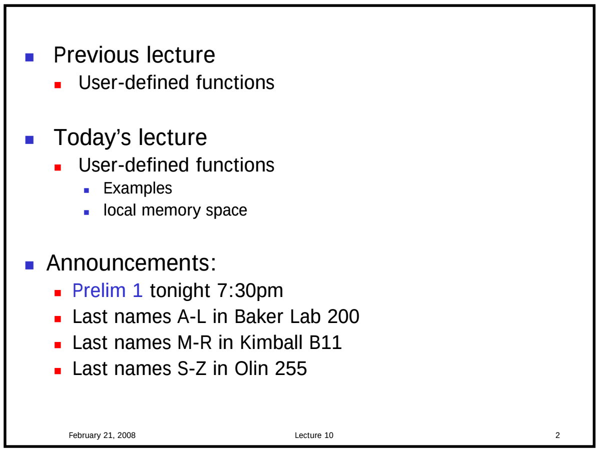
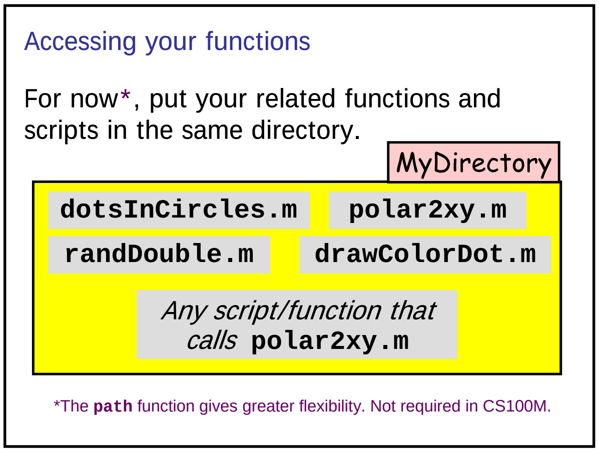


- Previous lecture
  - User-defined functions
- Today's lecture
  - User-defined functions
    - Examples
    - local memory space
- Announcements:
  - Prelim 1 tonight 7:30pm
  - Last names A-L in Baker Lab 200
  - Last names M-R in Kimball B11
  - Last names S-Z in Olin 255

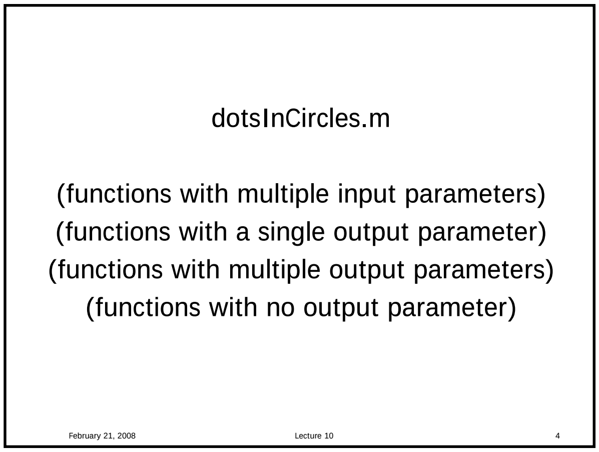


Accessing your functions

For now*, put your related functions and scripts in the same directory.

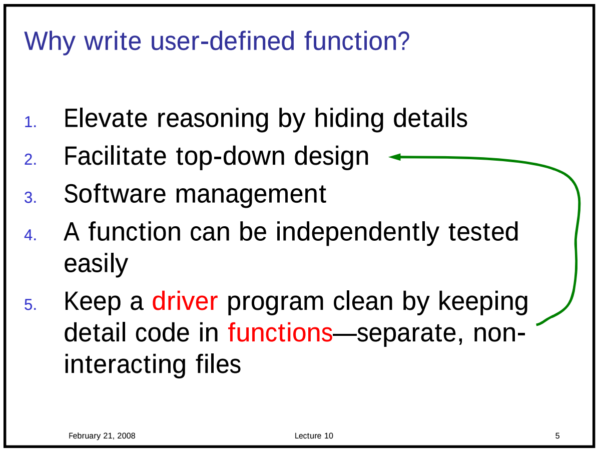
MyDirectory
dotsInCircles.m
polar2xy.m
randDouble.m
drawColorDot.m
Any script/function that calls polar2xy.m

*The `path` function gives greater flexibility. Not required in CS100M.



