UNIVERSITY OF ENGINEERING AND TECHNOLOGY PESHAWAR, JALOZAI CAMPUS

Lab : Classes 2 –repr-- and –str--

Lab Title: EE-271 "OOP & Data Structures Lab"

Time: 10 min/ Task

Lab report task:

1. Define a class circle. Your class must have the appropriate __init__ method.
    i. Add appropriate property methods. (@property and @property.setter)
    ii. In addition, add an instance method for the volume of a cylinder with the given radius.
    iii. Add property method for area, circumference, and diameter.
    iv. Add __repr__ and __str__ to the class.
    v. Add proper annotation and doc string to every class and instance method.
    a. Define inst_1 and pass two numbers.
    b. Make another instance inst2.
    c. Print inst_1 and inst_2, for this use the print command and pass the inst_1 and inst_2.
    d. Call the __dict__ by the class name.
    e. Also pass the class name to the vars built-in function.
    f. Call the __dict__ on the object of the class.
    g. Pass the class name to help.
    h. Print the doc-string and annotations of both the class and each instance method.
    i. Modify the __init__ by making its parameters default and verify by instances.

Lab work tasks:

1. Define a class point in 2 dimension coordinate system. Your class must have the appropriate __init__ method.
    i. Add appropriate property methods. (@property and @property.setter)
    ii. In addition, add instance method for the distance between points.
    iii. Add another instance method for calculating the distance from origin.
    iv. Add another method with the name locate to display the coordinate in which the point is located.
    v. Add __repr__ and __str__ to the class.
    vi. Add proper annotation and doc string to every class and instance method.
    a. Define inst_1 and pass two numbers.
    b. Make another instance inst2.
    c. Print inst_1 and inst_2, for this use the print command and pass theinst_1 and inst_2.
    d. Also print the location of both points. (Add a display method).

e. Calculate the distance between these two pints.

f. Call the __dict__ by the class name.

g. Also pass the class name to the vars built-in function.

h. Call the __dict__ on the object of the class.

i. Pass the class name to help.

j. Print the doc-string and annotations of both the class and each instance method.

k. Modify the __init__ by by making its parameters default and verify by instances.

2. Define a class circle. Your class must have the appropriate __init__ method.
    i. Add appropriate property methods. (@property and @property.setter)
    ii. In addition, add an instance method for the volume of a cylinder with the given radius.
    iii. Add property method for area, circumference, and diameter.
    iv. Add __repr__ and __str__ to the class.
    v. Add proper annotation and doc string to every class and instance method.

    j. Define inst_1 and pass two numbers.

    k. Make another instance inst2.

    l. Print inst_1 and inst_2, for this use the print command and pass the inst_1 and inst_2.

    m. Call the __dict__ by the class name.

    n. Also pass the class name to the vars built-in function.

    o. Call the __dict__ on the object of the class.

    p. Pass the class name to help.

    q. Print the doc-string and annotations of both the class and each instance method.

    r. Modify the __init__ by making its parameters default and verify by instances.

3. Define a class RLC for a series RLC circuit. Your class must have the appropriate __init__ method.
    i. Add appropriate property methods. (@property and @property.setter)
    ii. Add property method for impedance, phase, and power factor.
    iii. In addition, add an instance method for the current in the circuit with a given input voltage.
    iv. Add __repr__ and __str__ to the class.
    v. Add proper annotation and doc string to every class and instance method.

    a. Define inst_1 and pass two numbers.

    b. Make another instance inst2.

    c. Print inst_1 and inst_2, for this use the print command and pass theinst_1 and inst_2.

    d. Call the __dict__ by the class name.

e. Also pass the class name to the vars built-in function.

f. Call the __dict__ on the object of the class.

g. Pass the class name to help.

h. Print the doc-string and annotations of both the class and each instance method.

i. Modify the __init__ by making its parameters default and verify by instances.

Note: Please make a class for each of the following using the detail procedure from the above tasks.

Math:

4. Square: A four-sided polygon (quadrilateral) with all sides of equal length and all angles 90 degrees.
5. Rectangle: A four-sided polygon where opposite sides are equal in length and all angles are 90 degrees.
6. Triangle: A three-sided polygon with three edges and three vertices. There are various types of triangles (e.g., equilateral, isosceles, scalene).
7. Trapezoid (US) / Trapezium (UK): A four-sided figure with at least one pair of parallel sides.
8. Parallelogram: A four-sided shape with opposite sides that are equal and parallel.
9. Rhombus: A parallelogram where all sides are of equal length, but the angles are not necessarily 90 degrees.
10. These shapes serve as building blocks for more complex forms in geometry and design.

Circuit:

11. Make a class for a circuit with only one resistor.
12. Making a class for the RL circuit.
13. Making a class for the RC circuits.
14. Making a class for RLC parallel circuits.
15. Modify the RLC series for the series resonance case.
16. Modify the RLC parallel for the parallel resonance case.