ID:_____ Name:_____

HomeWork:
MATLAB Intro & Complex Arithmetic

MATLAB as a Complex Number Calculator

- Functions used: `real()`, `imag()`, `abs()`, `angle()`
- Compare the three angle producing functions: `angle()`, `atan2()`, and `atan()`

Practice Problems (very similar to Set #1)

For each of the problem below work out the answer using both MATLAB and your calculator

1. Write $127 - j75$ in polar form; find the angle in both radians and degrees.

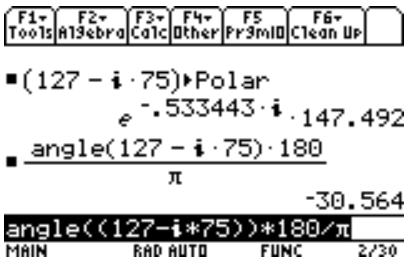
```
>> z = 127 - j*75;


>> abs(z) = ____________________


>> angle(z) = ____________________
```

Hand/Calculator workspace:

Using a TI-89



2. Write $z = 22\angle -110°$ in rectangular form.

```
>> z = 22*exp(-j*110*pi/180);


>> ____________________________________________


>> ____________________________________________
```

Hand/Calculator workspace:

3. Evaluate $z = (15 - j37) - 60\angle 45°$ to a rectangular form solution.

MATLAB Steps:

Hand/Calculator workspace:

4. Evaluate $z = (15 - j37) / 60\angle 45°$ to a polar form solution.

MATLAB Steps:

Hand/Calculator workspace:

MATLAB for Plotting Data and Functions

- Functions used: `plot()`, `xlabel()`, `ylabel()`, `title()`, `grid`, and `axis`

1. Plot $x(t) = 25\sin(\pi t/5 + \pi/4)$ for $0 \le t \le 15$ s. Include a grid and axis labels.

```
>> t = 0:.1:15; % create a time axis vector with sample spacing 0.1s
>> ?
```

For the $x(t)$ above, plot $x(t-2)$ for $0 \le t \le 15$ s, overlaid on the plot of $x(t)$ of part (1).

```
>> hold on % will hold the previous plot so you can overlay a new plot
>> ?
```

User Defined Functions in MATLAB

One of the most power capabilities of Matlab is being able write your own user defined functions. Consider a custom trig function of the form

$$y(t) = 3\cos(5t) + 4\sin(3t) \tag{1}$$

The input to this function is time, t, and the output is y. The function *prototype* we require is of the form:

```
function y = my_trig(t)
% y = my_trig(t) is a function that evaluates the simple trig
% based function y = 3*cos(5t) + 4*sin(3*t).
%
% Author: My Name
% Date: January 2011
%
…
…
function body
…
…
make sure that you return output to variable y
```

Write the Function

Test the Function

To test the function input a time vector that runs from -2s to 10s using a time step of 0.05s. Output the results in a plot using `plot(t,y)`.