

## EXPERIMENT NUMBER – 3.2

STUDENT'S NAME- Sameer Saifi

STUDENT'S UID- 20BCS7619

CLASS AND GROUP- PH20BCS719-A

SUBJECT- PYTHON PROGRAMMING LAB

SEMESTER- 4<sup>TH</sup>

**Question 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

### CODE IN TEXT FORM

```
class Student:
```

```
    student_id = 'V10'
```

```
    student_name = 'James'
```

```
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 = 'V'
```

```
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:")
```

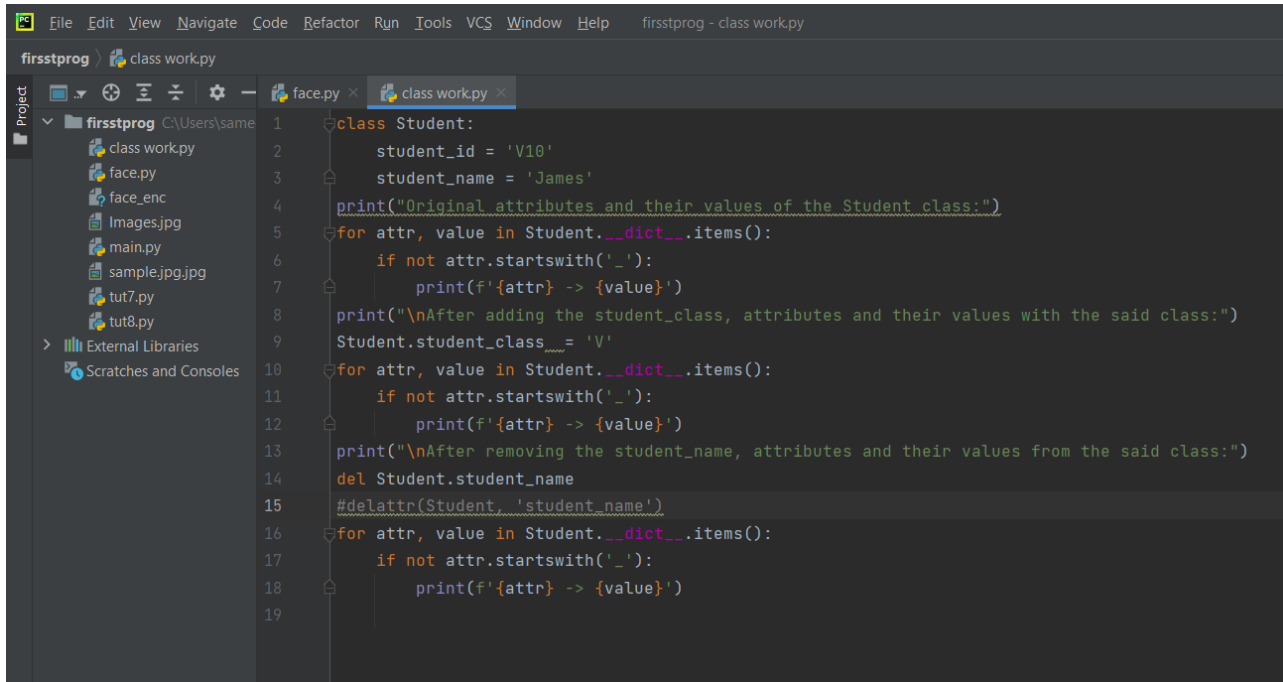
```
del Student.student_name
```

```
#delattr(Student, 'student_name')
```

```
for attr, value in Student.__dict__.items():
```

