IC 152

Computing & Data Science

Lab 9

Mohit Verma

B20215

Instructor- Dr.Padmanabhan Rajan

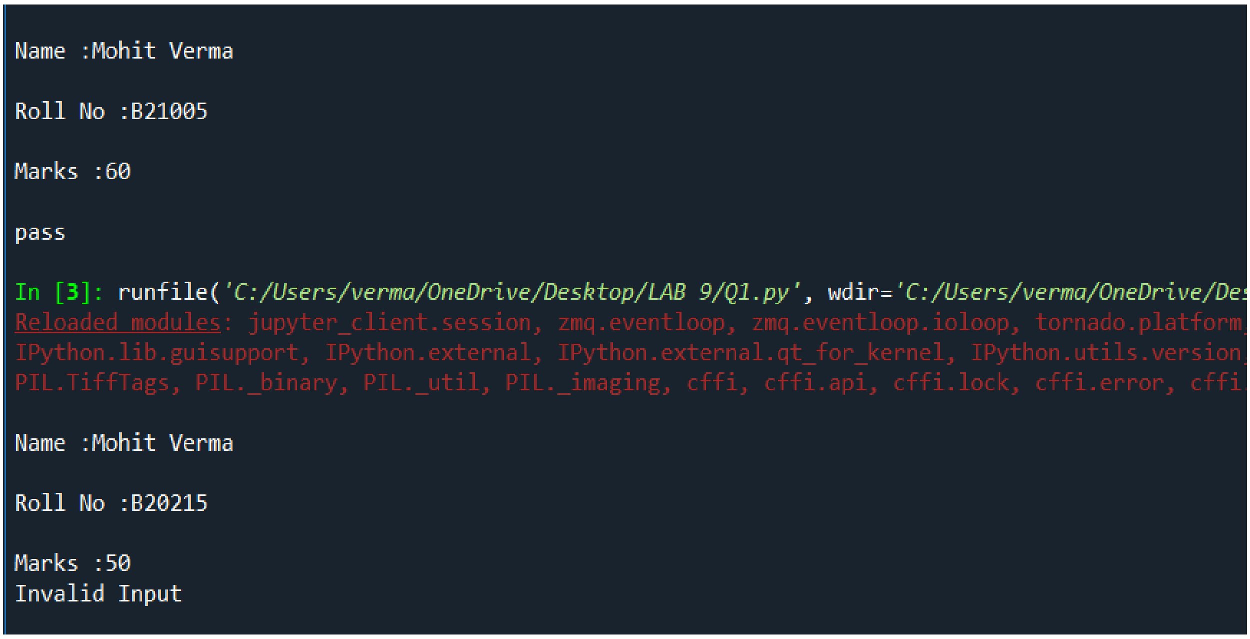
Questions & Answers

Q1) Create a class IC152 which captures the name, roll no and marks obtained by a student for IC152.The Code takes as input the name, roll no and marks of the student. Before you create an object, make sure that the object is of proper form by checking the following things-

- **Check the roll no to be of the form B21***.**
- **Check the grade to be in the range of 1 to 100.**

