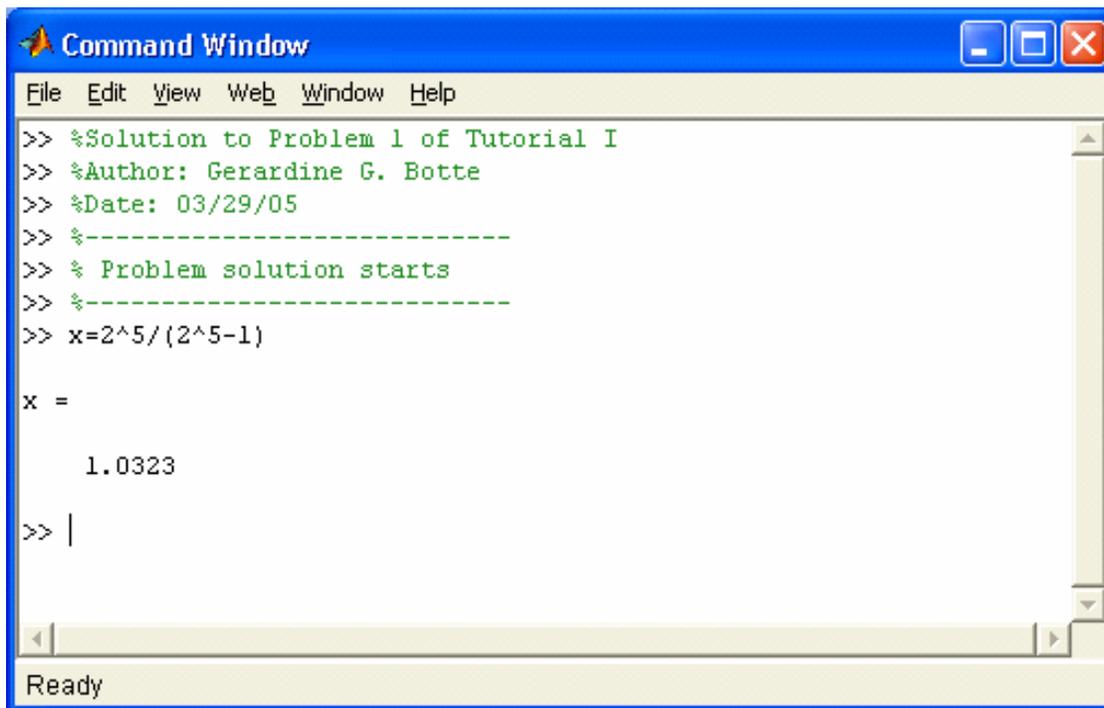
**SOLVED PROBLEMS**

Perform the following arithmetic operations. Print the results of your calculations.

1.  $\frac{2^5}{2^5 - 1}$

Solution: The first step in the command window is to write down the following useful information:

- Problem #
- Your name
- Date
- Any comments to help the reader understand your solution (e.g., define your variables if necessary)



```
Command Window
File Edit View Web Window Help
>> %Solution to Problem 1 of Tutorial I
>> %Author: Gerardine G. Botte
>> %Date: 03/29/05
>> %-----
>> % Problem solution starts
>> %-----
>> x=2^5/(2^5-1)

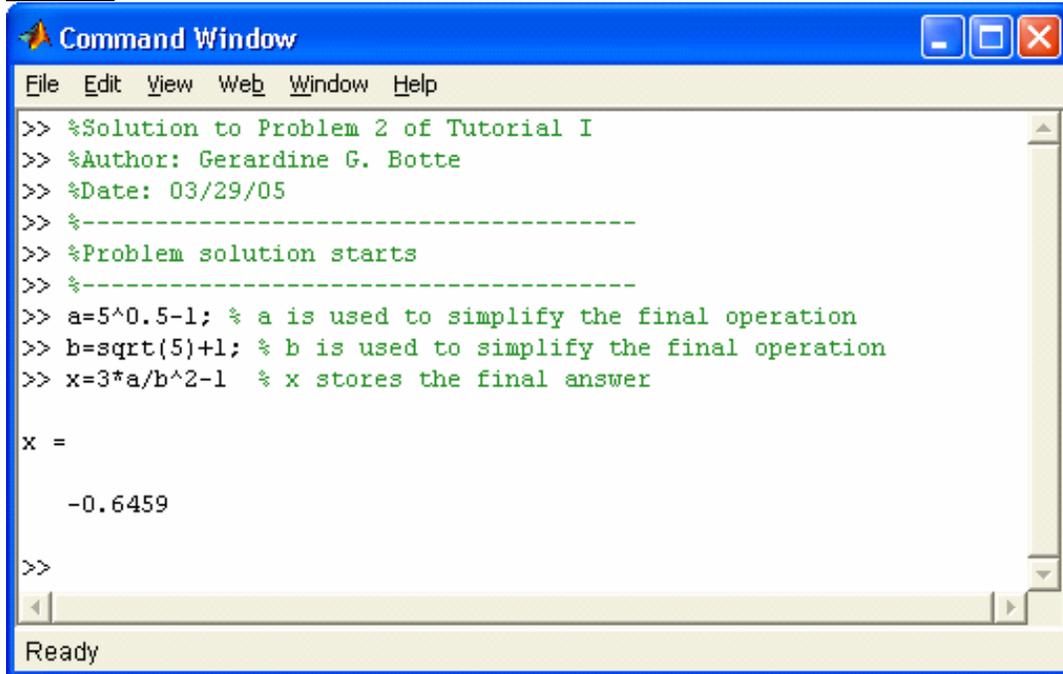
x =

    1.0323

>> |
Ready
```

