For lab 1 in general a lack of comments were not marked down. For future labs the grading will be a little stricter, only to make sure someone new reading the code could follow along.

An example of good comments might be as follows:

In [ ]:
```
#calculating sin output for given angles
ang_1_sine = math.sin(180 * (math.pi/180))
ang_2_sine = math.sin(90 * (math.pi/180))
ang_3_sine = math.sin(45 * (math.pi/180))
ang_4_sine = math.sin(30 * (math.pi/180))

#storing the outputs into a list
sine_list = [ang_1_sine, ang_2_sine, ang_3_sine, ang_4_sine]
```

This is a rather simple example, and arguably comments probably aren't needed to understand the code. But in general giving a sentence or so at each major step of a program is useful.

For instance consider the following code:

:
```
import itertools
import numpy as np


def generate_unique_pairs_from_list(my_list):

    return list(itertools.combinations(my_list, 2))


my_list = [1,2,3,4,5,6,7,8,9]

all_pairs = generate_unique_pairs_from_list(my_list)

print("All Pairs Before Filter", all_pairs)

all_pairs_filter = [pair[0] > 2 for pair in all_pairs]

all_pairs = [pair for (pair, remove) in zip(all_pairs, all_pairs_filter) if remove]

print("All Pairs After Filter", all_pairs)


```

```
All Pairs Before Filter [(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (2, 3), (2, 4), (2, 5), (2, 6), (2,
7), (2, 8), (2, 9), (3, 4), (3, 5), (3, 6), (3, 7), (3, 8), (3, 9), (4, 5), (4, 6), (4, 7), (4, 8), (4, 9), (5, 6), (5, 7), (5,
8), (5, 9), (6, 7), (6, 8), (6, 9), (7, 8), (7, 9), (8, 9)]
All Pairs After Filter [(3, 4), (3, 5), (3, 6), (3, 7), (3, 8), (3, 9), (4, 5), (4, 6), (4, 7), (4, 8), (4, 9), (5, 6), (5, 7),
(5, 8), (5, 9), (6, 7), (6, 8), (6, 9), (7, 8), (7, 9), (8, 9)]
```

After looking at it for a while it might be clear what the code is doing .But comments would help. Here is the same code with comments:

```python
import itertools
import numpy as np


#a function that returns the unique
#pairs from the input list
def generate_unique_pairs_from_list(my_list):

    #package itertools finds all unique combinations
    #for us
    return list(itertools.combinations(my_list, 2))


my_list = [1,2,3,4,5,6,7,8,9]

#create the combination list
all_pairs = generate_unique_pairs_from_list(my_list)

#print the list prior to filtering
print("All Pairs Before Filter", all_pairs, "\n")

#generate and apply a filter that to only
#keep pairs whose first number is greater than 2
all_pairs_filter = [pair[0] > 2 for pair in all_pairs]
all_pairs = [pair for (pair, remove) in zip(all_pairs, all_pairs_filter) if remove]

#print the list after filtering
print("All Pairs After Filter", all_pairs, "\n")


```

```
All Pairs Before Filter [(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (2, 3), (2, 4), (2, 5), (2, 6), (2,
7), (2, 8), (2, 9), (3, 4), (3, 5), (3, 6), (3, 7), (3, 8), (3, 9), (4, 5), (4, 6), (4, 7), (4, 8), (4, 9), (5, 6), (5, 7), (5,
8), (5, 9), (6, 7), (6, 8), (6, 9), (7, 8), (7, 9), (8, 9)]

All Pairs After Filter [(3, 4), (3, 5), (3, 6), (3, 7), (3, 8), (3, 9), (4, 5), (4, 6), (4, 7), (4, 8), (4, 9), (5, 6), (5, 7),
(5, 8), (5, 9), (6, 7), (6, 8), (6, 9), (7, 8), (7, 9), (8, 9)]
```

With just a little bit more comments the code feels a lot more approachable. It can be seen the code gets all unique pairs from a list, then removes all pairs who don't have their first number larger than 2. It also **helps to choose good variable and function names**. Without the comments it's still possible to get an idea of what's going on based on the variable and function names. The comments should just be there to help guide whoever is reading your code, and make it easy for them to jump right in.

This **doesn't mean you have to write a ton of paragraphs** of comments. A sentence is usually enough, maybe if the idea is a little complex two can be helpful. The goal is not to overwhelm the reader, but just to give them a good entry point if they've never seen your code before.

Comments aren't always necessary, but **in general maybe a sentence or two describing how to get from a to b is enough** just to make sure everyone is documenting their code.