Experiment 3.2

Student Name:Rajiv Paul **UID:20BCS1812**
Branch: CSE **Section/Group:607A**
Semester: 4th **Date of Performance: 22/04/2022**
Subject Name:Programming in Python Lab **Subject Code: 22E-20CSP-259**

1) Aim/Overview of the practical:

Q1. 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

2) Task to be done/ Which logistics used:

To 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 and also to remove the student_name attribute and display the entire attribute with values

3) Algorithm/Flowchart (For programming based labs):

4) Steps for experiment/practical/Code:

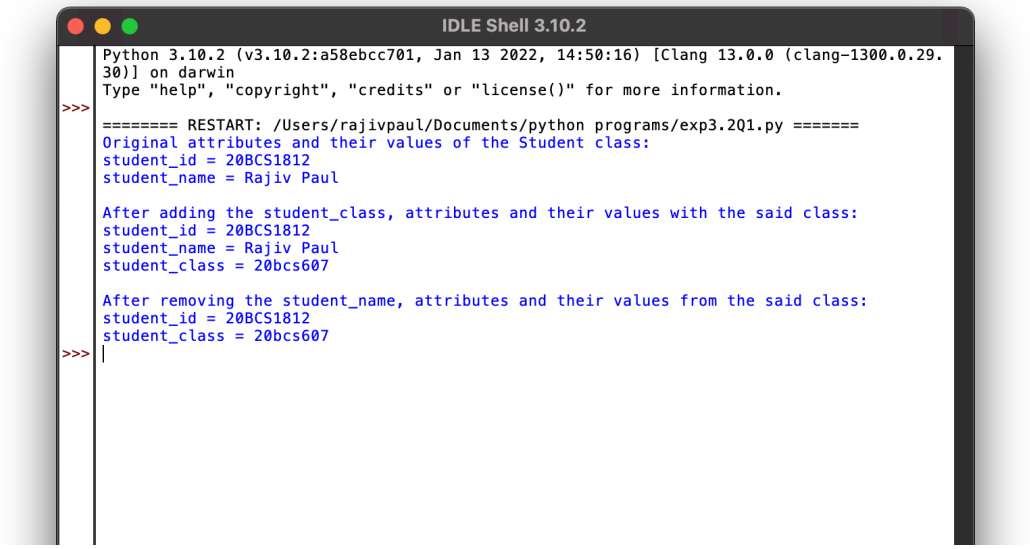
exp3.2Q1.py - /Users/rajivpaul/Documents/python programs/exp3.2Q1.py (3.10.2)

```
class Student:
    student_id = '20BCS1812'
    student_name = 'Rajiv Paul'
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  = '20bcs607'
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

for attr, value in Student.__dict__.items():
    if not attr.startswith('_'):
        print(f'{attr} = {value}')
```

5. Observations/Discussions/ Complexity Analysis:

6. Result/Output/Writing Summary:



1) **Aim/Overview of the practical:**

Q2. 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.

2) Task to be done/ Which logistics used:

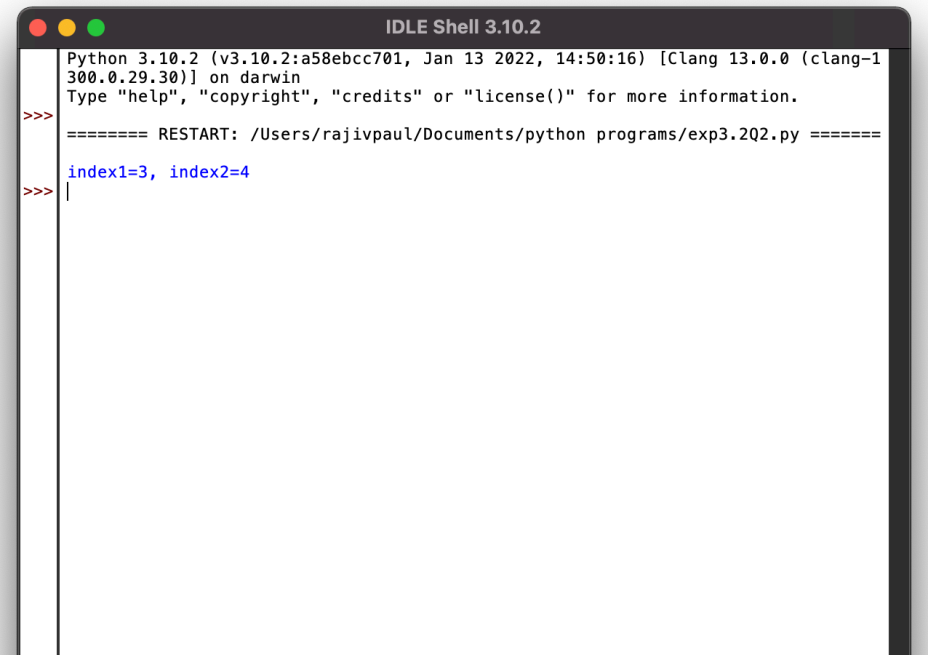
To write a python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number.