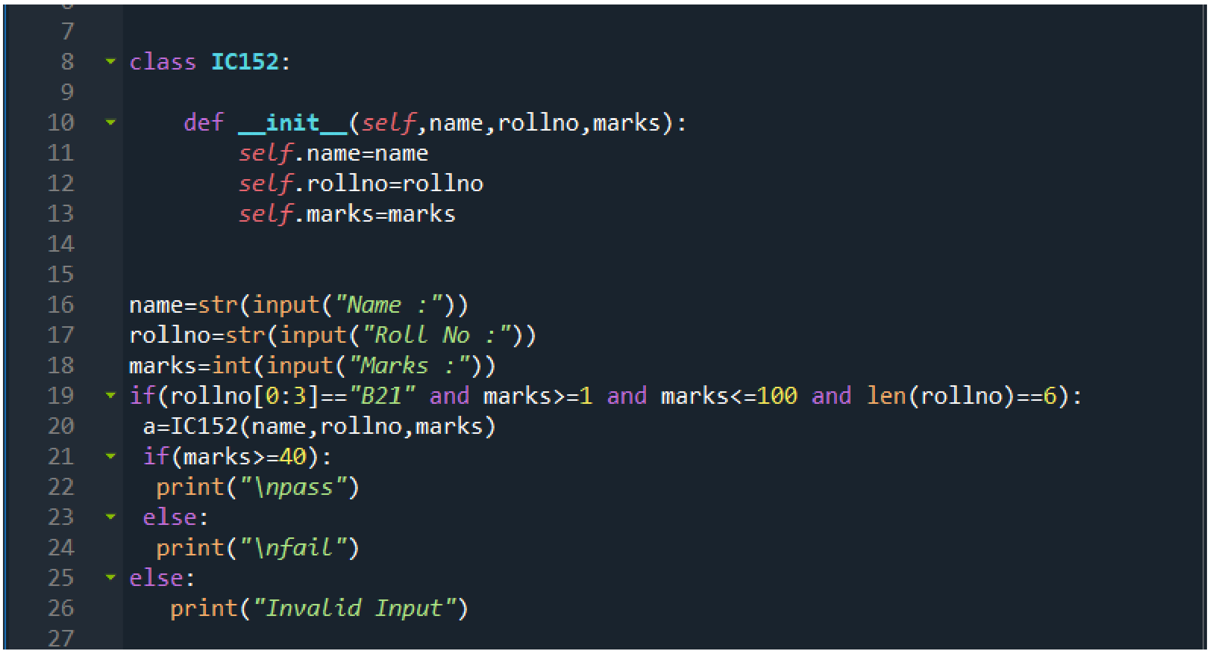
Your code should take the input and if the input is valid, output "pass" (without quotes) if the student has secured at least 40 marks and output "fail" if the student has secured less than 40 marks. If the input is invalid, output "Invalid Input".

Ans 1 (a) The output is shown-



```
Name :Mohit Verma

Roll No :B21005

Marks :60

pass

In [3]: runfile('C:/Users/verma/OneDrive/Desktop/LAB 9/Q1.py', wdir='C:/Users/verma/OneDrive/De
Reloaded modules: jupyter_client.session, zmq.eventloop, zmq.eventloop.ioloop, tornado.platform
IPython.lib.guisupport, IPython.external, IPython.external.qt_for_kernel, IPython.utils.version
PIL.TiffTags, PIL._binary, PIL._util, PIL._imaging, cffi, cffi.api, cffi.lock, cffi.error, cffi

Name :Mohit Verma

Roll No :B20215

Marks :50
Invalid Input
```

