

Practice Reading

Consider these (separate) m-files:

```
function y = foo(x)
y = sqrt(x);
end
```

```
function y = bar(x)
y = x^2;
end
```

```
function [r,s,t] = bang(x)
r = foo(x) + bar(x);
s = foo(bar(x));
t = bar(foo(x));
end
```

```
function f = bust(x)
[a,b,c] = bang(x);
f = a/b + c;
end
```

1. Just by looking at the code for `foo`, `bar`, `bang` and `bust`, can you tell what is the output of the following statements? Evaluate the results manually, without using MATLAB.

```
>> foo(25)
>> foo(bar(5))
>> bar(foo(256))
>> [x,y,z] = bang(5)
>> bust(2)
```

The goal of these exercises is to gain skill and confidence with the transfer of data via input and output parameters. The calculations being performed are not important.

2. Create the four m-file functions `foo.m`, `bar.m`, `bang.m` and `bust.m` in MATLAB. Verify the outcome of your manual evaluation of these functions. In other words, make sure that you understand how get the same results with pencil and paper that MATLAB gives you.

Calling a Function with Your Functions

1. Download the `quadraticRoots` code from the web site of Lab 2.
2. Write another function called `testQuadraticRoots` that supplies `quadraticRoots` with a series of sample inputs, and for each sample input, verifies whether both roots satisfy the original quadratic equation.

Hint

```
a = ... ; b = ... ; c = ... ;
r = quadraticRoots(a,b,c);
err1 = a*r(1)^2 + b*r(1) + c;
err2 = a*r(2)^2 + b*r(2) + c;
```

3. Write a subfunction (local function) that performs the calculations necessary to test the roots for a given set of a , b and c values. Use the main function to feed values of a , b and c to the subfunction.

Hint:

```
function testQuadraticRoots

verifyRoots(1,5,2);
verifyRoots(-3,5,2);

end

% -----
function verifyRoots(a,b,c)

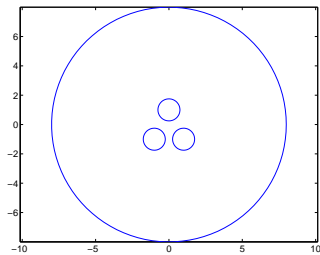
r = quadraticRoots(a,b,c);
err1 = a*r(1)^2 + b*r(1) + c;
err2 = a*r(2)^2 + b*r(2) + c;

fprintf('\na, b, c = %g, %g %g\n',a,b,c);
fprintf('err1, err1/c = %12.3e %12.3e\n',err1,err1/c);
fprintf('err2, err2/c = %12.3e %12.3e\n',err2,err2/c);

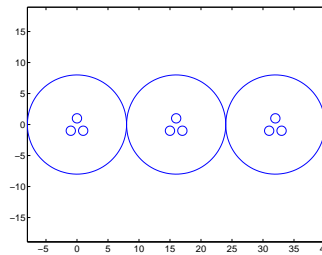
end
```

Working with Functions

1. Download the `drawCircle` and `fillCircle` codes from the web site of Lab 2.
2. Write an m-file function, say `bowlingBall`, that makes a simple cartoon of a bowling ball as shown at the bottom of the page. *Hints:* (1) Sketch the geometry first before you write the code, and (2) you will need to use `hold('on')`, `hold('off')` and `axis('equal')`.
3. Modify your bowling ball function so that it draws a bowling ball with its origin at (x_o, y_o) , where x_o and y_o are inputs. *Hint:* In addition to introducing `x0` and `y0` variables, make the radius of the ball, the radius of the finger holes, and the radial position of the finger holes all variable.
4. Write another m-file function, say `threeBalls` that calls your bowling-ball-drawing m-file three times with different (x_o, y_o) values so that the three bowling balls appear side-by-side. How many times is the `drawCircle` function called?



Output of `bowlingball`



Output of `threeBalls`