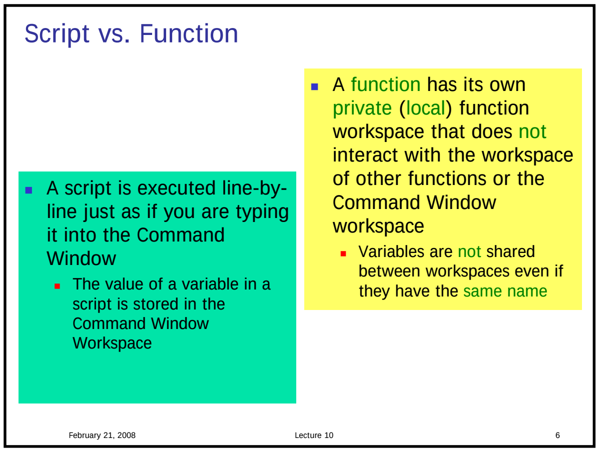
dotsInCircles.m

(functions with multiple input parameters)
(functions with a single output parameter)
(functions with multiple output parameters)
(functions with no output parameter)



Why write user-defined function?

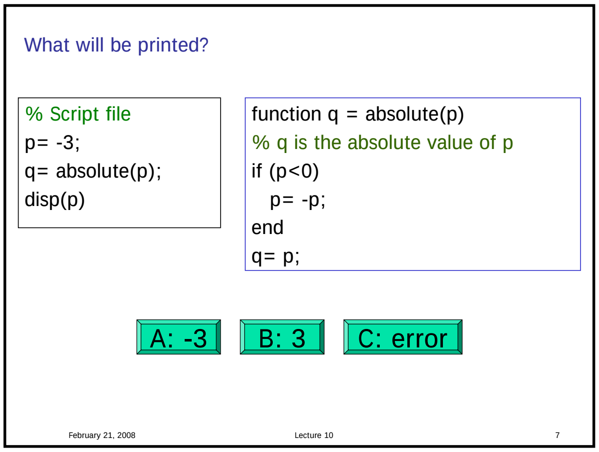
1. Elevate reasoning by hiding details
2. Facilitate top-down design
3. Software management
4. A function can be independently tested easily
5. Keep a driver program clean by keeping detail code in functions—separate, non-interacting files



Script vs. Function

- A script is executed line-by-line just as if you are typing it into the Command Window
  - The value of a variable in a script is stored in the Command Window Workspace

- A function has its own private (local) function workspace that does not interact with the workspace of other functions or the Command Window workspace
  - Variables are not shared between workspaces even if they have the same name

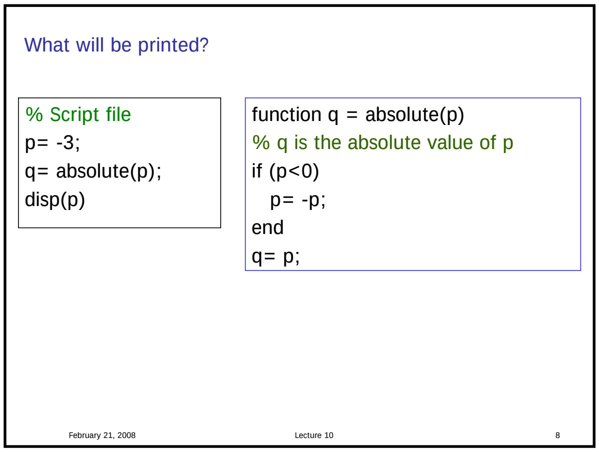


What will be printed?

```
% Script file
p= -3;
q= absolute(p);
disp(p)
```

```
function q = absolute(p)
% q is the absolute value of p
if (p<0)
  p= -p;
end
q= p;
```

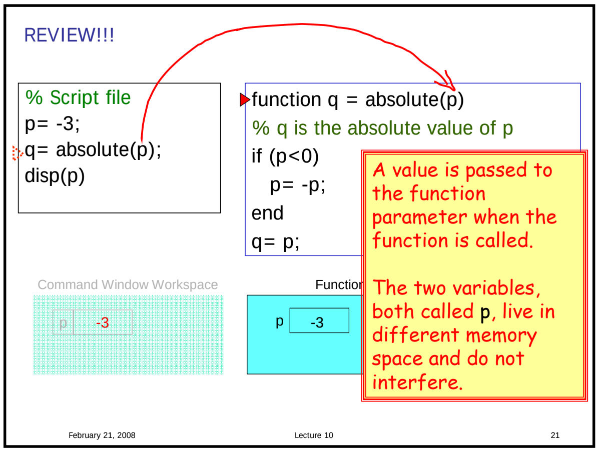
A: -3 B: 3 C: error



What will be printed?

```
% Script file
p= -3;
q= absolute(p);
disp(p)
```

```
function q = absolute(p)
% q is the absolute value of p
if (p<0)
   p= -p;
end
q= p;
```



