Classes and Objects

Create and save the following code in a file called **Circle.py**

```
import math
class Circle:
    # Construct a circle object
    def __init__(self, radius = 1):   # note, use two underscore characters
        self.radius = radius  # radius is a public variable

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

    def getArea(self):
        return self.radius * self.radius * math.pi

    def setRadius(self, radius):
        self.radius = radius
```

Create and save the following code in a file called **TestCircle.py**

```
# Given the radius, display the area of a circle #
from Circle import Circle

def main():
    # Create a circle1 with radius 1
    circle1 = Circle()
    print("The area of the circle of radius", circle1.radius, "is %.2f" % circle1.getArea())

    # Create a circle with radius 25
    circle2 = Circle(25)
    print("The area of the circle of radius", circle2.radius, "is %.2f" % circle2.getArea())

    # Create a circle3 with radius 125
    circle3 = Circle(125)
    print("The area of the circle of radius", circle3.radius, "is %.2f" % circle3.getArea())

    # Modify circle2 radius
    circle2.radius = 100
    print("The area of the circle of radius", circle2.radius, "is %.2f" % circle2.getArea())
    print("The perimeter of the circle of radius", circle2.radius, "is %.2f" %
circle2.getPerimeter())

    #Call the main function
    main()
```

```
========= RESTART: /Users/staff/Downloads/TestCircle.py =============== The area of the
circle of radius 1 is 3.14
The area of the circle of radius 25 is 1963.50
The area of the circle of radius 125 is 49087.39

The area of the circle of radius 100 is 31415.93 The perimeter of the circle of radius 100 is
628.32
```

Create and save the following code in a file called **Circle.py**

```
import math
class Circle:
    # Construct a circle object
    def __init__(self, radius = 1):   # note, use two underscore characters
        self.radius = radius  # radius is a public variable

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

    def getArea(self):
        return self.radius * self.radius * math.pi

    def setRadius(self, radius):
        self.radius = radius
```

How to create python library and import them in your program

1. First, create all the needed functions and put them in a file with .py extension.
2. Next, write all your functions and save them into a single file. You can use any name. Example: The following functions are saved in a file called myLib.py. Note: Your library program and your main program must be in the same directory.

```
def sum(x, y):
  return(x+y)
```

```
def product (x, y):
   return(x*y)

def diff(x, y):
   return(x-y)
```

Now, you can write your program and import all your functions:

```
’’’
This program demonstrates how to create modules and then import them in your program.
Author: Sam
’’’
from myLib import *

x = eval(input("Enter a number: "))
y = eval(input("Enter another number: "))

total = sum(x,y)
print("The sum is", total)

p = product(x,y)
print("The product is", p)
```

```
Enter a number: 5
Enter another number: 3
The sum is 8
The product is 15
```

By default, Python does not allow library located on different folder. They must be in the same folder. However, if your library in in a different folder, you can modify your import statement as following:

```
import sys
sys.path.insert(0, '/Users/downloads/TEMP/')  # your path may be different here
from myLib import *
```

Circle class using private fields

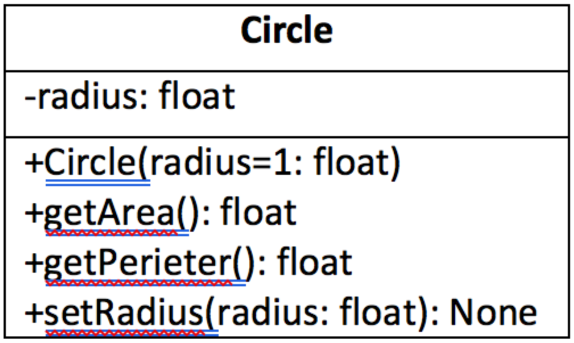
Create and save the following code in a file called **Circle.py**

```
import math
class Circle:
    # Construct a circle object
    def __init__(self, radius = 1):   # note, use two dashes, init, and then two more underscores
        self.__radius = radius  # radius is private. Use two underscore characters

    def getPerimeter(self):
        return 2 * self.__radius * math.pi

    def getArea(self):
        return self.__radius * self.__radius * math.pi

    def setRadius(self, radius):
        self.__radius = radius

    def getRadius(self, radius):
        return self.__radius
```

Create and save the following code in a file called **TestCircle.py**

```
# Given the radius, display the area of a circle #
from Circle import Circle

def main():
    # Create a circle1 with radius 1
    circle1 = Circle()
    print("The area of the circle of radius", circle1.getRadius(), "is %.2f" % circle1.getArea())

    # Create a circle with radius 25
    circle2 = Circle(25)
    print("The area of the circle of radius", circle2.getRadius(), "is %.2f" % circle2.getArea())

    # Create a circle3 with radius 125
    circle3 = Circle(125)
    print("The area of the circle of radius", circle3.getRadius(), "is %.2f" % circle3.getArea())
    # Modify circle2 radius
    circle2.setRadius(100)
    print("The area of the circle of radius", circle2.getRadius(), "is %.2f" % circle2.getArea())

    print("The perimeter of the circle of radius", circle2.getRadius(), "is %.2f" %
circle2.getPerimeter())

    #Call the main function
    main()
```

Unified Modeling Language (UML) class diagram



Circle

-radius: float

+Circle(radius=1: float)
+getArea(): float
+getPerieter(): float
+setRadius(radius: float): None

Symbols
- private
+ publlic