Worksheet 3.2

Student Name: Sahil Kaundal UID: 21BCS8197

Branch: CSE (Lateral Entry) Section/Group:20BCS-807\_B Semester: 4th Date of Performance: 21/04/2022

Subject Name: Programming in Python Lab Subject Code: 20CSP-259

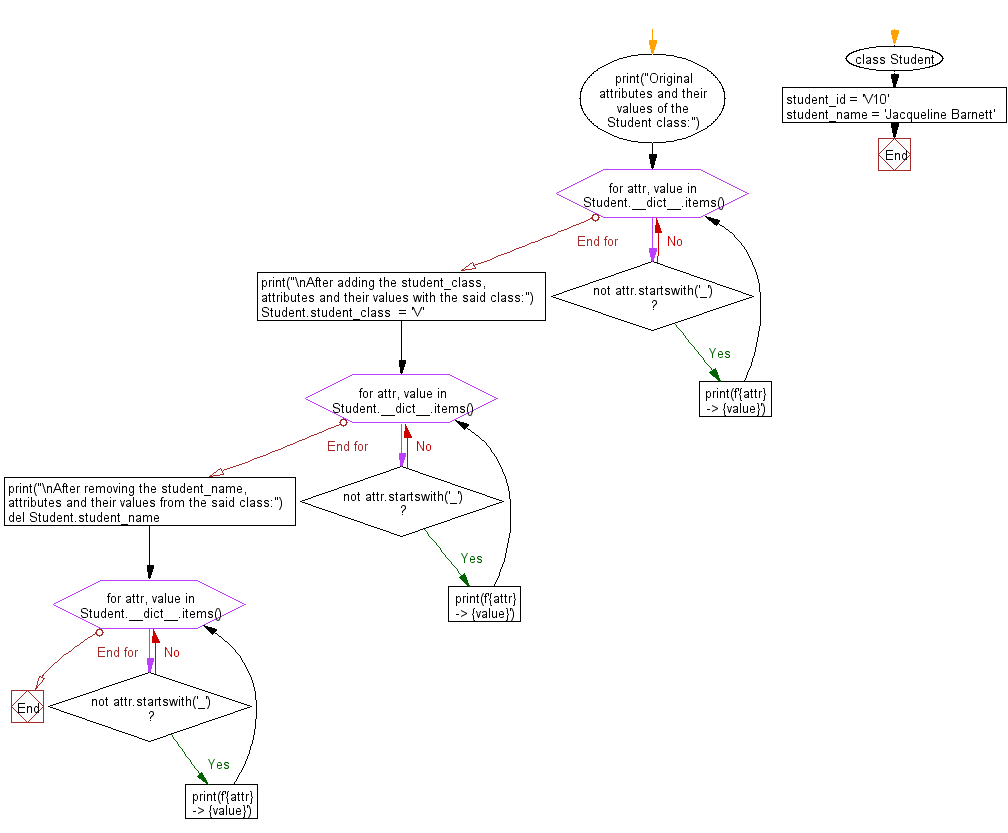
1. Aim/Overview of the practical:

* 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.
* 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.
* Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.
* Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.
* 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.

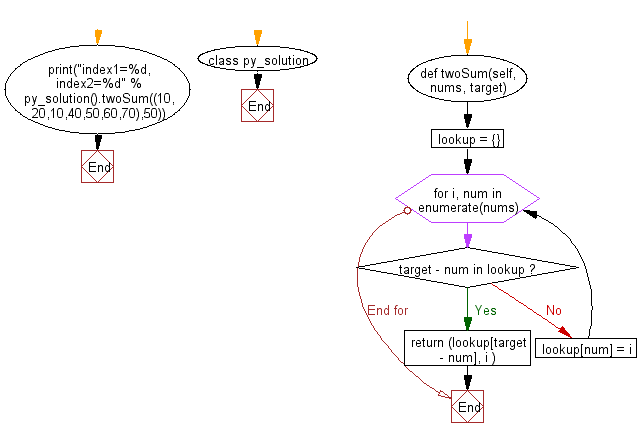
1. Task to be done/ Which logistics used:



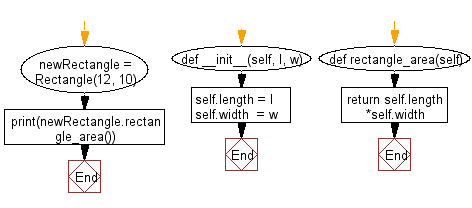
* Write a Python class named Student with two attributes student\_id, student\_name. Add a new attribute student\_class and display the entire attribute and their values of the said class. Now remove the student\_name attribute and display the entire attribute with values.