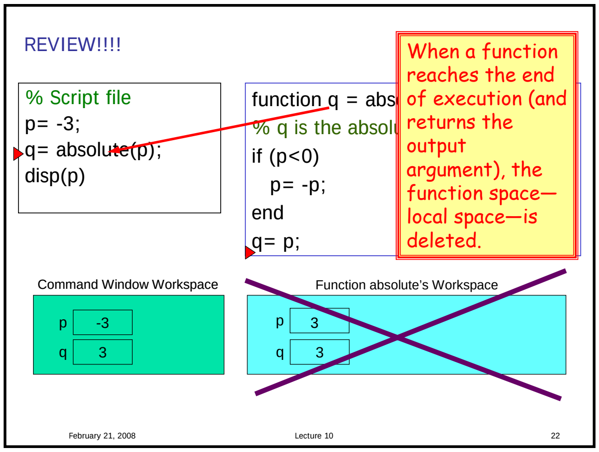
REVIEW!!!

```
% Script file
p= -3;
q= absolute(p);
disp(p)
```

```
function q = absolute(p)
% q is the absolute value of p
if (p<0)
   p= -p;
end
q= p;
```

A value is passed to the function parameter when the function is called.

The two variables, both called p, live in different memory space and do not interfere.

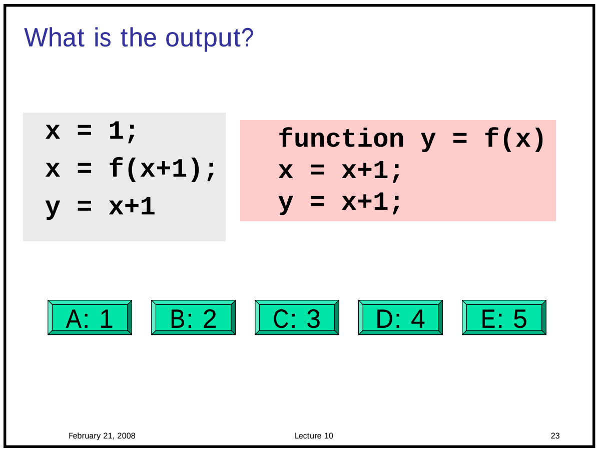


REVIEW!!!!

```
% Script file
p= -3;
q= absolute(p);
disp(p)
```

```
function q = abs
% q is the absolu
if (p<0)
   p= -p;
end
q= p;
```

When a function reaches the end of execution (and returns the output argument), the function space—local space—is deleted.

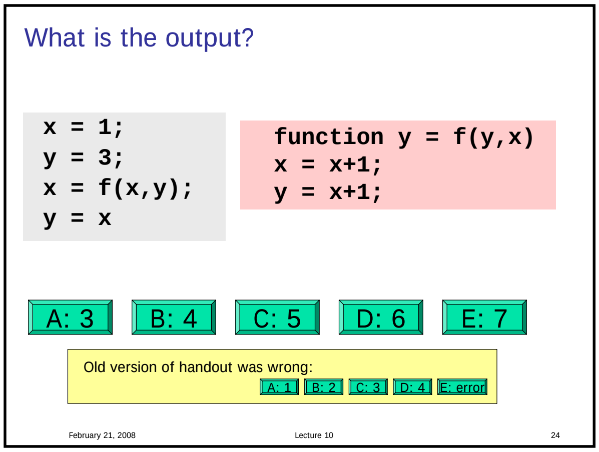


What is the output?

```
x = 1;
x = f(x+1);
y = x+1
```

```
function y = f(x)
x = x+1;
y = x+1;
```

A: 1 B: 2 C: 3 D: 4 E: 5



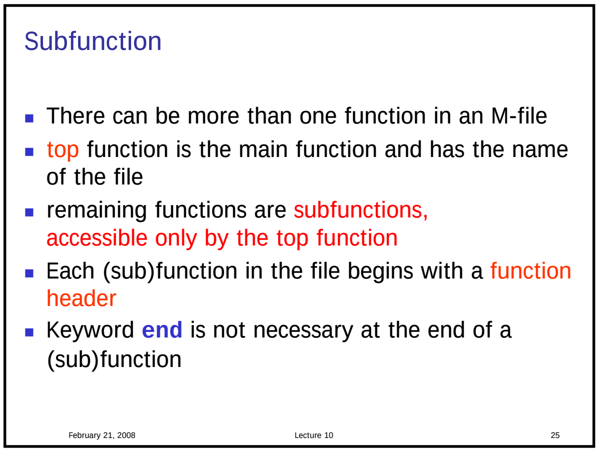
What is the output?

```
x = 1;
y = 3;
x = f(x,y);
y = x
```

```
function y = f(y,x)
x = x+1;
y = x+1;
```

A: 3 B: 4 C: 5 D: 6 E: 7

Old version of handout was wrong:
A: 1 B: 2 C: 3 D: 4 E: error



Subfunction

- There can be more than one function in an M-file
- top function is the main function and has the name of the file
- remaining functions are subfunctions, accessible only by the top function
- Each (sub)function in the file begins with a function header
- Keyword **end** is not necessary at the end of a (sub)function