Started on	Thursday, 22 August 2024, 3:29 PM
State	Finished
Completed on	Thursday, 22 August 2024, 4:04 PM
Time taken	34 mins 59 secs
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Write a Python program for simply using the overloading operator for adding two objects.

For example:

Input	Result
23 21 hello world	: 44 : helloworld

Answer: (penalty regime: 0 %)

```
class example:
    def __init__(self,x):
        self.x=x
    def __add__(self,u):
        return self.x+u.x
object_1=example(int(input()))
object_2=example(int(input()))
print(": ",object_1+object_2)
object_3=example(str(input()))
object_4=example(str(input()))
print(": ",object_3+object_4)

```

	Input	Expected	Got	
✔	23 21 hello world	: 44 : helloworld	: 44 : helloworld	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Define the abstract base class named Polygon and also define the abstract method. This base class inherited by the various subclasses. Implement the abstract method in each subclass. Create the object of the subclasses and invoke the **sides()** method.

For example:

Result
Triangle has 3 sides I have 4 sides Pentagon has 5 sides Hexagon has 6 sides

Answer: (penalty regime: 0 %)

Reset answer

```
from abc import ABC

class Polygon(ABC):

   # abstract method
   def sides(self):
      pass

class Triangle(Polygon):


   def sides(self):
      print("Triangle has 3 sides")

class Pentagon(Polygon):


   def sides(self):
       print("Pentagon has 5 sides")
class Hexagon(Polygon):

   def sides(self):
```

	Expected	Got	
✔	Triangle has 3 sides I have 4 sides Pentagon has 5 sides Hexagon has 6 sides	Triangle has 3 sides I have 4 sides Pentagon has 5 sides Hexagon has 6 sides	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Create two new, independent classes: `Turtle` and `Frog`. When you instantiate an object from the `Turtle` class, the object will use the `type` method as it is defined in that class. The same will be true of objects instantiated from the `Frog` class, despite the fact that the methods have the same name.

For example:

Result
turtle frog

Answer: (penalty regime: 0 %)

Reset answer

```
class Turtle:
    def type(self):
        print("turtle")

class Frog:
    def type(self):
        print("frog")
obj_sea_turtle=Turtle()
obj_treefrog=Frog()

obj_sea_turtle.type()
obj_treefrog.type()
```

	Expected	Got	
✔	turtle frog	turtle frog	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Incorrect

Mark 0.00 out of 20.00

Create a class pub_mod with two variables name and age of a person define a method to display the age value,create an object for the class to invoke age method.

For example:

Result
Name: Jason Age: 35

Answer: (penalty regime: 0 %)

Reset answer

```
# illustrating public members & public access modifier
class pub_mod:
    # constructor
    def __init__(self, name, age):
        self.name = name;
        self.age = age;

    def Age(self):
        # accessing public data member
        ____________
# creating object with values jason,35

# accessing public data member
print("Name: ", obj.name)
# calling public member function of the class
obj.Age()

```

	Expected	Got	
✖	Name: Jason Age: 35	***Run error*** Traceback (most recent call last): File "__tester__.python3", line 14, in <module> print("Name: ", obj.name) NameError: name 'obj' is not defined	✖

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Write a python code to create a Employee class with the parameterized constructors with the parameters as id and name,print the details of the Employee using def display(self) methd.

For example:

Input	Result
1 nithin	Hello my id is : 1 My name is : nithin

Answer: (penalty regime: 0 %)

```
class Employee:
    def __init__(self,id,name):
        self.id=id
        self.name=name
    def display(self):
        print("Hello my id is :",self.id)
        print("My name is :",self.name)

id=int(input())
name=input()
a=Employee(id,name)
a.display()
```

	Input	Expected	Got	
✔	1 nithin	Hello my id is : 1 My name is : nithin	Hello my id is : 1 My name is : nithin	✔
✔	110 saveetha	Hello my id is : 110 My name is : saveetha	Hello my id is : 110 My name is : saveetha	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.