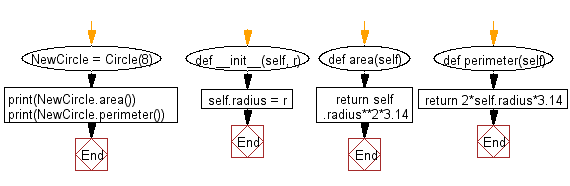
* Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number.



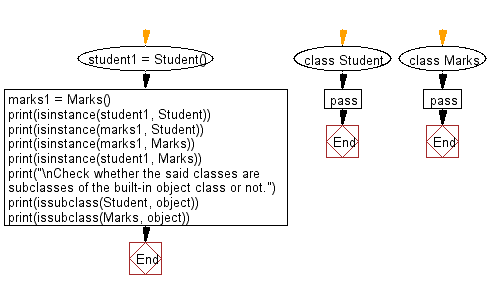
* Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.



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



* 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.



1. Steps for experiment/practical/Code:

* 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.

class Student:

  student\_id = '21BCS8197'

  student\_name = 'Sahil'

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'

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}')

* 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.

class sahil:

 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" % sahil().twoSum((10,20,10,40,50,60,70),50))

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

class Rectangle():

  def \_\_init\_\_(self, l, w):

    self.length = l

    self.width = w

  def rectangle\_area(self):

    return self.length\*self.width

newRectangle = Rectangle(17, 12)

print(newRectangle.rectangle\_area())

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

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(NewCircle.area())

print(NewCircle.perimeter())

* 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.

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))

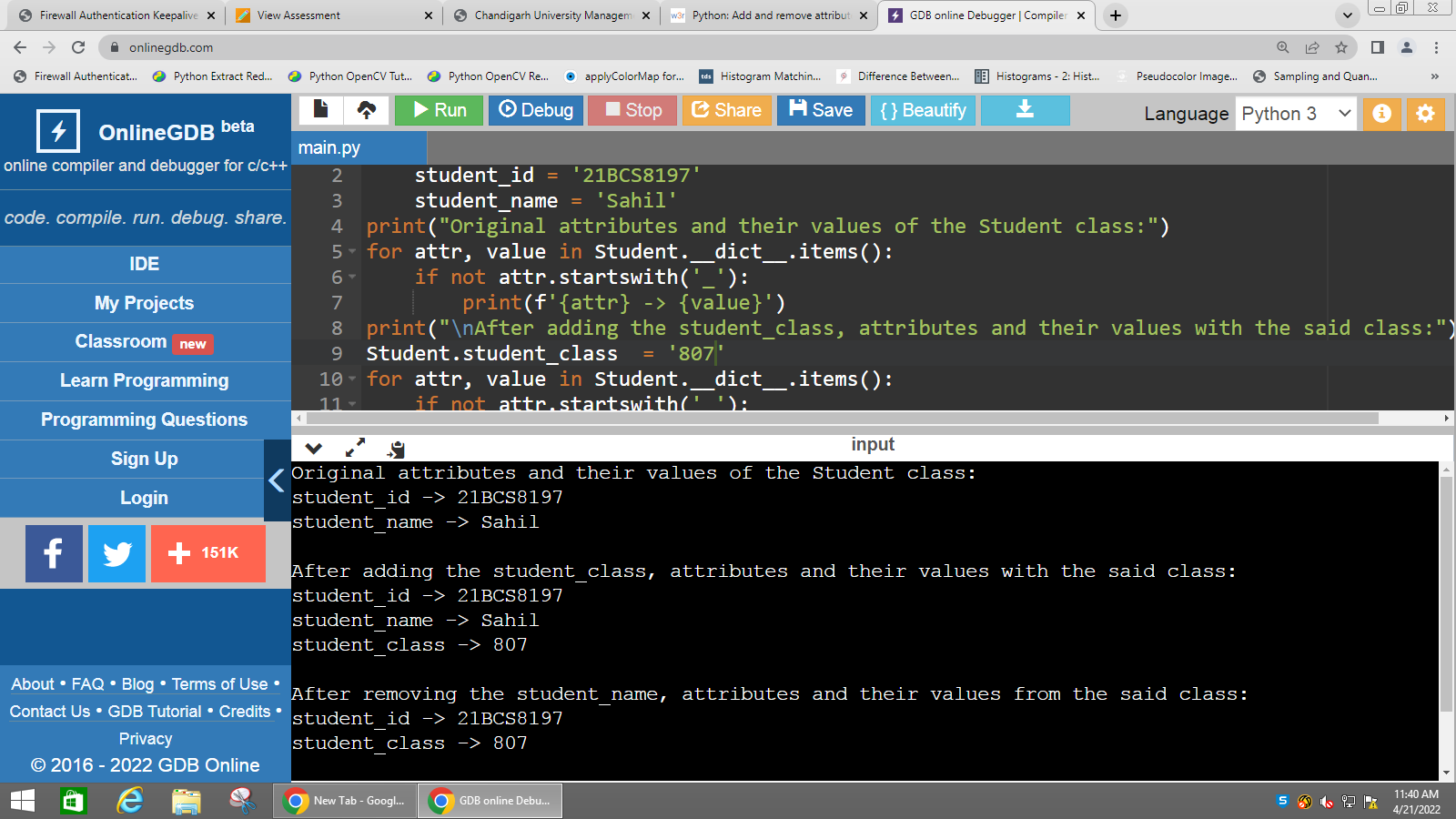
print("\nCheck whether the said classes are subclasses of the built-in object class or not.")

