

## Week 4 Answers

Feel free to copy the code provided into your own MATLAB scripts to check your work! Please remember that although these are answers, there are multiple ways of doing the problem(s). We encourage you to develop your own method of writing and organizing your code! - [Lectures](#) - [Practice Problems](#)

### Question 1

```
``` matlab %Q1 %Defining the Function numeric_Q1 = @(x) 3.x.^3 + 42.x.^2 - 13.*x + 7;

%1a: ans1a = numeric_Q1(77);

%1b: ans1b = numeric_Q1(0:100);

plot(0:100,ans1b); hold on title('Week 4 Numeric'); xlabel('x-values'); ylabel('y-values'); hold off ```
```

**Notes:** Notice how I call which values should be evaluated in part b. The values here could also be generalized using **linspace** as well. For example `matlab x_values = linspace(0,100) ans1b = numeric_Q1(x_values)` [Week 4: Numeric Functions](#)

### Question 2

```
``` matlab %Q2 %Defining the Symbols syms x y %Defining the Function symb_Q2(x,y) = 7.x.y - 12x.^2 + 8.y.^4;

%2a ans2a = symb_Q2(2,4);

%2b y_val = linspace(0,1,100); ans2b = symb_Q2(1,y_val);

%When Evaluating this, your answers may appear as fractions, to remedy
%this
ans2b = eval(ans2b);

plot(0:99,ans2b); hold on title('Week 4 Numeric'); xlabel('x-values'); ylabel('y-values'); hold off ```
```

**Notes:** Sometimes when evaluating your answers, MATLAB may give you an answer in terms of a fraction. You can turn the answer in to a number by using the command **eval** as it is used in the answer above. [MathWorks: eval](#)

Here, we define the function using two variables, however, you can also do the same in the numerical case. Similar to how we define it before: `matlab example = @(x,y) x + y` [Week 4: Symbolic Function](#)