The code for output is shown below-

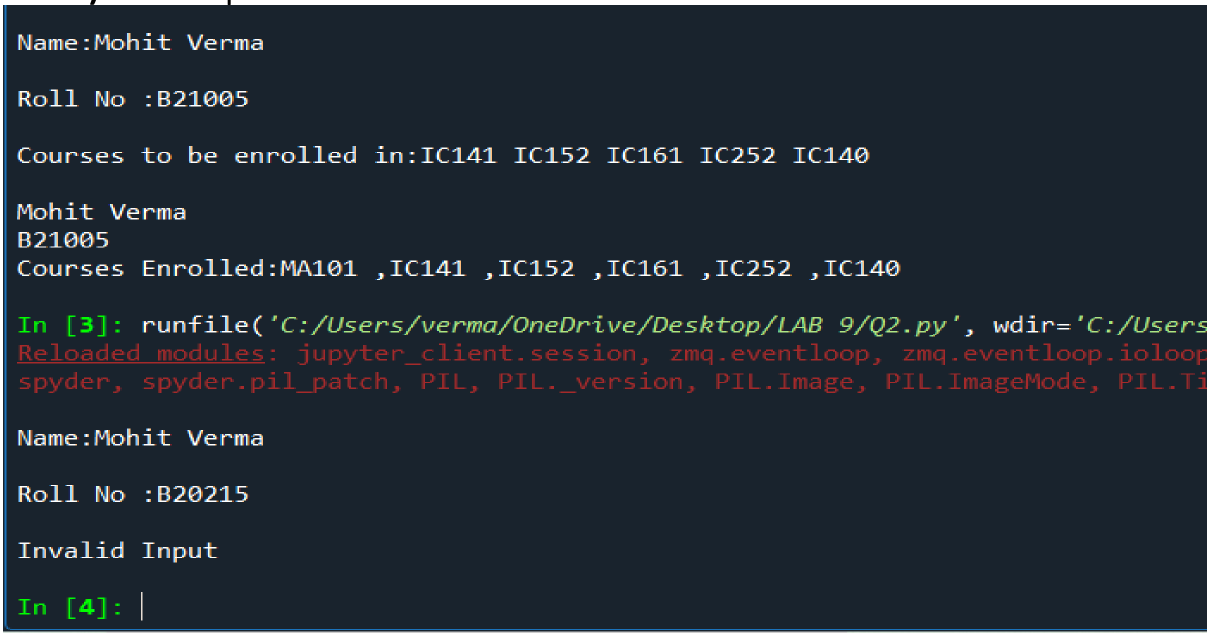


```python

class IC152:

    def __init__(self,name,rollno,marks):
        self.name=name
        self.rollno=rollno
        self.marks=marks


name=str(input("Name :"))
rollno=str(input("Roll No :"))
marks=int(input("Marks :"))
if(rollno[0:3]=="B21" and marks>=1 and marks<=100 and len(rollno)==6):
 a=IC152(name,rollno,marks)
 if(marks>=40):
  print("\npass")
 else:
  print("\nfail")
else:
   print("Invalid Input")

```

Q2) Create a class Student, which captures the name, roll no of a student and a list of courses enrolled by the student. The student is enrolled in MA101 by default. Create a method enrol, which can enrol a student in some course given as input.

