

```

1  # CS241 Team Activity 06
2
3  class Point:
4
5      def __init__(self):
6          self.x = 0.0
7          self.y = 0.0
8
9      def prompt_for_point(self):
10         self.x = float(input("Enter x: "))
11         self.y = float(input("Enter y: "))
12
13     def display(self):
14         print("{}, {}".format(self.x, self.y))
15
16     """
17     The IS-A relationship means that Circle has all the
18     member data and functions from the Point class. We can
19     just say "self." to use these.
20     """
21
22     class Circle(Point):
23
24         def __init__(self):
25             super().__init__()
26             self.radius = 0.0
27
28         def prompt_for_circle(self):
29             self.prompt_for_point()
30             self.radius = float(input("Enter radius: "))
31
32         def display(self):
33             print("Center:")
34             super().display()
35             print("Radius: {}".format(self.radius))
36
37         """
38         The HAS-A relationship means that Circle has a Point object (self.center)
39         We will need to use "self.center." to access any member data or member
40         function from the Point class.
41         """
42
43         class Circle_HASA():
44
45             def __init__(self):
46                 self.center = Point()
47                 self.radius = 0.0
48
49             def prompt_for_circle(self):
50                 self.center.prompt_for_point()
51                 self.radius = float(input("Enter radius: "))
52
53             def display(self):
54                 print("Center:")
55                 self.center.display()
56                 print("Radius: {}".format(self.radius))
57
58     def main():
59         c1 = Circle()
60         c1.prompt_for_circle()
61         c1.display()
62
63         c2 = Circle_HASA()
64         c2.prompt_for_circle()
65         c2.display()
66
67     if __name__ == "__main__":

```

```
67     main()  
68  
69
```