print(issubclass(Student, object))

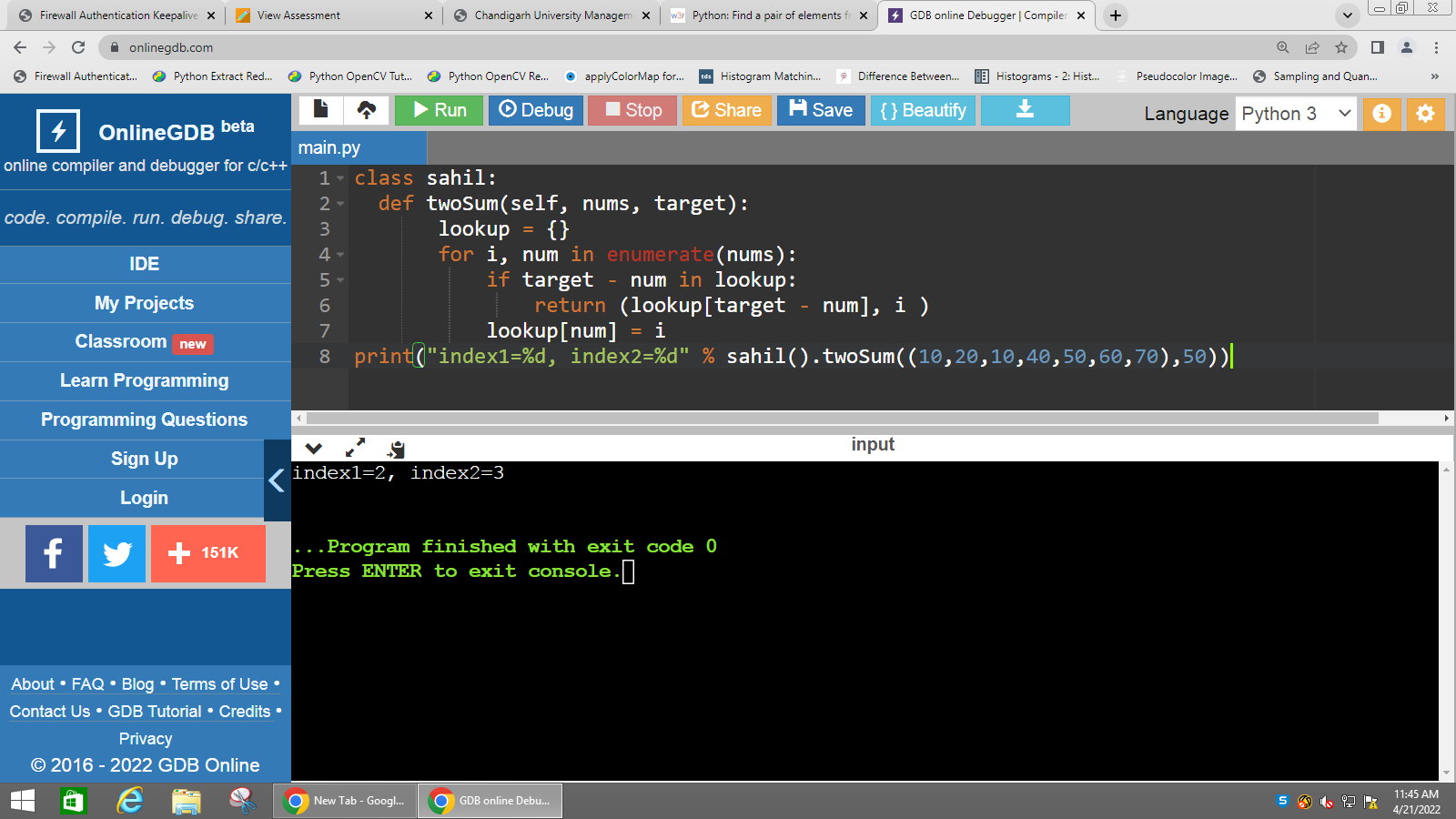
print(issubclass(Marks, object))

