## Exercise 1

```
import math
class Line():
  def __init__(self,coor1,coor2):
      """Initialize instance attributes with tuples (x1,y1)  and (x2,y2)
      """
      self.x1=coor1[0]
      self.y1=coor1[1]
      self.x2=coor2[0]
      self.y2=coor2[1]

  def distance(self):
      """Calculate the length of the segment (line)
      """
      return math.sqrt((self.x1-self.x2)**2+(self.y1-self.y2)**2)

  def slope(self):
      """ Return the slope of a line going through the ends ( the 'a' in y=ax+b)
      """
      return (self.y1-self.y2)/(self.x1-self.x2)

coordinate1 = (3,2)
coordinate2 = (8,10)

li = Line(coordinate1,coordinate2)
print(li.distance())
print(li.slope())
```

```
9.433981132056603
1.6
```

## Exercise 2

```
import math
class Cylinder(object):
  def __init__(self,height=1,radius=1):
      self.height=height
      self.radius=radius

  def volume(self):
      return math.pi*self.radius**2*self.height

  def surface_area(self):
      return 2*math.pi*self.radius**2+2*math.pi*self.radius*self.height

c = Cylinder(2,3)
print(c.volume())
print(c.surface_area())
```

```
56.548667764616276
94.24777960769379
```

## Exercise 3

```
#here i had a problem with example dataset
class DataFile(object):

  def __init__(self, filename='undef'):
    with open(filename) as f:
      header = f.readline()
      self.cols = list(map(str.rstrip,header.split(';')))
```

```
        r = 0
        self.mat = []
        for line in iter(f.readline, ''):
          self.mat.append([])
          for c in line.split(';'):
            try:
              self.mat[r].append(float(c))
            except:
              self.mat[r].append(c.rstrip())
          r += 1


    def info(self):
      print(f"{'':30} {'Min':^10} {'Max':^10} {'Avg':^10}")
      for cn in self.cols:
        mn=self.min(colname=cn)
        mx=self.max(colname=cn)
        ma=self.avg(colname=cn)
        print(f"{cn:<30} {mn:^10} {mx:^10} {ma:^10}")


    def avg(self, colnum=0, colname=''):
      """ The column name or colnum can be provided alternatively
      """
      icol = -1
      if colname != '':
        icol = self.cols.index(colname)
      s = 0;
      for r in self.mat:
        try:
          s += float(r[icol])
        except:
          pass

      return s / len(self.mat)

    def min(self, colnum=0, colname=''):
      icol = -1
      if colname != '':
        icol = self.cols.index(colname)
      m = 100000;
      for r in self.mat:
        try:
          v = float(r[icol])
          if (v < m):
            m = v
        except:
          pass
      return m

    def max(self, colnum=0, colname=''):
      return 0.0
```