In [10]:

```
print('Part 1: Object Oriented Programming')
print('1.Replace  pass with the appropriate code in the Line class methods to

import math

class Line():


    def __init__(self,coor1,coor2):
            self.coor1 = coor1
            self.coor2 = coor2
            #return line coordinate values as list
            for x in range (0, len(coor2)):
                   coor1.append(coor2[x])


    def distance(self):
        dis = str(abs(math.sqrt(((li.coor1[2]-li.coor1[0])**2 + (li.coor1[3]-
        print('li.distance() #' + dis)


    def slope(self):
        slp = str((li.coor1[3]-li.coor1[1])/(li.coor1[2]-li.coor1[0]))
        print('li.slope() # ' + slp)



cord1 = list(map(int,input("\nEnter coordinate 1 values : ").strip().split())
cord2 = list(map(int,input("\nEnter coordinate 2 values : ").strip().split())


#print inputs
coor = str(cord1)
print('Coordinate 1 =' + coor)
coor = str(cord2)
print('Coordinate 2 =' + coor)

Lst = [cord1,cord2]
Ls = str(Lst)
#print line coordinates
print('li = Line' + Ls)

li = Line(cord1,cord2)


li.distance()

li.slope()
```

```
Part 1: Object Oriented Programming
1.Replace  pass with the appropriate code in the Line class methods to ac
cept coordinates as a pair of lists and return the slope and distance of
the line.

Enter coordinate 1 values : 3 2
```

```
Enter coordinate 2 values : 8 10
Coordinate 1 =[3, 2]
Coordinate 2 =[8, 10]
li = Line[[3, 2], [8, 10]]
li.distance() #9.433981132056603
li.slope() # 1.6
```

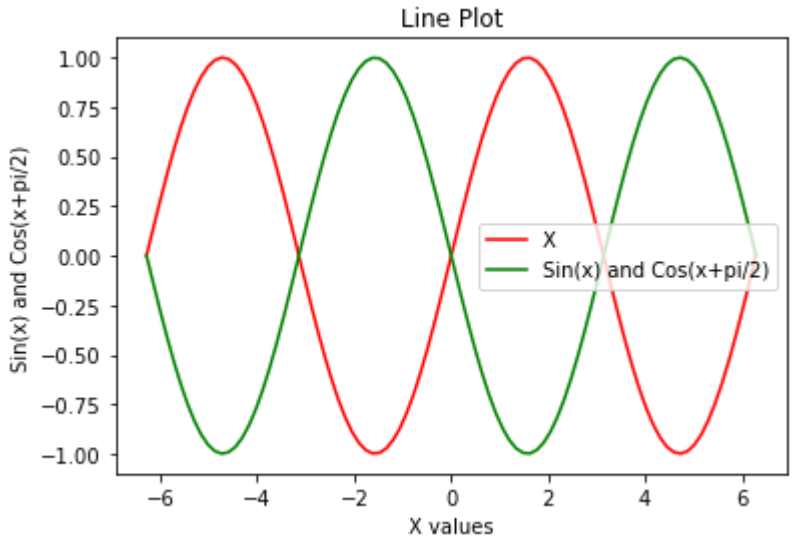
In [11]:

```
print('2.Replace  pass with the appropriate code in the Cylinder class method
import math

class Cylinder(object):

    def __init__(self,height=1,radius=1):
        self.pival = round(math.pi,2)
        self.height = height
        self.radius = radius
        h = str(height)
        r = str(radius)
        print('\nc = Cylinder(' + h + ',' + r + ')')
    def volume(self):
        Vol = str(round(c.pival*c.radius* c.radius *c.height,2))
        print('c.volume() #' + Vol)
    def surface_area(self):
        Sa = str(round(2*c.pival*c.radius* (c.radius + c.height),1))
        print('c.surface_area() #' + Sa)


hei = int(input("\nEnter height of cylinder : "))
rad = int(input("\nEnter radius of cylinder : "))

c = Cylinder(hei,rad)

c.volume()

c.surface_area()
```

```
2.Replace  pass with the appropriate code in the Cylinder class methods to
return the volume and the surface area of the cylinder.

Enter height of cylinder : 2

Enter radius of cylinder : 3

c = Cylinder(2,3)
c.volume() #56.52
c.surface_area() #94.2
```

