UNIT – 3

Experiment – 3.2

Name: Krish Monga	**UID**: 21BCS7600
Branch: CSE	**Section**: 702-A
Semester: 4	**Date of Performance**: 08-05-2023
Subject: Programming in Python	**Subject Code**: 21CSP-259

Aim:- Program to implement concepts of Object Oriented Programming such as classes, Inheritance and Polymorphism.

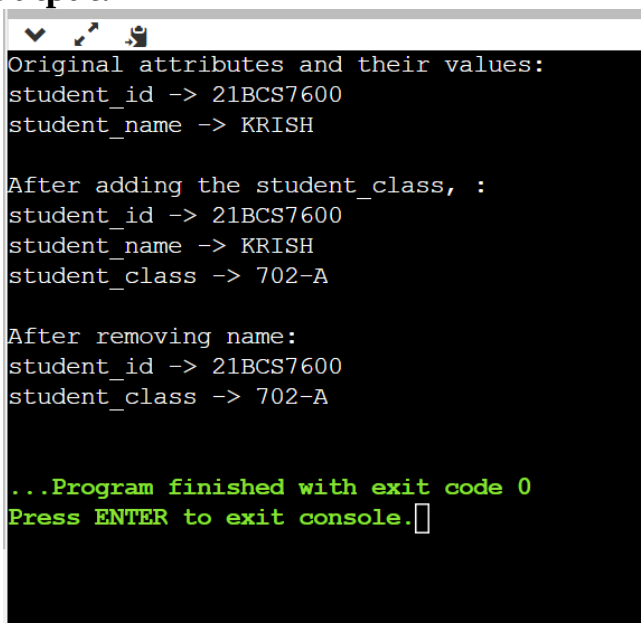
Program 3.2.1:

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.

Program Code:

```
class Student:
student_id = '21BCS7600'
student_name = 'KRISH'
   print("Original attributes and their values:")
      for attr, value in Student.__dict__.items():
     if not attr.startswith('_'):
  print(f'{attr} -> {value}')
  print("\nAfter adding the student_class, :")
Student.student_class = '702-A'
    for attr, value in Student.__dict__.items():
     if not attr.startswith('_'):
   print(f'{attr} -> {value}')
  print("\nAfter removing name:")
  del Student.student_name
    for attr, value in Student.__dict__.items():
   if not attr.startswith('_'):
   print(f'{attr} -> {value}')
```

Output:



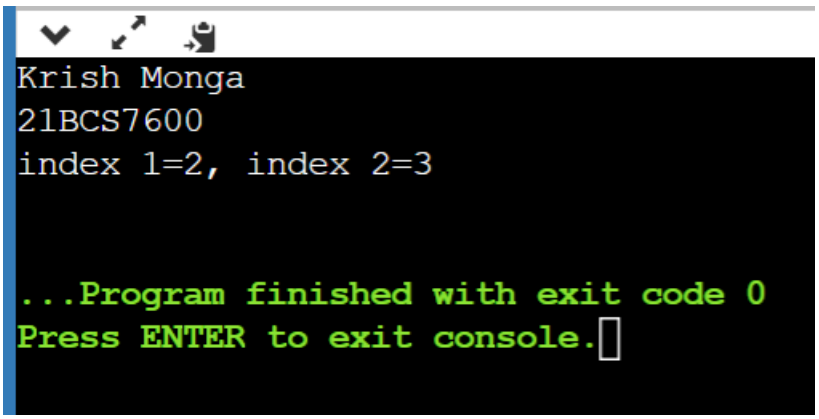
Program 3.2.2:

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.

Program Code:

```
 print("Krish Monga")
 print("21BCS7600")
class pairs:
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("index 1=%d, index 2=%d" %
   pairs().twoSum((10,20,10,40,50,60,70),50))
```

Output:



Program 3.2.3:

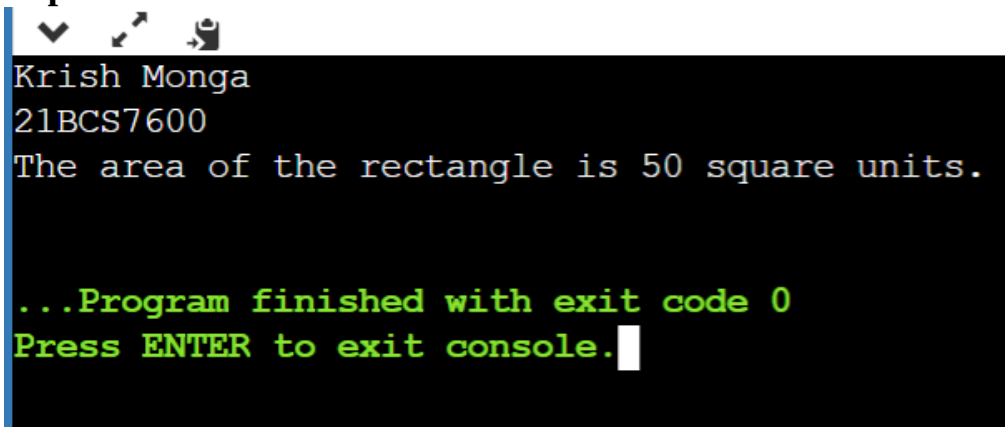
Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.