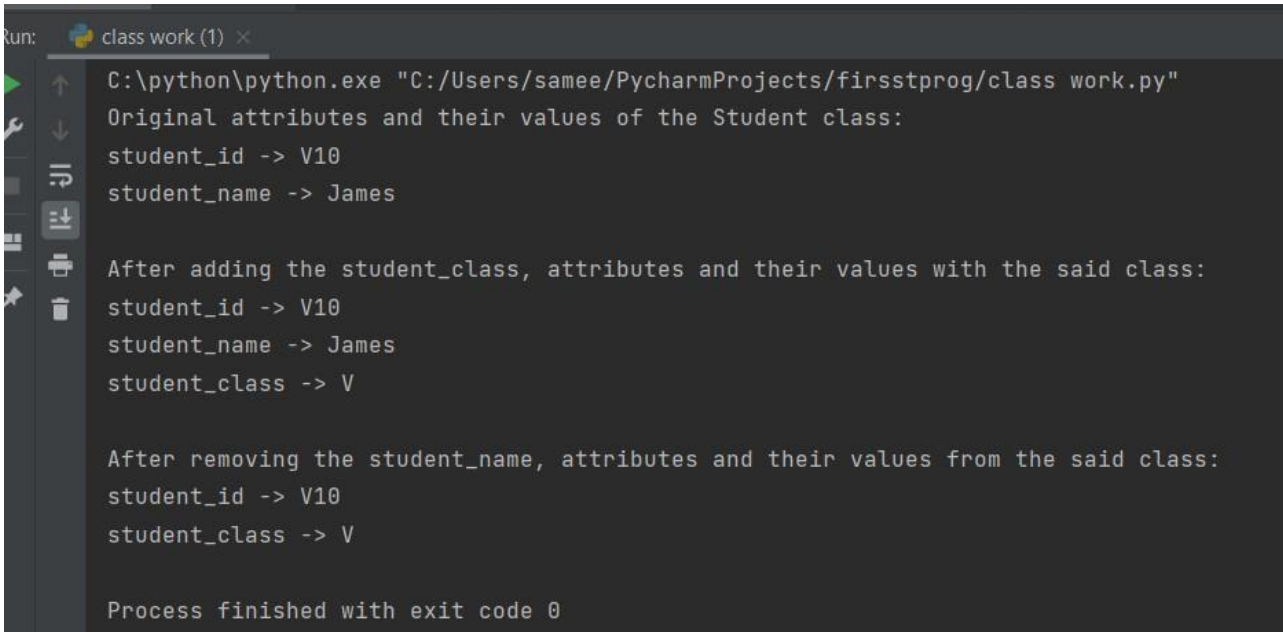
```
if not attr.startswith('_'):  
    print(f'{attr} -> {value}')
```

## CODE IN COMPILER



```
1 class Student:  
2     student_id = 'V10'  
3     student_name = 'James'  
4     print("Original attributes and their values of the Student class:")  
5     for attr, value in Student.__dict__.items():  
6         if not attr.startswith('_'):  
7             print(f'{attr} -> {value}')8     print("\nAfter adding the student_class, attributes and their values with the said class:")  
9     Student.student_class = 'V'  
10    for attr, value in Student.__dict__.items():  
11        if not attr.startswith('_'):  
12            print(f'{attr} -> {value}')13    print("\nAfter removing the student_name, attributes and their values from the said class:")  
14    del Student.student_name  
15    #delattr(Student, 'student_name')  
16    for attr, value in Student.__dict__.items():  
17        if not attr.startswith('_'):  
18            print(f'{attr} -> {value}')19
```

## OUTPUT-



```
Run: class work (1) x  
C:\python\python.exe "C:/Users/samee/PycharmProjects/firsstprog/class work.py"  
Original attributes and their values of the Student class:  
student_id -> V10  
student_name -> James  
  
After adding the student_class, attributes and their values with the said class:  
student_id -> V10  
student_name -> James  
student_class -> V  
  
After removing the student_name, attributes and their values from the said class:  
student_id -> V10  
student_class -> V  
  
Process finished with exit code 0
```