$$2. 3 \frac{\sqrt{5}-1}{(\sqrt{5}+1)^2} - 1$$

Solution



```

Command Window
File Edit View Web Window Help
>> %Solution to Problem 2 of Tutorial I
>> %Author: Gerardine G. Botte
>> %Date: 03/29/05
>> %-----
>> %Problem solution starts
>> %-----
>> a=5^0.5-1; % a is used to simplify the final operation
>> b=sqrt(5)+1; % b is used to simplify the final operation
>> x=3*a/b^2-1 % x stores the final answer

x =

    -0.6459

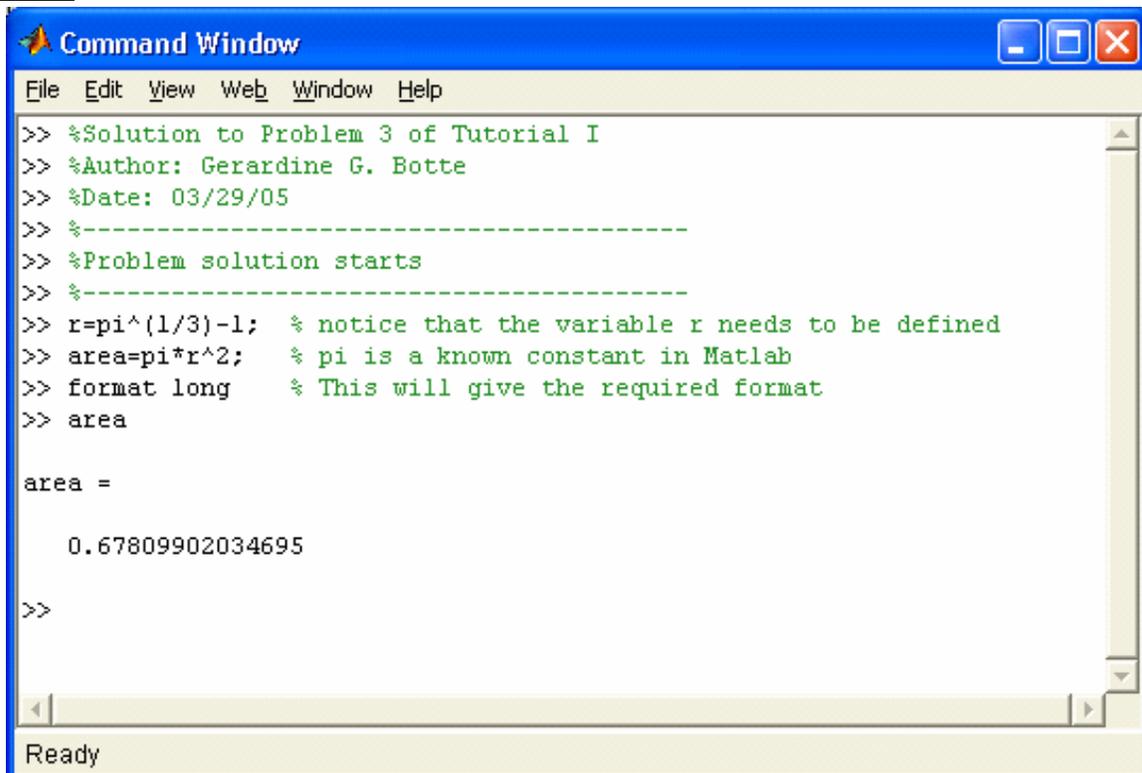
>>
Ready

```

*Notice that the square root was calculated in two different ways. In this problem it was useful to define variables to simplify the final calculation. This is recommended to avoid making mistakes by typing a large expression.*

3. Area =  $\pi r^2$  where  $r = \pi^{1/3} - 1$ . Print your results using 16 digits.

Solution



```

Command Window
File Edit View Web Window Help
>> %Solution to Problem 3 of Tutorial I
>> %Author: Gerardine G. Botte
>> %Date: 03/29/05
>> %-----
>> %Problem solution starts
>> %-----
>> r=pi^(1/3)-1; % notice that the variable r needs to be defined
>> area=pi*r^2; % pi is a known constant in Matlab
>> format long % This will give the required format
>> area

area =

    0.67809902034695

>>
Ready

```

**PROPOSED PROBLEMS**

Perform the following calculations in Matlab.

1.  $\sin^2\left(\frac{\pi}{6}\right) + \cos^2\left(\frac{\pi}{6}\right)$       answer: 1
2.  $y = \cosh^2 x - \sinh^2 x$  with  $x = 32\pi$       answer:  $y = 0$