1. Result/Output/Writing Summary:

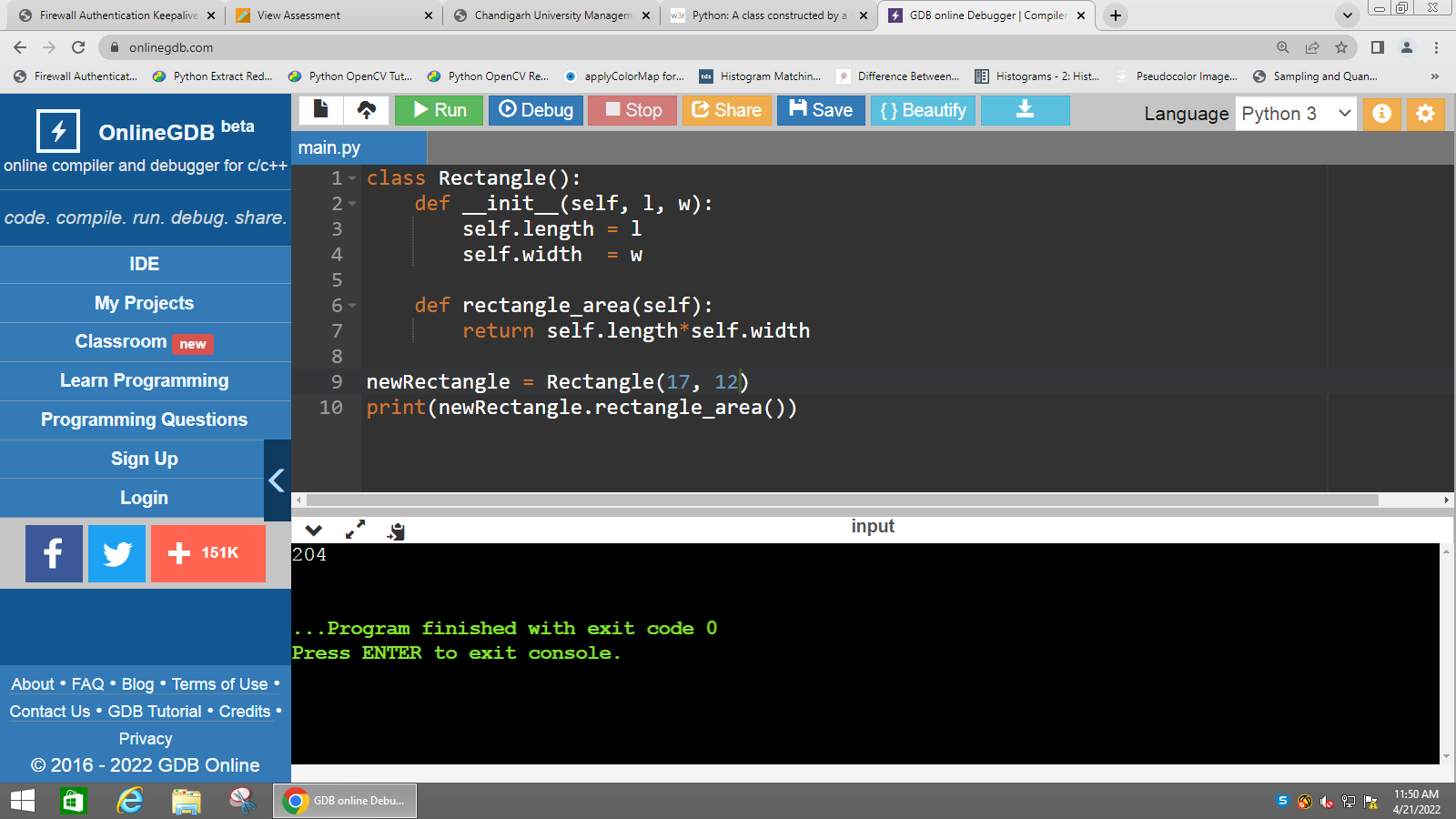
* Write a Python class named Student with two attributes student\_id, student\_name. Add a new attribute student\_class and display the entire attribute and their values of the said class. Now remove the student\_name attribute and display the entire attribute with values.