3) Algorithm/Flowchart (For programming based labs):

4) Steps for experiment/practical/Code:

5. Observations/Discussions/ Complexity Analysis:

6. Result/Output/Writing Summary:



1) **Aim/Overview of the practical:**

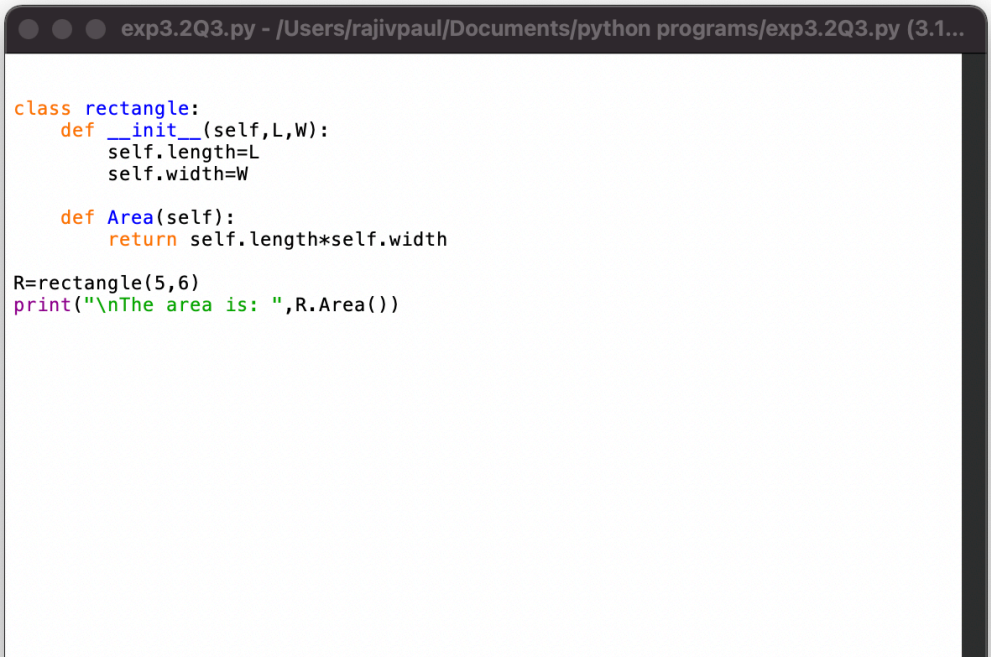
Q3. Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.

2) Task to be done/ Which logistics used:

To write python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.

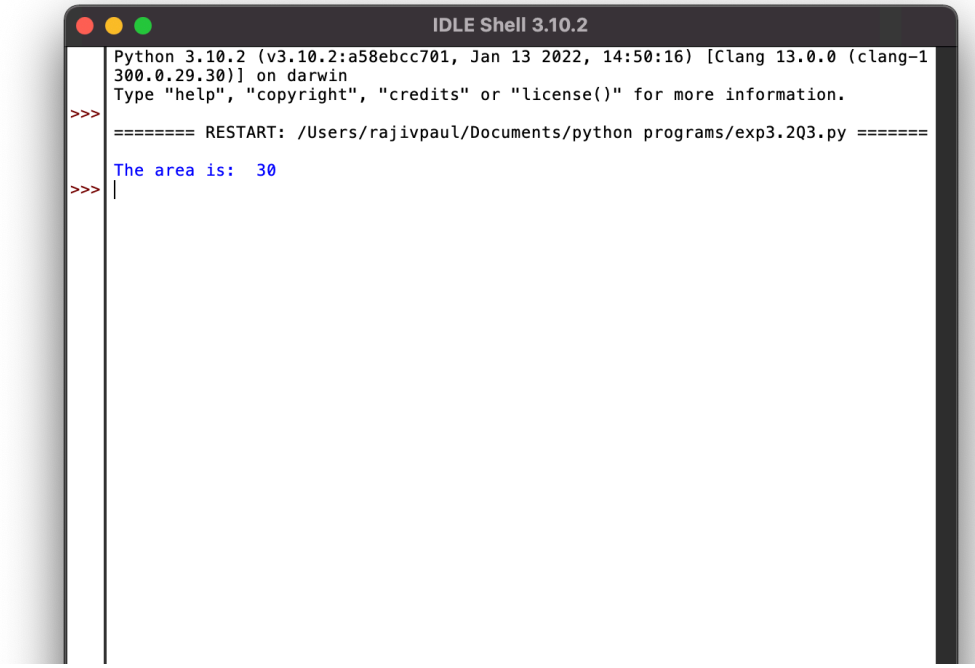
3) Algorithm/Flowchart (For programming based labs):