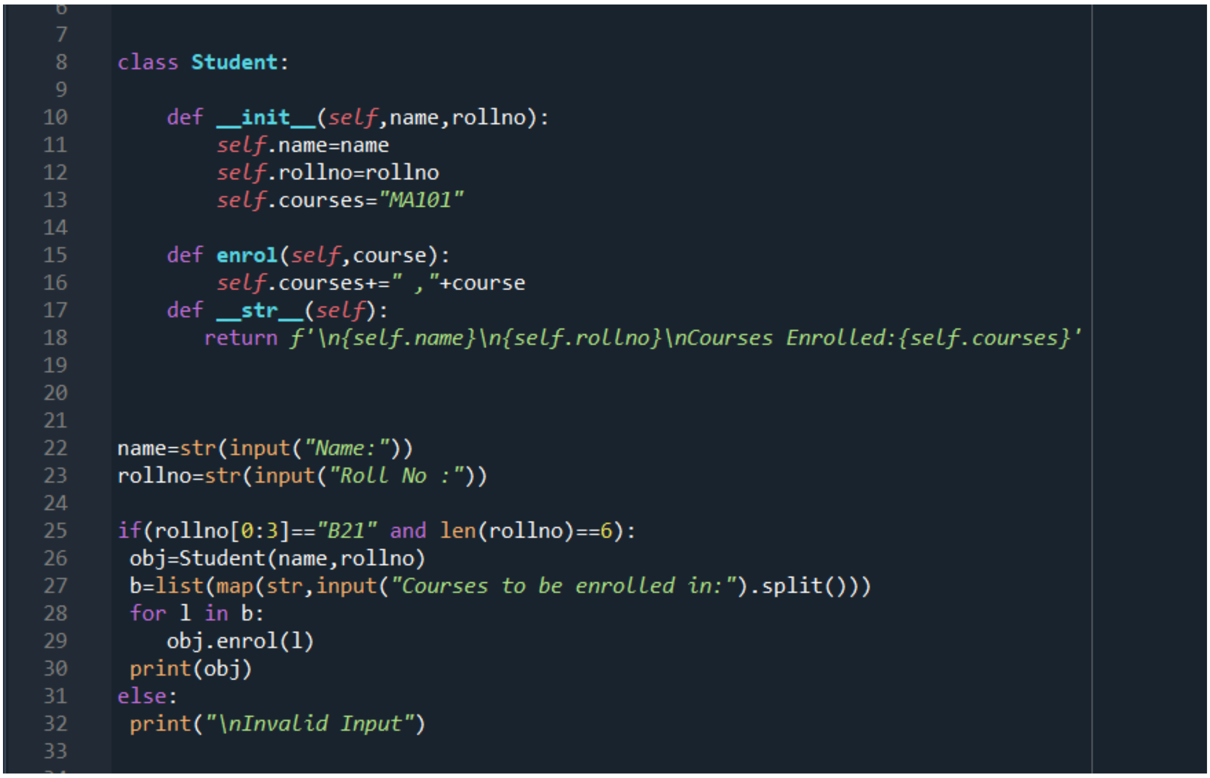
• Your task is to take as input a pair of operands which capture the name and roll no of a student and create an object of the class Student. After the object is created, your code asks for courses needed to be enrolled in. The user inputs the course codes separated by a single blank space and the student is enrolled in those course codes. Finally, your code outputs the name, roll no and courses enrolled by the student each on a different line. Courses enrolled are shown on the same line separated by a comma.