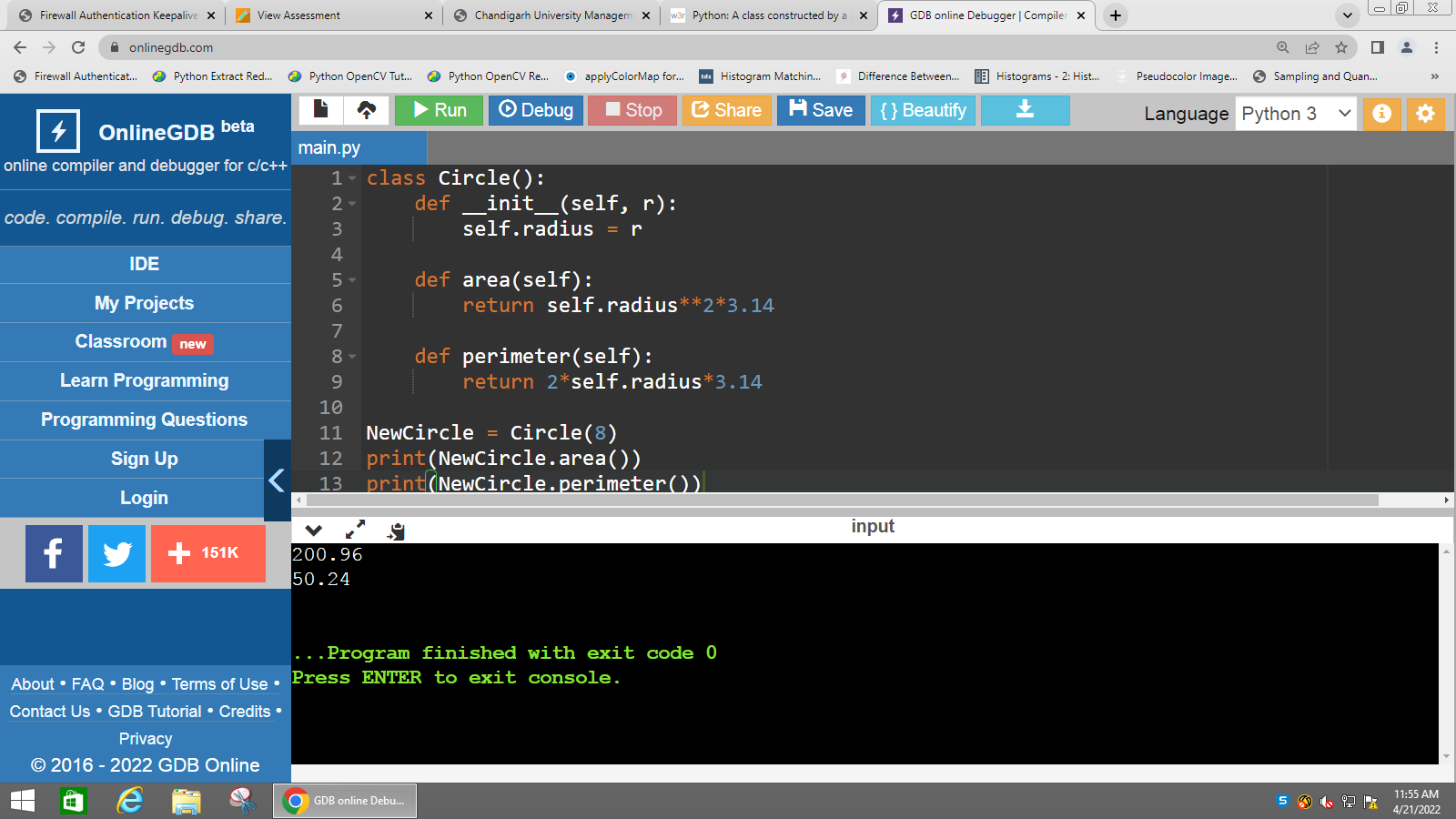
* Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number.



