Final Exam Practice Problems

OOP1

Complete the missing code below.

```
# file point.py
class Point:
''' Models a point in the plane using Cartesian coordinates '''

   def __init__(self,x_val = 0,y_val = 0)
   ''' Initializes a point object with float value x_val,y_val
   '''
      self.x = x_val   # x-coordinate of the point
      self.y = y_val   # y-coordinate of the point

   def move(self,dx,dy):
   ''' Shifts the point dx units along the x-axis
       and dy units along the y-axis
   '''
   # fill in the code

   def dist(self,other):
   ''' returns the distance from Point self to Point other
   '''
   # fill in the code

if __name__ == "__main__":
# Create a Point corresponding to the origin, then create a
# second Point whose coordinates you input from the user.
# Then print the distance between the two points.
# Next, input a horizontal shift amount and a vertical shift
# amount and move the user-defined point by those amounts.
# Finally, print a message that shows the new location of the
# Point
```

OOP2

Define a new method for your point class so that execution of the following code

```
P = Point(2.5,3.6)
print('Here is my Point:',P)
```

produces the following output:

```
Here is my Point: (2.5,3.6)
```

OOP3

Complete the missing code below.

```
from point import Point
import math

class Circle:
''' Models a circle in the plane a center Point and a radius '''

   def __init__(self,center_point,r):
      self.center = center_point
      self.radius = r

   def get_area(self):
   ''' returns the area of the circle '''
   # complete the code


   def move(self,dx,dy):
   ''' shifts the circle dx units along the x-axis
       and dy units along the y-axisl
   '''
   # complete the code


   def contains(self,other):
   ''' returns True if and only if circle other is completely
       contained inside circle self
   '''
   # complete the code; use pictures to guide you work

def getCircle():
''' Inputs the necessary values from the user to create and a
    circle and returns that circle
'''
# complete the code

if __name__ == "__main__":
# Input two circles from the user (getCircle), then print a
# message indicating whether the second circle is contained in
# the first
```