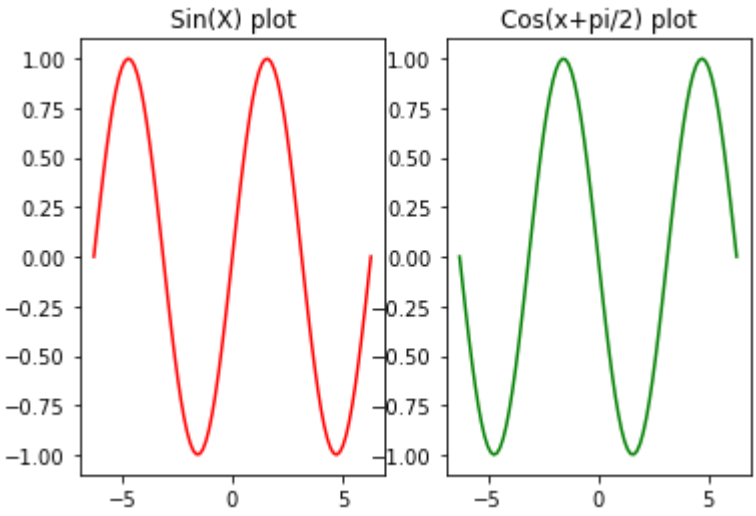
In [9]:
```
print('Part 2: 2D Plot')
print('1. Create a line plot of sin(x) and cos(x + π/2) for -2π < x < 2π wher
print('i. Make the sin(x) graph red and make the cos(x+π/2) graph green, Put

import matplotlib.pyplot as plt
from pylab import *
import math
import numpy as np

#x increases at intervals of π/4 i.e it generates 17 points, by changing 90 t
x = np.linspace(-2*math.pi , 2*math.pi, 90)

plt.plot(x , np.sin(x), color = 'r' )
plt.plot(x, np.cos(x+math.pi/2), color = 'g')
plt.title('Line Plot')
plt.xlabel('X values')
plt.ylabel('Sin(x) and Cos(x+pi/2)')
plt.legend(['X', 'Sin(x) and Cos(x+pi/2)'])
plt.show()

print('ii. Using the same info as above, make a subplot with 2 different grap

subplot(1,2,1)
title('Sin(X) plot')
plot(x,np.sin(x), color = 'r')

subplot(1,2,2)
title('Cos(x+pi/2) plot')
plot(x, np.cos(x+math.pi/2), color = 'g')
```

```
Part 2: 2D Plot
1. Create a line plot of sin(x) and cos(x + π/2) for -2π < x < 2π where x i
ncreases at intervals of π/4.
i. Make the sin(x) graph red and make the cos(x+π/2) graph green, Put both
lines onto the same plot
```



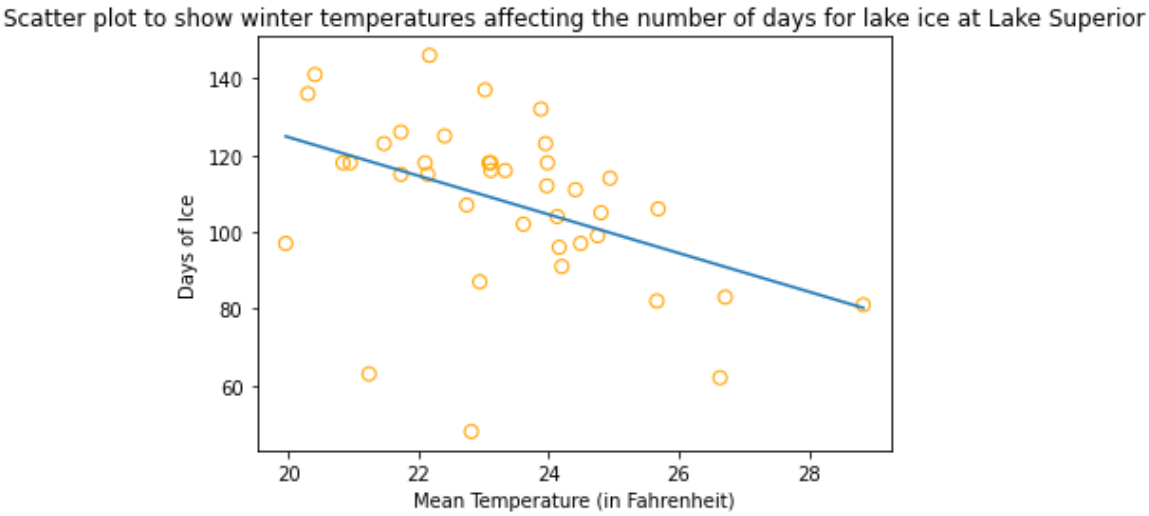
ii. Using the same info as above, make a subplot with 2 different graphs- one graph for sin(x) and one graph for cos(x+π/2)

Out[9]: [<matplotlib.lines.Line2D at 0x2052e839d60>]



In [12]:
```
print('2. Scatter Plot:')
print('Using the following data about winter temperatures affecting the numbe

import numpy as np

temperatures = [22.94, 23.02, 25.68, 19.96, 24.80, 23.98, 22.10, 20.30, 24.20
Days_of_ice = [87, 137, 106, 97, 105, 118, 118, 136, 91, 107, 96, 114, 125, 1

plt.scatter(temperatures, Days_of_ice, s=50, facecolor = 'none', edgecolors='

plt.xlabel('Mean Temperature (in Fahrenheit)')
plt.ylabel('Days of Ice')
plt.title('Scatter plot to show winter temperatures affecting the number of d
plt.plot(np.unique(temperatures),np.poly1d(np.polyfit(temperatures,Days_of_ic

#show scatter plot
plt.show()
```

2. Scatter Plot:
Using the following data about winter temperatures affecting the number of days for lake ice at Lake Superior, construct a scatter plot to display the data. Include a line of best fit.



In [ ]:
```

```