Ans 2) The output is shown below-



```
Name:Mohit Verma

Roll No :B21005

Courses to be enrolled in:IC141 IC152 IC161 IC252 IC140

Mohit Verma
B21005
Courses Enrolled:MA101 ,IC141 ,IC152 ,IC161 ,IC252 ,IC140

In [3]: runfile('C:/Users/verma/OneDrive/Desktop/LAB 9/Q2.py', wdir='C:/Users
Reloaded modules: jupyter_client.session, zmq.eventloop, zmq.eventloop.ioloop
spyder, spyder.pil_patch, PIL, PIL._version, PIL.Image, PIL.ImageMode, PIL.Ti

Name:Mohit Verma

Roll No :B20215

Invalid Input

In [4]: 
```

The code for output is shown below-



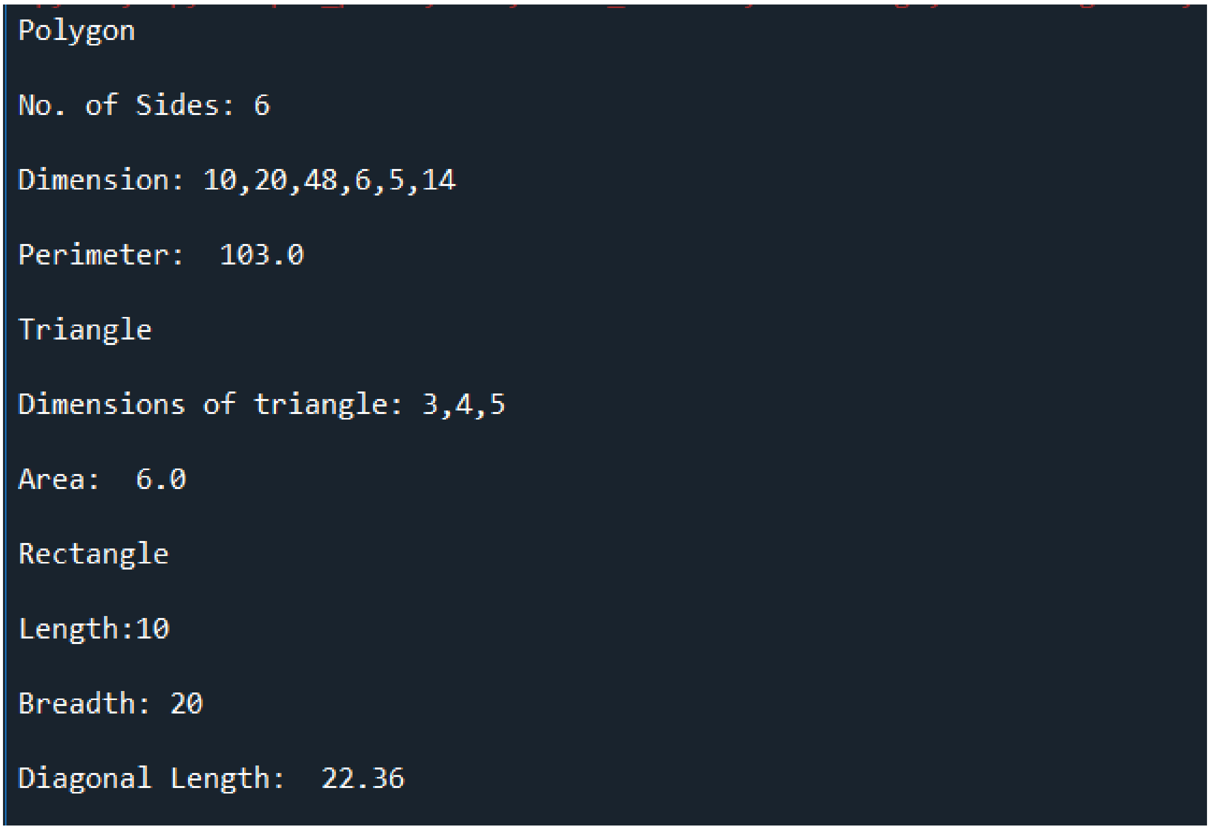
```python

class Student:

    def __init__(self,name,rollno):
        self.name=name
        self.rollno=rollno
        self.courses="MA101"

    def enrol(self,course):
        self.courses+=" ,"+course
    def __str__(self):
       return f'\n{self.name}\n{self.rollno}\nCourses Enrolled:{self.courses}'



name=str(input("Name:"))
rollno=str(input("Roll No :"))

if(rollno[0:3]=="B21" and len(rollno)==6):
 obj=Student(name,rollno)
 b=list(map(str,input("Courses to be enrolled in:").split()))
 for l in b:
    obj.enrol(l)
 print(obj)
else:
 print("\nInvalid Input")

```

Q3) Create a class Polygon, which has a method that prompts for a list, that stores the size of each edge of the Polygon. The class also has a method to compute the perimeter of the polygon. If we try the object of class Polygon, it should prompt for the number of sides of the polygon and then prompt for the measurement of the sides of the polygon. The edge lengths are given on single line separated by comma. Create two classes Triangle and Rectangle which inherit the class Polygon.

- **The class Triangle has a method to compute the area of the Triangle.**
- **The class Rectangle has a method which overrides the method to capture the sides of polygon and instead of asking for the lengths of 4 sides, only asks for lengths of two perpendicular sides. It also has a method to compute the length of the diagonal of the Rectangle. If we create an object of class Triangle, it should ask for the lengths of the sides and output the area.**