**Question 2:** 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.

**CODE IN TEXT FORM**

```
class pair:

    def twoSum(self, nums, target):

        lookup = {}

        for i, num in enumerate(nums):

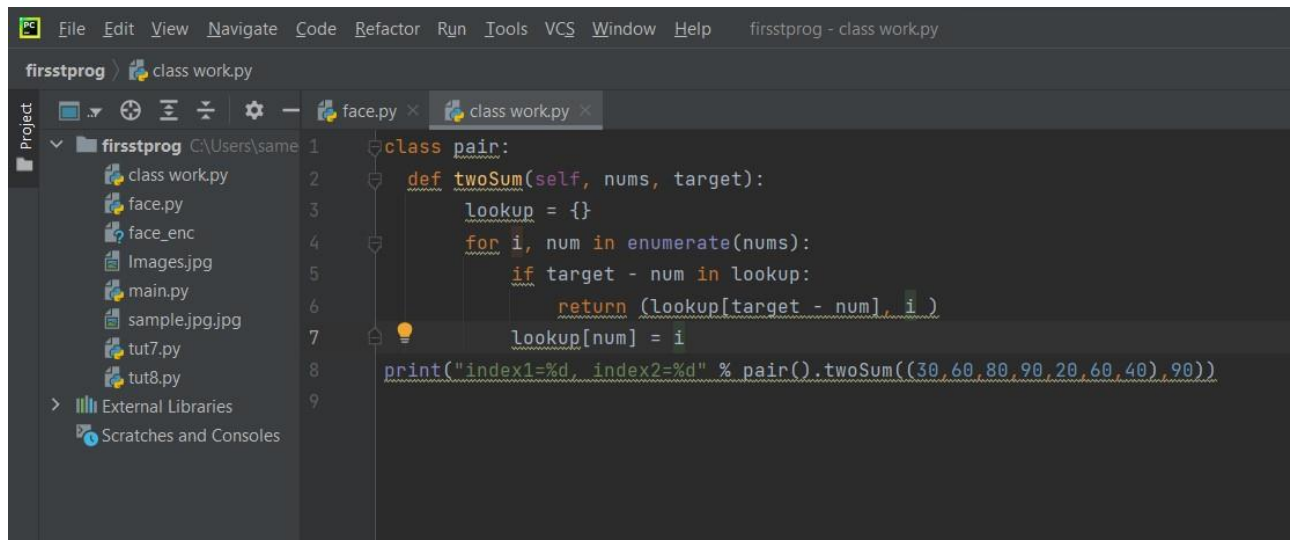
            if target - num in lookup:

                return (lookup[target - num], i )

            lookup[num] = i

print("index1=%d, index2=%d" %
pair().twoSum((30,60,80,90,20,60,40),90))
```

**CODE IN COMPILER –**

A screenshot of a code editor window titled 'firststprog - class work.py'. The editor shows the same Python code as the 'CODE IN TEXT FORM' block. The code defines a class 'pair' with a method 'twoSum' that uses a dictionary 'lookup' to find a pair of indices whose sum equals a target. The code is executed, and the output 'index1=0, index2=5' is printed. The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help) and a project explorer on the left showing the file structure of the 'firststprog' project.

