Assignment 2: OOP and 2D Plots

Part 1: Object Oriented Programming

```
1. Replace  pass with the appropriate code in the Line class methods to accept
coordinates as a pair of lists and return the slope and distance of the line.
class Line(object):
    def __init__(self,coor1,coor2):
        pass
    def distance(self):
        pass
    def slope(self):
        pass
Example output:
coordinate1 = [3,2]
coordinate2 = [8,10]
li = Line(coordinate1,coordinate2)
li.distance()       # 9.433981132056603
li.slope()      # 1.6
```

In [2]:

```
import matplotlib.pyplot as plt
import numpy as np
class Line(object):
    def __init__(self,coor1,coor2):
        self.coor1=coor1
        self.coor2=coor2
    def distance(self):
        dist=np.sqrt(((self.coor2[0]-self.coor1[0])*
                      (self.coor2[0]-self.coor1[0]))+
                     ((self.coor2[1]-self.coor1[1])*
                      (self.coor2[1]-self.coor1[1])))
        return dist;
    def slope(self):
        s=(self.coor2[1]-self.coor1[1])/(self.coor2[0]-self.coor1[0])
        return s;
coordinate1 = [int(i) for i in input().split(',')]
coordinate2 = [int(i) for i in input().split(',')]
if(coordinate2[0]==coordinate1[0]):
    print("infinite")
else:
    li = Line(coordinate1,coordinate2)
    print("distance: ",li.distance())
    print("slope: ",li.slope())
```

```
3,2
8,10
distance:  9.433981132056603
slope:  1.6
```

```
2. Replace  pass with the appropriate code in the Cylinder class methods to
return the volume and the surface area of the cylinder.
class Cylinder(object):
    def __init__(self,height=1,radius=1):
        pass
    def volume(self):
        pass
    def surface_area(self):
        pass
Example output:
c = Cylinder(2,3)
c.volume()      # 56.52
c.surface_area()    # 94.2
```

In [3]:

```
import numpy as np
class Cylinder(object):
    def __init__(self,height=1,radius=1):
        self.height=height
        self.radius=radius
    def volume(self):
        vol=(np.pi)*self.radius*self.radius*self.height
        return vol;
    def surface_area(self):
        s_a=(2*(np.pi)*self.radius*self.radius)+
        (2*np.pi*self.radius*self.height)
        return s_a;
c1=[int(x) for x in input().split(',')]
c = Cylinder(c1[0],c1[1])
print("volume: ",round(c.volume(),2))
print("surface_area: ",round(c.surface_area(),1))
```

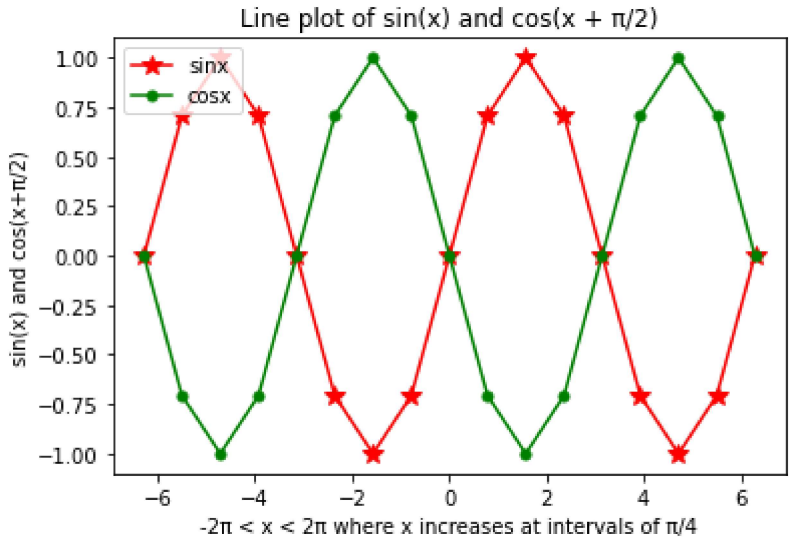
```
2,3
volume:  56.55
surface_area:  94.2
```