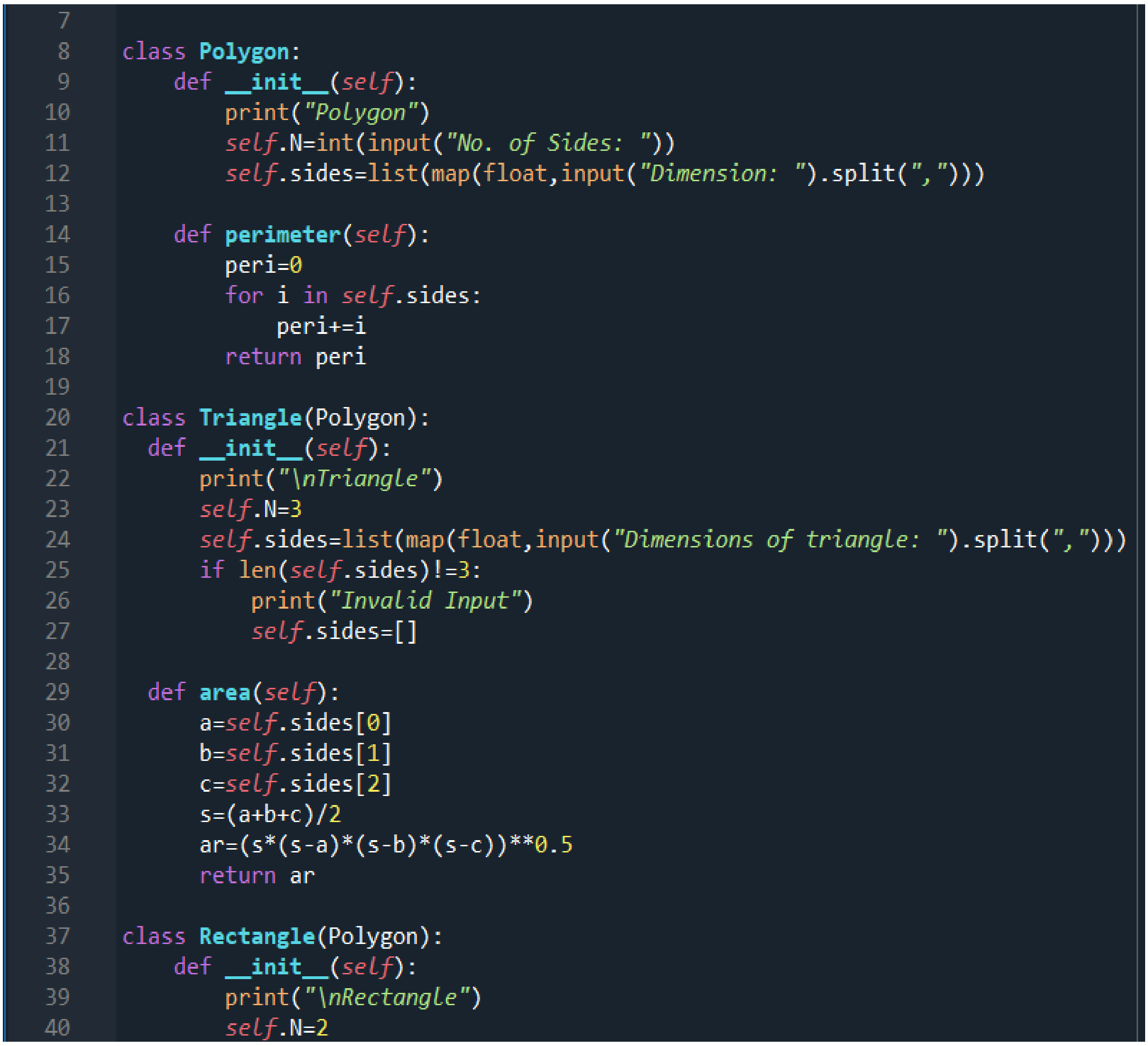
If we create an object of class Rectangle, it should ask for lengths of the 2 perpendicular sides and output the length of the diagonal of the Rectangle.