## OUTPUT –

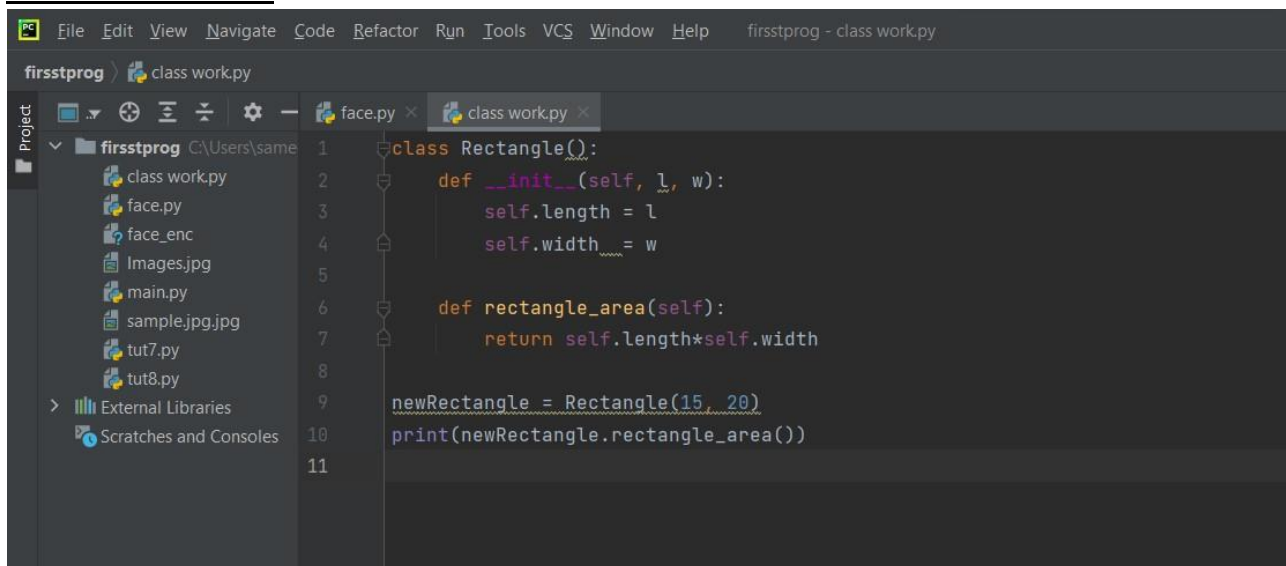
```
class work (1) ×  
C:\python\python.exe "C:/Users/samee/PycharmProjects/firsstprog/class work.py"  
index1=0, index2=1  
  
Process finished with exit code 0  
|
```

**Question 3:** Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle

**CODE IN TEXT FORM –**

```
class Rectangle():  
    def __init__(self, l, w):  
        self.length = l  
        self.width = w  
  
    def rectangle_area(self):  
        return self.length*self.width  
  
newRectangle = Rectangle(12, 10)  
print(newRectangle.rectangle_area())
```

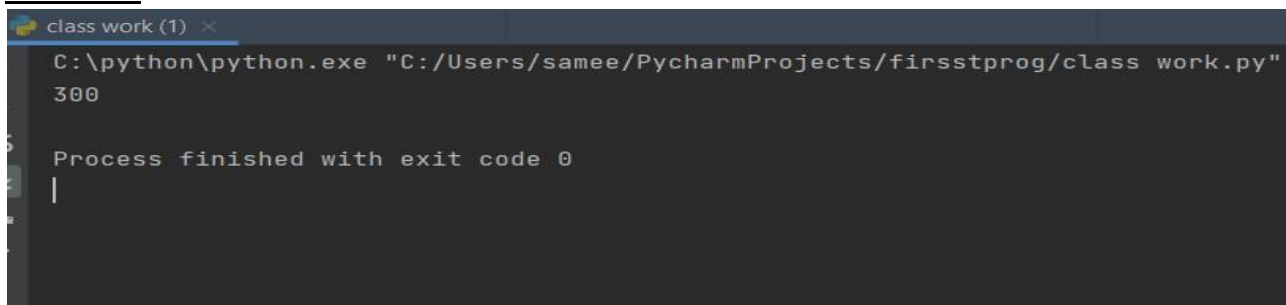
**CODE IN COMPILER-**



The screenshot shows the PyCharm IDE with a project named 'firsstprog'. The file explorer on the left shows the project structure. The main editor window displays the code for 'class work.py'. The code defines a 'Rectangle' class with an '\_\_init\_\_' method and a 'rectangle\_area' method. It then creates an instance 'newRectangle' with length 15 and width 20, and prints the result of 'rectangle\_area()'. The output is 300.

```
1 class Rectangle():  
2     def __init__(self, l, w):  
3         self.length = l  
4         self.width = w  
5  
6     def rectangle_area(self):  
7         return self.length*self.width  
8  
9 newRectangle = Rectangle(15, 20)  
10 print(newRectangle.rectangle_area())  
11
```

**OUTPUT –**



The screenshot shows the output console of the PyCharm IDE. It displays the command used to run the code: 'C:\python\python.exe "C:/Users/samee/PycharmProjects/firsstprog/class work.py"'. The output is '300'. Below this, it says 'Process finished with exit code 0'.