Program Code:

```
print("Krish Monga")
print("21BCS7600")
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def area(self):
        return self.length * self.width
rect = Rectangle(5, 10)
print(f"The area of the rectangle is {rect.area()} square units.")
```

Output:



Program 3.2.4:

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

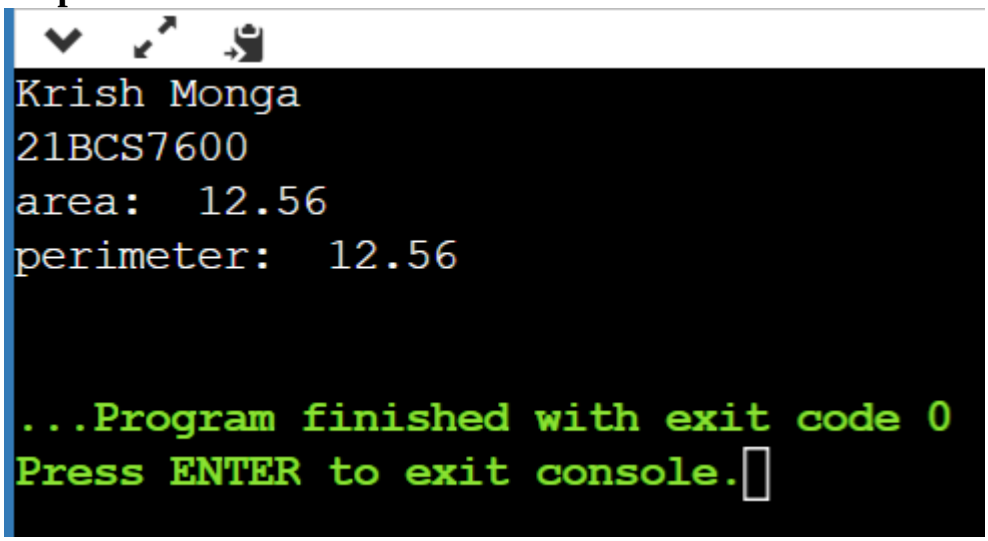
Program Code:

```
print("Krish Monga")
print("21BCS7600")
class Circle:
    def __init__(self,radius):
        self.radius= radius

    def area(self):
        return 3.14*(self.radius**2)

    def perimeter(self):
        return 2*3.14*(self.radius)
  circle=Circle(2)
  print("area: ",circle.area())
  print("perimeter: ", circle.perimeter())
```

Output:



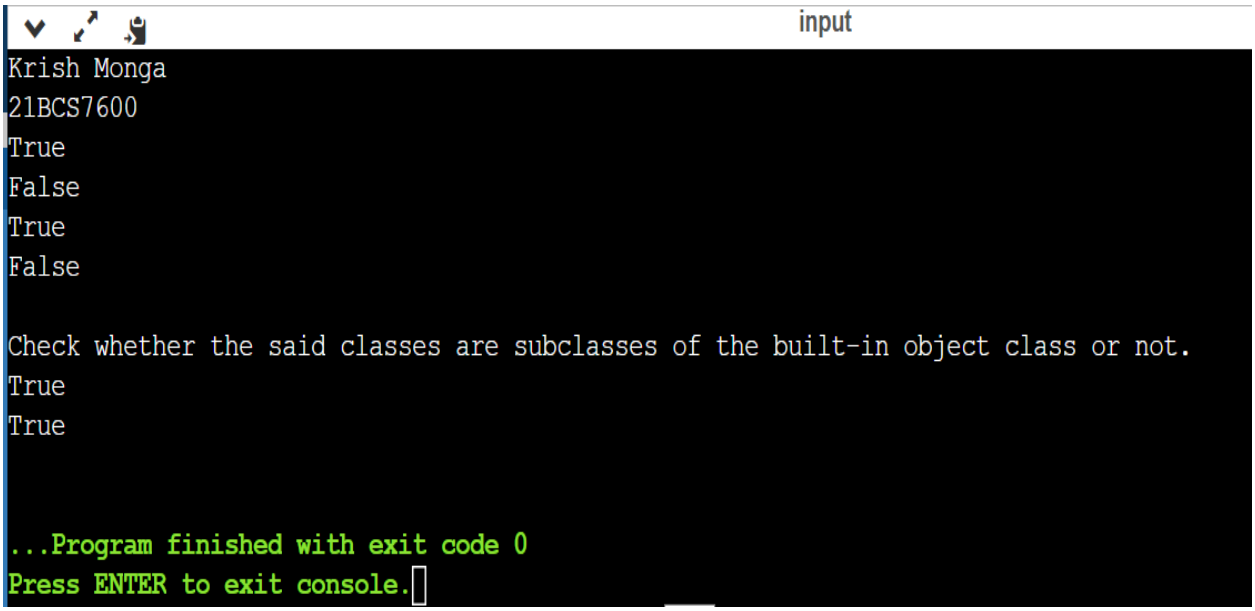
Program 3.2.5:

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.

Program Code:

```
print("Krish Monga")
print("21BCS7600")
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:



Learning Outcomes:

- ➢ Basic concepts of OOPs in Python.
- ➢ Implementation of classes and objects.
- ➢ Implementation of Polymorphism, Inheritance.