

```

1  """
2  CS241 Checkpoint 07B
3  Written by Chad Macbeth
4  """
5
6  """
7  File: check07b.py
8  Author: Br. Burton
9
10 Demonstrates abstract base classes.
11 """
12
13 from abc import ABC
14 from abc import abstractmethod
15
16 ### Shape is an abstract class that requires subclasses
17 ### to implement the get_area function.
18 class Shape(ABC):
19     def __init__(self):
20         self.name = ""
21
22     def display(self):
23         # This function calls the abstract method get_area
24         # The individual derived classes will provide an
25         # implementation for get_area.
26         print("{} - {:.2f}".format(self.name, self.get_area()))
27
28     @abstractmethod
29     def get_area(self):
30         pass
31
32
33 class Circle(Shape):
34
35     def __init__(self):
36         super().__init__()
37         self.name = "Circle"
38         self.radius = 0.0
39
40     def get_area(self): # Notice that in subclass, the get_area is not abstract
41         return 3.14 * self.radius * self.radius
42
43
44 class Rectangle(Shape):
45
46     def __init__(self):
47         super().__init__()
48         self.name = "Rectangle"
49         self.length = 0.0
50         self.width = 0.0
51
52     def get_area(self):
53         return self.length * self.width
54
55 def main():
56
57     shapes = []
58     command = ""
59
60     while command != "q":
61         command = input("Please enter 'c' for circle, 'r' for rectangle or 'q' to quit:
62             ")
63
64         if command == "c":
65             radius = float(input("Enter the radius: "))
66             c = Circle()

```

```
66         c.radius = radius
67         shapes.append(c)
68
69     elif command == "r":
70         length = float(input("Enter the length: "))
71         width = float(input("Enter the width: "))
72         r = Rectangle()
73         r.length = length
74         r.width = width
75         shapes.append(r)
76
77
78     # Done entering shapes, now lets print them all out:
79
80     for shape in shapes:
81         shape.display()    # Polymorphism ... calling the abstract method and python will
82                             # figure out which get_area to call
83
84 if __name__ == "__main__":
85     main()
86
87
```