WORKSHEET 3.2

Student Name: Ravi Shankar Singh UID: 21BCS11619

Branch: CSE Section/Group: 808-B

Semester: 4th Date of Performance:

Subject Name: Programming in Python Lab Subject Code: 21CSP-259

**Aim:**

**Program to implement concept of object-oriented programming such as classes, inheritance and polymorphism**

1. **Write a Python class named Student with two attributes student\_id, student\_name. Add a new attribute student\_class and display the entire attribute and their values of the said class. Now remove the student\_name attribute and display the entire attribute with values**

**Source Code:**

class Student:

student\_id = '11428'

student\_name = 'Aman Raj'

print("Original attributes and their values of the Student class:")

for attr, value in Student.\_\_dict\_\_.items():

if not attr.startswith('\_'):

print(f'{attr} -> {value}')

print("\nAfter adding the student\_class, attributes and their values with the said class:")

Student.student\_class = '807-B'

for attr, value in Student.\_\_dict\_\_.items():

if not attr.startswith('\_'):

print(f'{attr} -> {value}')

print("\nAfter removing the student\_name, attributes and their values from the said class:")

del Student.student\_name

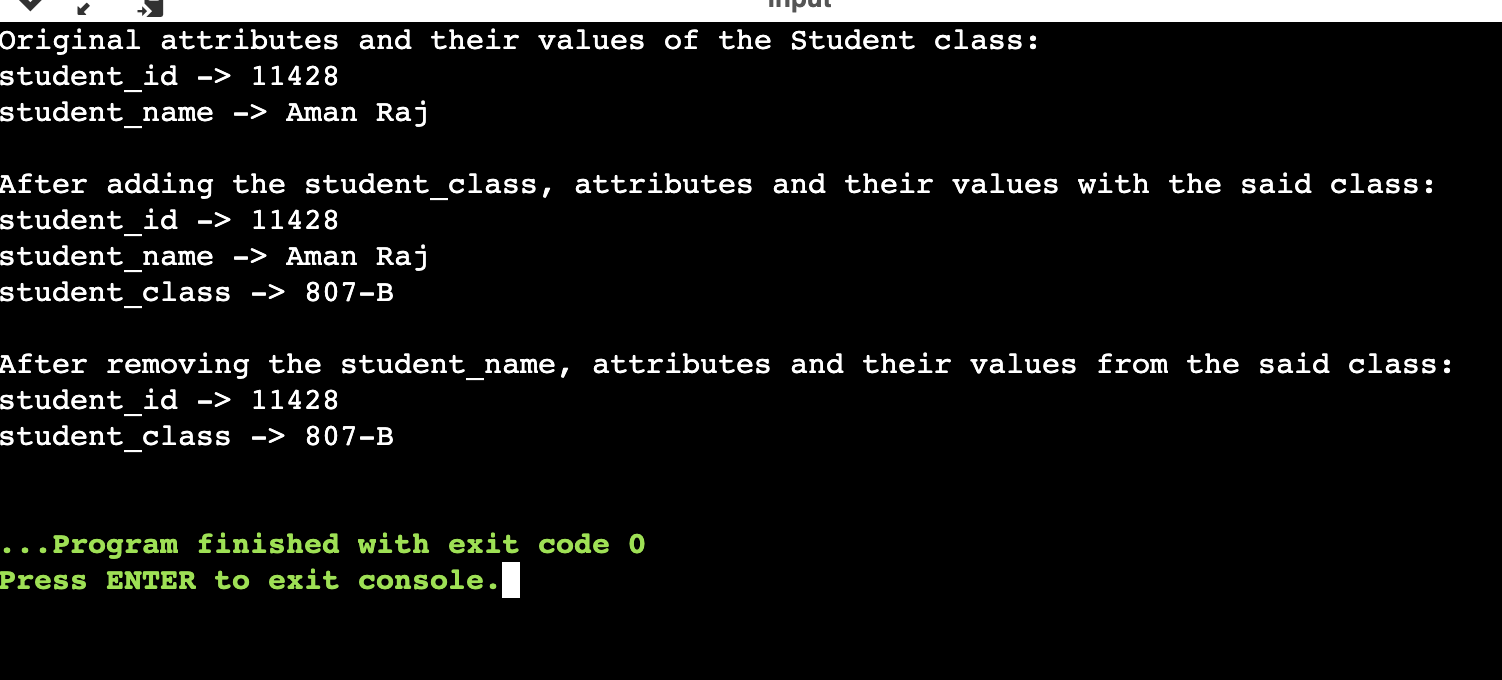
#delattr(Student, 'student\_name')