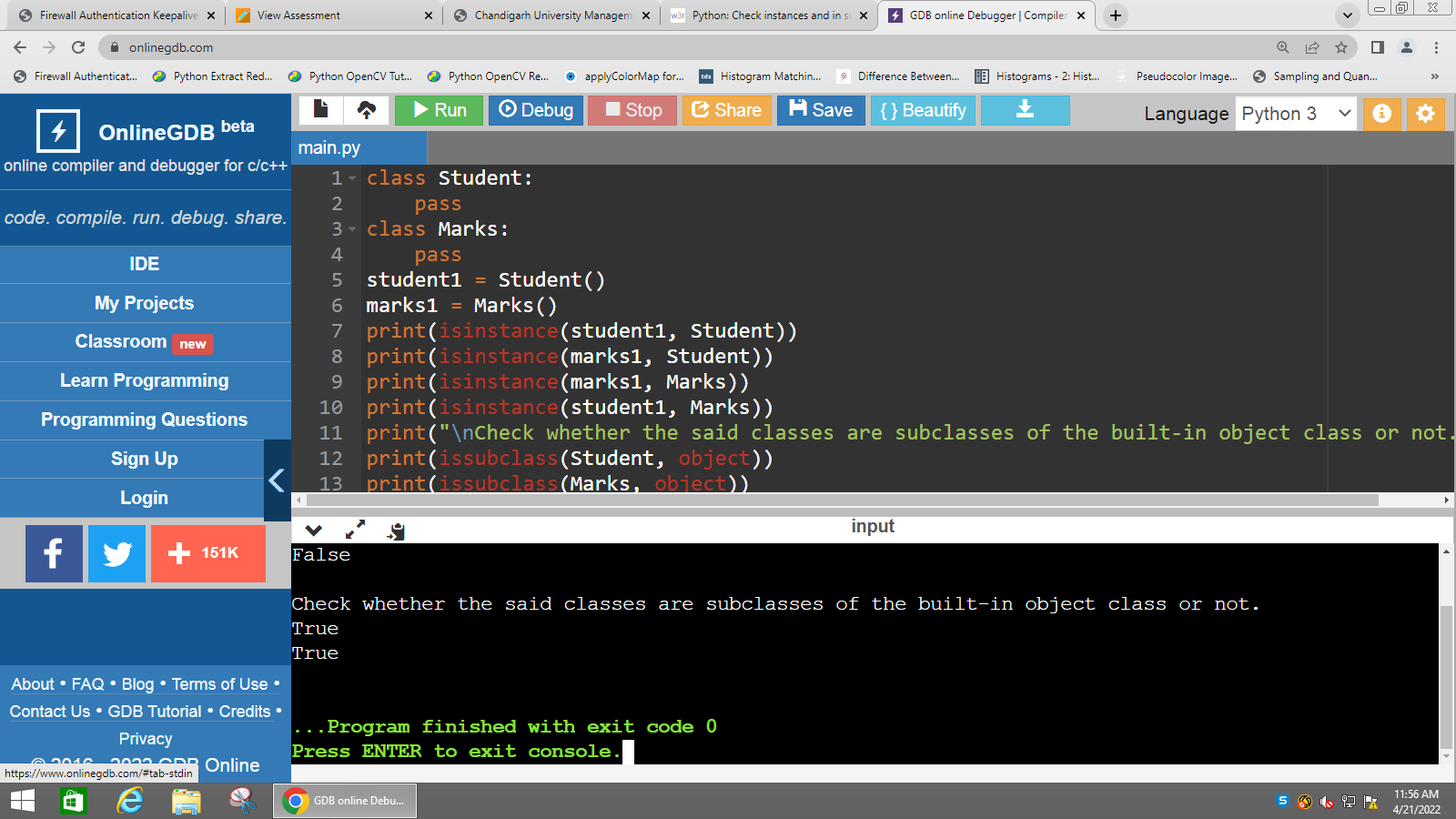
* Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.



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



* 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.



Learning outcomes (What I have learnt):

* 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.
* Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number.
* Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.
* Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.
* 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.

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. |  |  |  |