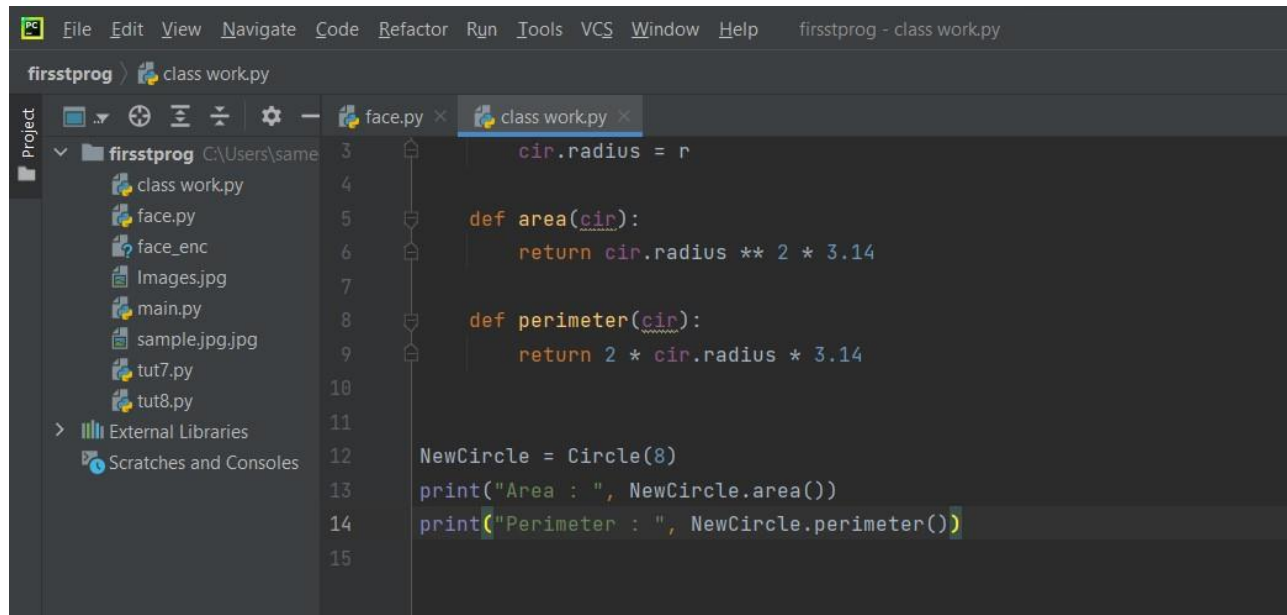
```
class work (1) ×  
C:\python\python.exe "C:/Users/samee/PycharmProjects/firsstprog/class work.py"  
300  
  
Process finished with exit code 0  
|
```

**Question 4:** Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle

**CODE IN TEXT –**

```
class Circle():  
    def __init__(cir, r):  
        cir.radius = r  
  
    def area(cir):  
        return cir.radius**2*3.14  
  
    def perimeter(cir):  
        return 2*cir.radius*3.14  
  
NewCircle = Circle(8)  
print("Area :",NewCircle.area())  
print("Perimeter :",NewCircle.perimeter())
```