for attr, value in Student.\_\_dict\_\_.items():

if not attr.startswith('\_'):

print(f'{attr} -> {value}')

**OUTPUT:**

****

1. **Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number.**

**Source Code:**

class py\_solution:

def twoSum(self, nums, target):

lookup = {}

for i, num in enumerate(nums):

if target - num in lookup:

return (lookup[target - num], i )

lookup[num] = i

print("index1=%d, index2=%d" % py\_solution().twoSum((1,2,1,7,9,9,4),5))

# **OUTPUT:**

# 

1. **Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.**

**Source Code:**

class Rectangle():

def \_\_init\_\_(self, l, w):

self.length = l

self.width = w

def rectangle\_area(self):

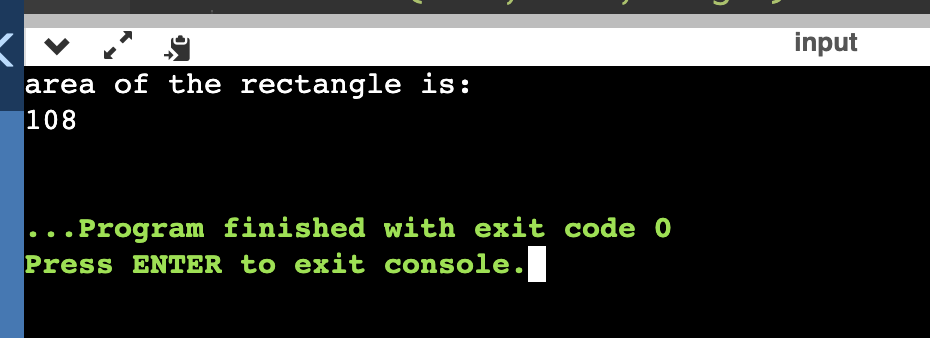
return self.length\*self.width

print("area of the rectangle is:")

newRectangle = Rectangle(6, 18)

print(newRectangle.rectangle\_area())

**OUTPUT:**

****

1. **Write a Python named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle**

**Source Code:**

class Circle():

def \_\_init\_\_(self, r):

self.radius = r

def area(self):

return self.radius\*\*2\*3.14

def perimeter(self):

return 2\*self.radius\*3.14

NewCircle = Circle(8)

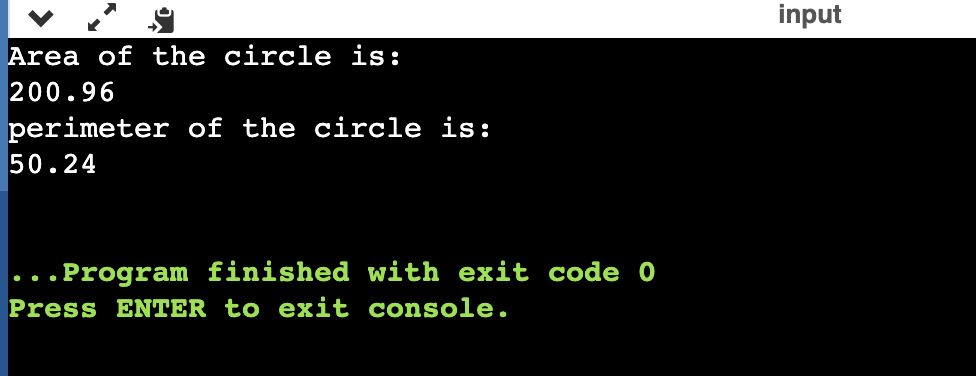
print("Area of the circle is:")

print(NewCircle.area())

print("perimeter of the circle is:")

print(NewCircle.perimeter())

