```python
# CS241 Team Activity 06

class Point:

    def __init__(self):
        self.x = 0.0
        self.y = 0.0

    def prompt_for_point(self):
        self.x = float(input("Enter x: "))
        self.y = float(input("Enter y: "))

    def display(self):
        print("({},{})" .format(self.x, self.y))

"""
The IS-A relationship means that Circle has all the
member data and functions from the Point class.  We can
just say "self." to use these.
"""
class Circle(Point):

    def __init__(self):
        super().__init__()
        self.radius = 0.0

    def prompt_for_circle(self):
        self.prompt_for_point()
        self.radius = float(input("Enter radius: "))

    def display(self):
        print("Center:")
        super().display()
        print("Radius: {}" .format(self.radius))

"""
The HAS-A relationship means that Circle has a Point object (self.center)
We will need to use "self.center." to access any member data or member
function from the Point class.
"""
class Circle_HASA():

    def __init__(self):
        self.center = Point()
        self.radius = 0.0

    def prompt_for_circle(self):
        self.center.prompt_for_point()
        self.radius = float(input("Enter radius: "))

    def display(self):
        print("Center:")
        self.center.display()
        print("Radius: {}" .format(self.radius))


def main():
    c1 = Circle()
    c1.prompt_for_circle()
    c1.display()

    c2 = Circle_HASA()
    c2.prompt_for_circle()
    c2.display()

if __name__ == "__main__":
    main()


```