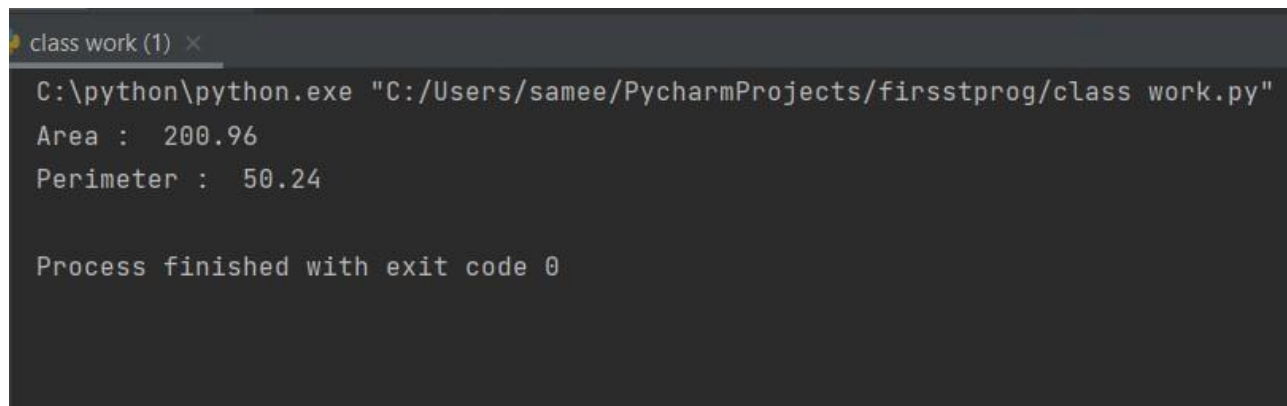
## CODE IN COMPILER –



The screenshot shows a code editor with a project named 'firsstprog'. The file 'class work.py' is open, displaying the following Python code:

```
1  class Circle:
2      def __init__(self, r):
3          self.radius = r
4
5      def area(self):
6          return self.radius ** 2 * 3.14
7
8      def perimeter(self):
9          return 2 * self.radius * 3.14
10
11
12  NewCircle = Circle(8)
13  print("Area : ", NewCircle.area())
14  print("Perimeter : ", NewCircle.perimeter())
15
```

## OUTPUT –



The screenshot shows a terminal window titled 'class work (1)'. It displays the command to run the Python file and the resulting output:

```
C:\python\python.exe "C:/Users/samee/PycharmProjects/firsstprog/class work.py"
Area : 200.96
Perimeter : 50.24

Process finished with exit code 0
```

**Question 5:** 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

**CODE IN TEXT –**

```
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))
```

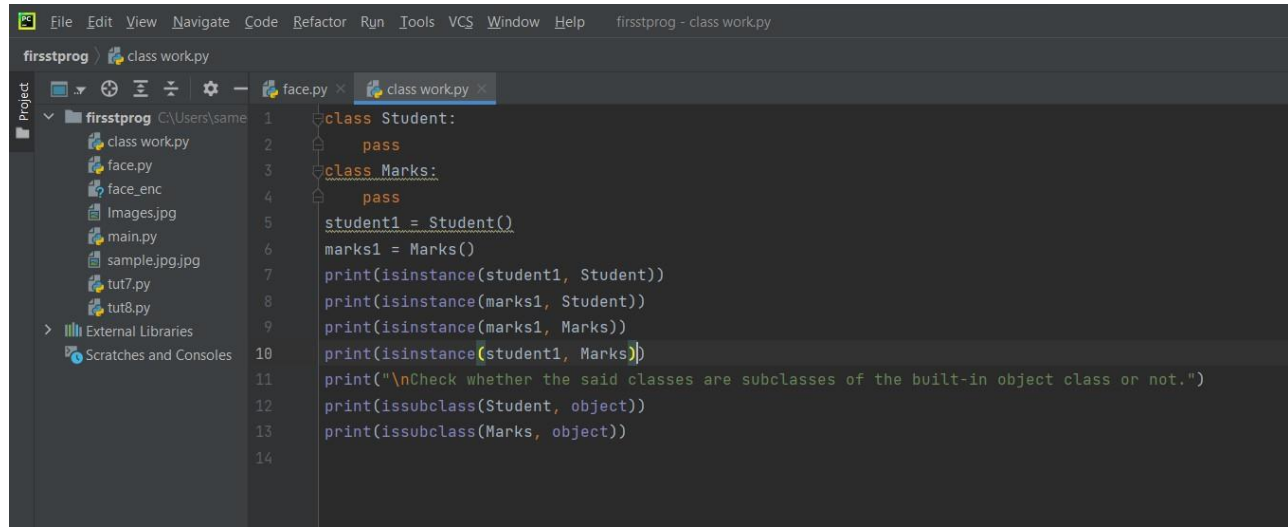
```
print("\nCheck whether the said classes are subclasses of the  
built-in object class or not.")
```

```
print(issubclass(Student, object))
```

```
print(issubclass(Marks, object))
```