**OUTPUT:**

****

1. **Write a Python program to create two empty classes, Student and Marks. Now create some instances and check whether they are instances of the said classes or not. Also, check whether the said classes are subclasses of the built-in object class or not.**

**Source Code:**

class Student:

pass

class Marks:

pass

student1 = Student()

marks1 = Marks()

print(isinstance(student1, Student))

print(isinstance(marks1, Student))

print(isinstance(marks1, Marks))

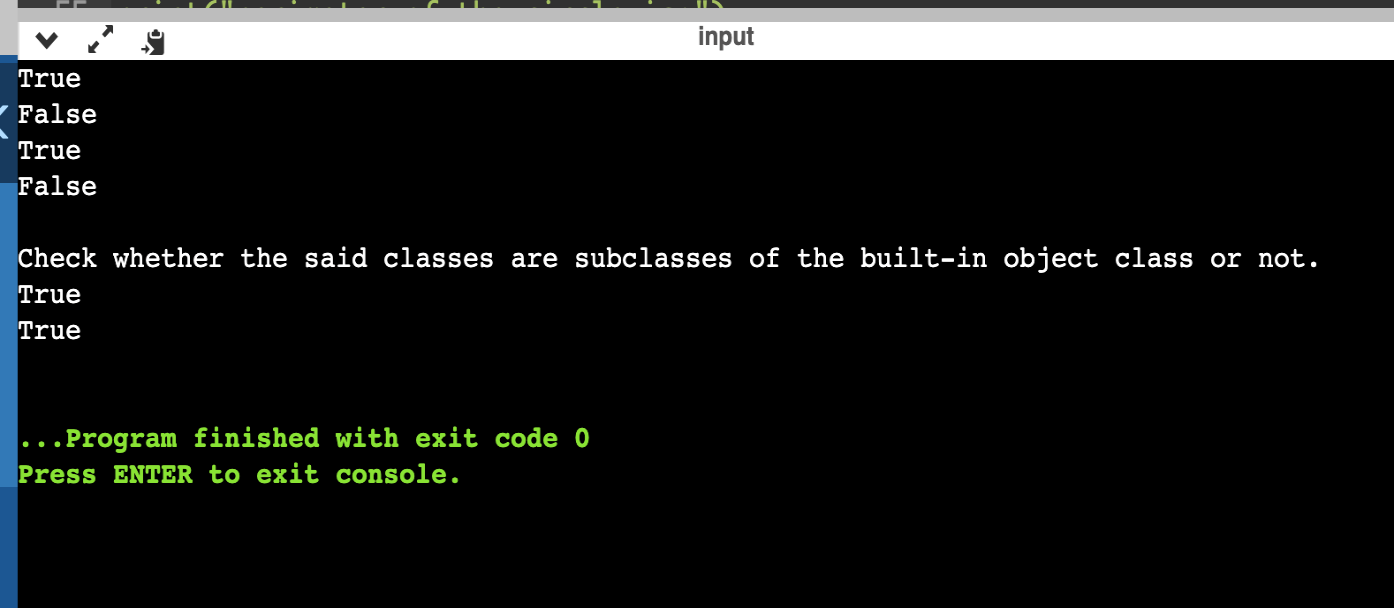
print(isinstance(student1, Marks))

print("\nCheck whether the said classes are subclasses of the built-in object class or not.")

print(issubclass(Student, object))

print(issubclass(Marks, object))

**OUTPUT:**

****