Ans 3) Output is shown below-

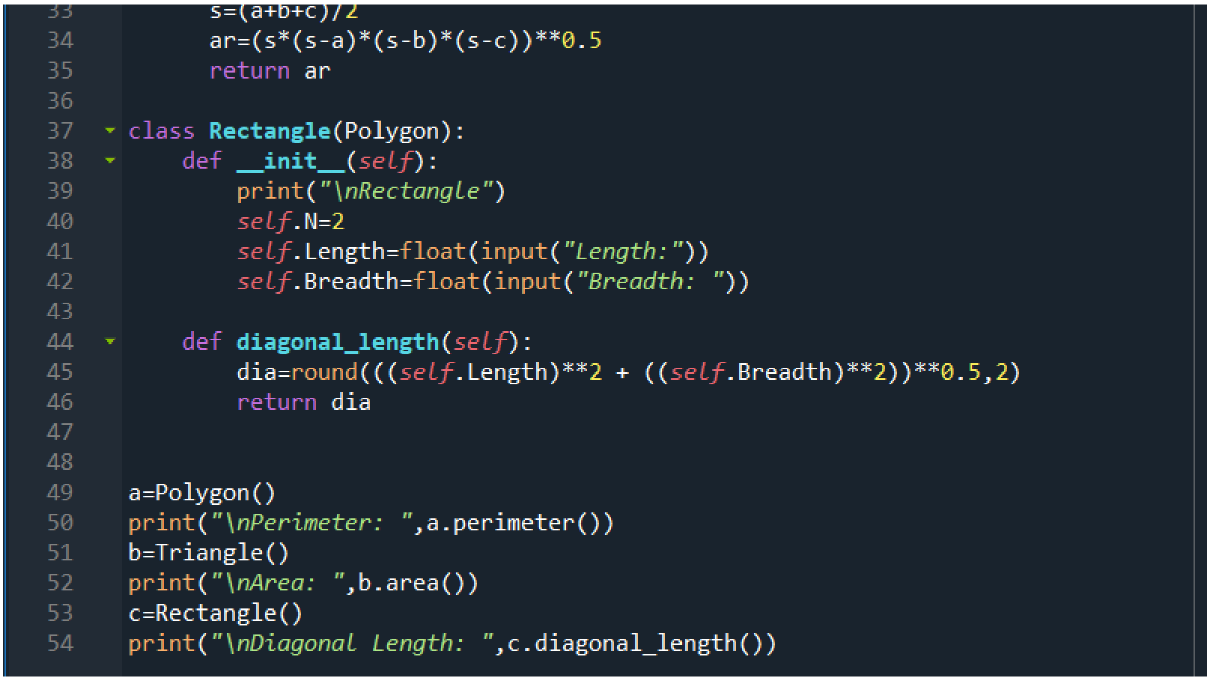


```
Polygon

No. of Sides: 6

Dimension: 10,20,48,6,5,14

Perimeter:  103.0

Triangle

Dimensions of triangle: 3,4,5

Area:  6.0

Rectangle

Length:10

Breadth: 20

Diagonal Length:  22.36
```

The code for output is shown below-



```python

class Polygon:
    def __init__(self):
        print("Polygon")
        self.N=int(input("No. of Sides: "))
        self.sides=list(map(float,input("Dimension: ").split(",")))

    def perimeter(self):
        peri=0
        for i in self.sides:
            peri+=i
        return peri

class Triangle(Polygon):
  def __init__(self):
      print("\nTriangle")
      self.N=3
      self.sides=list(map(float,input("Dimensions of triangle: ").split(",")))
      if len(self.sides)!=3:
          print("Invalid Input")
          self.sides=[]

  def area(self):
      a=self.sides[0]
      b=self.sides[1]
      c=self.sides[2]
      s=(a+b+c)/2
      ar=(s*(s-a)*(s-b)*(s-c))**0.5
      return ar

class Rectangle(Polygon):
    def __init__(self):
        print("\nRectangle")
        self.N=2
```



```
      s=(a+b+c)/2
      ar=(s*(s-a)*(s-b)*(s-c))**0.5
      return ar

class Rectangle(Polygon):
    def __init__(self):
        print("\nRectangle")
        self.N=2
        self.Length=float(input("Length:"))
        self.Breadth=float(input("Breadth: "))

    def diagonal_length(self):
        dia=round(((self.Length)**2 + ((self.Breadth)**2))**0.5,2)
        return dia


a=Polygon()
print("\nPerimeter: ",a.perimeter())
b=Triangle()
print("\nArea: ",b.area())
c=Rectangle()
print("\nDiagonal Length: ",c.diagonal_length())
```

…