4) Steps for experiment/practical/Code:



5. Observations/Discussions/ Complexity Analysis:

6. Result/Output/Writing Summary:



1) **Aim/Overview of the practical:**

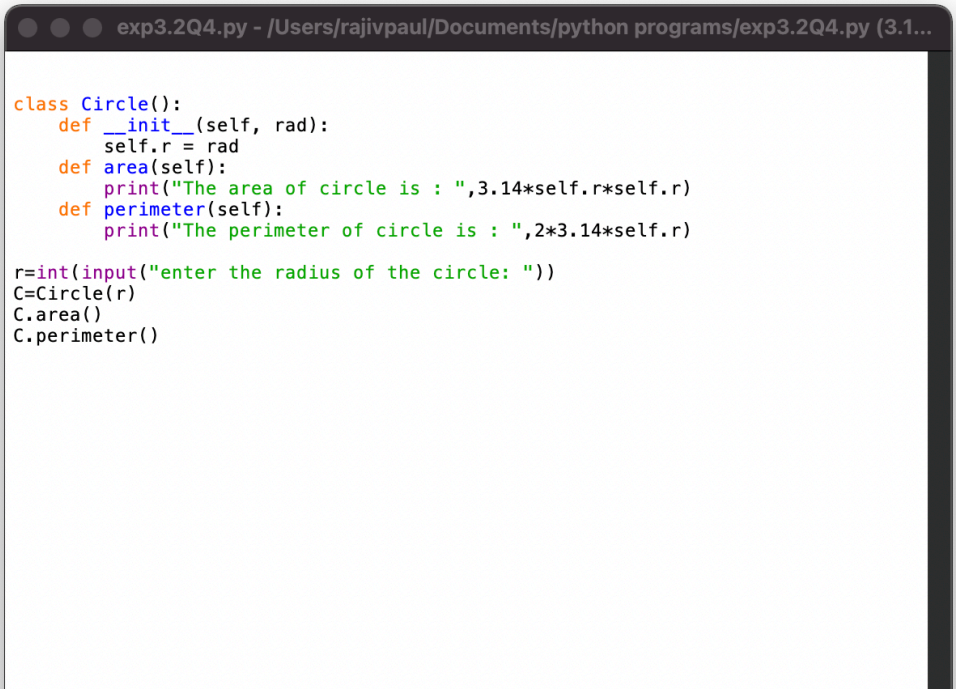
Q4. Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle

2) Task to be done/ Which logistics used:

To write a python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle

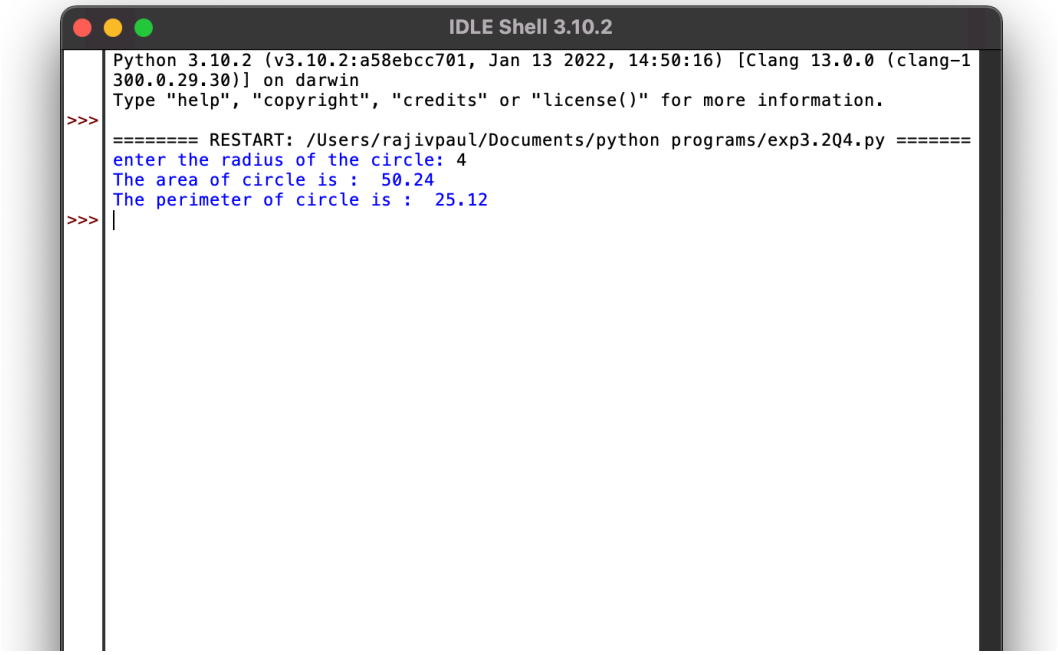
3) Algorithm/Flowchart (For programming based labs):

4) Steps for experiment/practical/Code:



5. Observations/Discussions/ Complexity Analysis:

6. Result/Output/Writing Summary:



1) **Aim/Overview of the practical:**

Q5. Write a Python program to crate 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

2) Task to be done/ Which logistics used:

To write a python program to crate two empty classes, Student and Marks. Now create some instances and check whether they are instances of the said classes or not, also to check whether the said classes are subclasses of the built-in object class or not

3) Algorithm/Flowchart (For programming based labs):

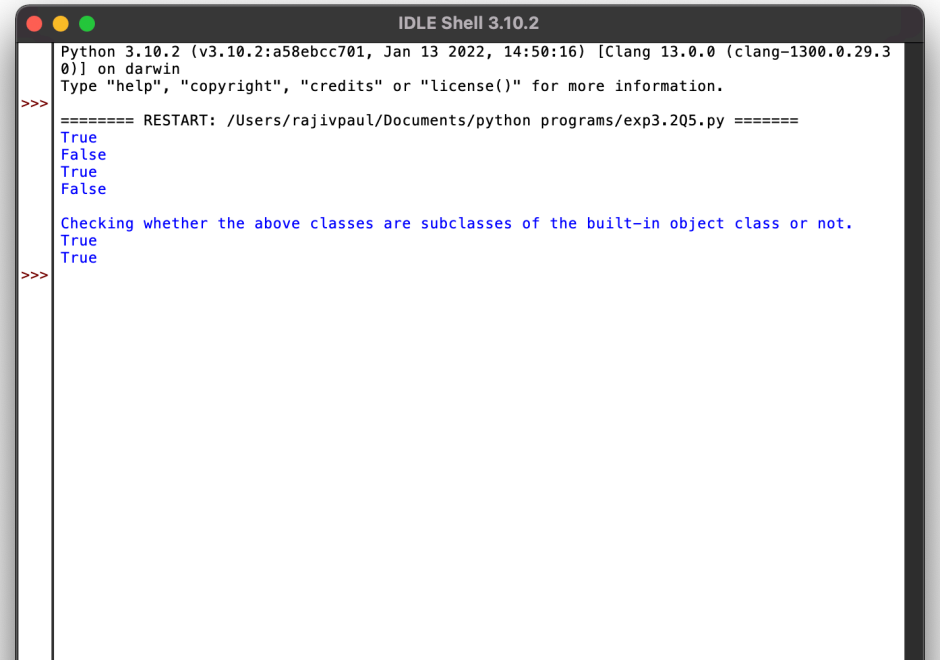
4) Steps for experiment/practical/Code:

exp3.2Q5.py - /Users/rajivpaul/Documents/python programs/exp3.2Q5.py (3.10.2)

```
class Student:
    pass
class Marks:
    pass
students = Student()
marks = Marks()
print(isinstance(students, Student))
print(isinstance(marks, Student))
print(isinstance(marks, Marks))
print(isinstance(students, Marks))
print("\nChecking whether the above classes are subclasses of the built-in object class or not.")
print(issubclass(Student, object))
print(issubclass(Marks, object))
```

5. Observations/Discussions/ Complexity Analysis:

6. Result/Output/Writing Summary:



Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			