Part 2: Simple Charts

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

2) 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)

x = -6.283, -5.498, -4.712, -3.927, -3.142, -2.356, -1.571, -.7854, 0, .7854,
1.571, 2.356, 3.142, 3.927, 4.712, 5.498, 6.283

sin(x) = 0, .70711, 1, .70711, 0, -.70711, -1, -.70711, 0, .70711, 1, .70711,
0, -.70711, -1, -.70711, 0

cox(x + π/2) = 0, -.70711, -1, -.70711, 0, .70711, 1, .70711, 0, -.70711, -1,
-.70711, 0, .70711, 1, .70711, 0
```

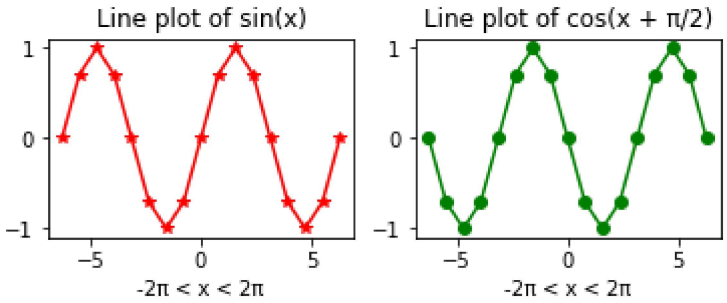
In [4]:
```
import matplotlib.pyplot as plt
import numpy as np
x =[-6.283, -5.498, -4.712, -3.927, -3.142,
    -2.356, -1.571, -.7854, 0, .7854,
    1.571, 2.356, 3.142, 3.927, 4.712, 5.498, 6.283]
t=(np.arange(-2*np.pi,9*(np.pi/4),np.pi/4))
plt.plot(t,np.sin(t),color='red',marker='*',markersize=10)
li=[i+(np.pi/2) for i in x]
plt.plot(t,np.cos(li),color='green',marker='o',markersize=5)
plt.title('Line plot of sin(x) and cos(x + π/2)')
plt.xlabel('-2π < x < 2π where x increases at intervals of π/4')
plt.ylabel('sin(x) and cos(x+π/2)')
plt.legend(['sinx','cosx'],loc='upper left')
```

Out[4]: <matplotlib.legend.Legend at 0x2e091798cd0>



In [5]:
```
import matplotlib.pyplot as plt
import numpy as np
#first plot
x =[-6.283, -5.498, -4.712, -3.927, -3.142, -2.356, -1.571, -.7854, 0,
    .7854, 1.571, 2.356, 3.142, 3.927, 4.712, 5.498, 6.283]
plt.subplot(2,2,1)
plt.plot(x,np.sin(x),color='red',marker='*')
plt.title('Line plot of sin(x)')
plt.xlabel('-2π < x < 2π')
#second plot
plt.subplot(2,2,2)
li=[i+(np.pi/2) for i in x]
plt.plot(x,np.cos(li),color='green',marker='o')
plt.title('Line plot of cos(x + π/2)')
plt.xlabel('-2π < x < 2π')
```

Out[5]: Text(0.5, 0, '-2π < x < 2π')



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.
Mean Temperature (in Fahrenheit): 22.94, 23.02, 25.68, 19.96, 24.80, 23.98, 22.10, 20.30, 24.20, 22.74, 24.16, 24.94, 22.40, 22.14, 20.84, 25.66, 21.73, 24.49, 24.13, 22.17, 21.73, 20.41, 24.41, 23.95, 20.95, 26.71, 22.81, 23.11, 23.33, 28.83, 23.11, 21.47, 23.97, 24.75, 23.61, 23.08, 21.24, 26.63, 23.88
Days of Ice: 87, 137, 106, 97, 105, 118, 118, 136, 91, 107, 96, 114, 125, 115, 118, 82, 115, 97, 104, 146, 126, 141, 111, 123, 118, 83, 48, 118, 116, 81, 116, 123, 112, 99, 102, 118, 63, 62, 132

In [6]:

```
import matplotlib.pyplot as plt
import numpy as np
temp=np.array([22.94, 23.02, 25.68, 19.96, 24.80, 23.98, 22.10,
               20.30, 24.20, 22.74, 24.16,24.94, 22.40, 22.14,
               20.84, 25.66, 21.73, 24.49, 24.13, 22.17, 21.73,
               20.41, 24.41, 23.95, 20.95, 26.71, 22.81, 23.11,
               23.33, 28.83, 23.11,21.47, 23.97, 24.75, 23.61,
               23.08, 21.24, 26.63, 23.88])
Days=np.array([87, 137, 106, 97, 105, 118, 118, 136, 91, 107,
               96, 114, 125,115, 118, 82, 115, 97, 104, 146,
               126, 141, 111, 123,118,83, 48,118, 116, 81,
               116, 123, 112, 99, 102, 118, 63, 62, 132])
plt.scatter(temp,Days,color='teal',s=50,marker="x")
plt.title('Number of Days vs. Temperature at Lake Superior')
plt.xlabel('Mean Temperature at Lake Superior(in Fahrenheit)')
plt.ylabel('Number of Days')
plt.plot(np.unique(temp),np.poly1d(np.polyfit(temp,Days,1))(np.unique(temp)))
plt.grid()
plt.show()
```