## CODE IN COMPILER –

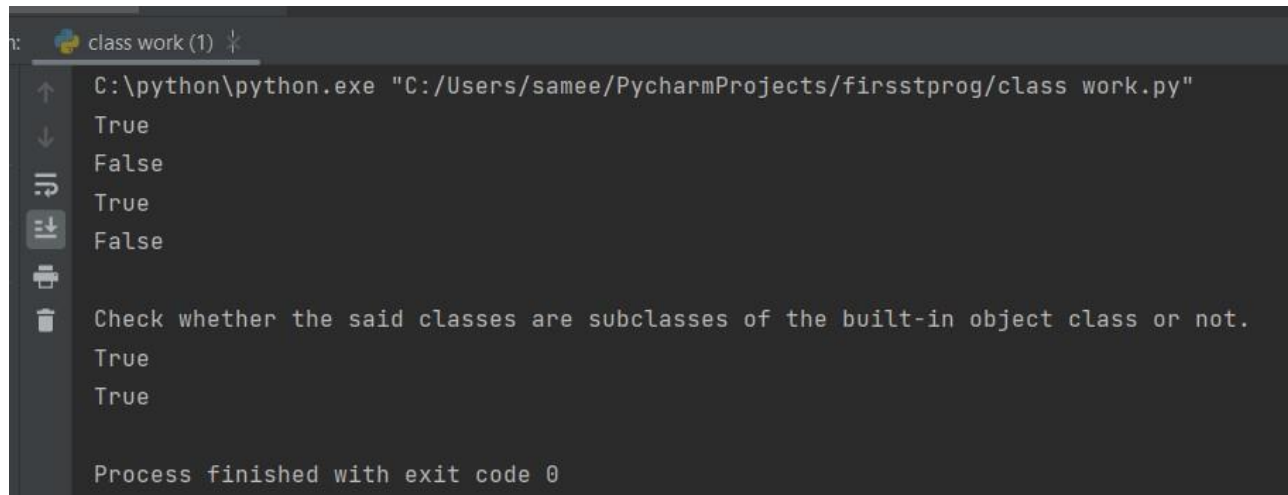


```
File Edit View Navigate Code Refactor Run Tools VCS Window Help firstprog - class work.py

firstprog > class work.py
class work.py x
face.py x

1 class Student:
2     pass
3 class Marks:
4     pass
5 student1 = Student()
6 marks1 = Marks()
7 print(isinstance(student1, Student))
8 print(isinstance(marks1, Student))
9 print(isinstance(marks1, Marks))
10 print(isinstance(student1, Marks))
11 print("\nCheck whether the said classes are subclasses of the built-in object class or not.")
12 print(issubclass(Student, object))
13 print(issubclass(Marks, object))
14
```

## OUTPUT –



```
Run: class work (1)
C:\python\python.exe "C:/Users/samee/PycharmProjects/firsstprog/class work.py"
True
False
True
False

Check whether the said classes are subclasses of the built-in object class or not.
True
True

Process finished with exit code 0
```