Started on	Wednesday, 21 August 2024, 10:30 AM
State	Finished
Completed on	Wednesday, 21 August 2024, 11:08 AM
Time taken	37 mins 58 secs
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

write a python program to check whether two persons name is same or not .print the result in true or false

For example:

Input	Result
wammu kars	False

Answer: (penalty regime: 0 %)

```
a=input()
b=input()
print (a==b)
```

	Input	Expected	Got	
✔	wammu kars	False	False	✔
✔	saveetha saveetha	True	True	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Create two new classes: Lion and Giraffe The outputs from this program are carnivore and herbivore, respectively. The two classes both use the method name diet, but they define those methods differently. An object instantiated from the Lion class will use the method as it is defined in that class. The Giraffe class may have a method with the same name, but objects instantiated from the Lion class won't interact with it.

For example:

Result
carnivore herbivore

Answer: (penalty regime: 0 %)

Reset answer

```
class Lion:
    def diet(self):
        print("carnivore")
class Giraffe:
   def diet(self):
       print("herbivore")

obj_lion=Lion()
obj_giraffe=Giraffe()

obj_lion.diet()

obj_giraffe.diet()
```

	Expected	Got	
✔	carnivore herbivore	carnivore herbivore	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

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

class name : accessories

For example:

Input	Result
69 68 APPLE LAPTOP	Rate is : 137 accessories are: APPLELAPTOP

Answer: (penalty regime: 0 %)

```
1
2  class accessories:
3
4      def show(self):
5          print("accessories are: ",a+b)
6
7  c=eval(input())
8  d=eval(input())
9  print("Rate is :",c+d)
10 a=input()
11 b=input()
12 k=accessories()
13 k.show()
```

	Input	Expected	Got	
✔	69 68 APPLE LAPTOP	Rate is : 137 accessories are: APPLELAPTOP	Rate is : 137 accessories are: APPLELAPTOP	✔

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

import the **abc module** to create the abstract base class. Create the Car class that inherit the ABC class and define an abstract method named mileage(). then inherit the base class from the three different subclasses and implement the abstract method differently. Create the objects to call the abstract method.

For example:

Result
The mileage is 30kmph The mileage is 27kmph The mileage is 25kmph The mileage is 24kmph

Answer: (penalty regime: 0 %)

Reset answer

```
    def mileage(self):
        print("The mileage is 30kmph")
class Suzuki(Car):
    def mileage(self):
        print("The mileage is 25kmph ")
class Duster(Car):
     def mileage(self):
          print("The mileage is 24kmph ")

class Renault(Car):
    def mileage(self):
            print("The mileage is 27kmph ")


t=Tesla()
t.mileage()
r=Renault()
r.mileage()
s = Suzuki()
s.mileage()
d = Duster()
d.mileage()
```

	Expected	Got	
✔	The mileage is 30kmph The mileage is 27kmph The mileage is 25kmph The mileage is 24kmph	The mileage is 30kmph The mileage is 27kmph The mileage is 25kmph The mileage is 24kmph	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.