Started on	Friday, 4 October 2024, 2:24 PM
State	Finished
Completed on	Friday, 4 October 2024, 2:43 PM
Time taken	19 mins 3 secs
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Create an abstract base class has a concrete method sleep() that will be the same for all the child classes. So, we do not define it as an abstract method, thus saving us from code repetition. On the other hand, the sounds that animals make are all different. For that purpose, define the sound() method as an abstract method. then implement it in all child classes.

For example:

Result
I am going to sleep in a while I can meow I can hiss

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
from abc import ABC,abstractmethod

class Animal(ABC):
    def sound(self):
        print("I am going to sleep in a while")

class Sleep(Animal):
    def sound(self):
        print("I am going to sleep in a while")

class Snake(Animal):
    def sound(self):
        print("I can hiss")

class Cat(Animal):
    def sound(self):
        print("I can meow")
e=Sleep()
```

	Expected	Got	
✔	I am going to sleep in a while I can meow I can hiss	I am going to sleep in a while I can meow I can hiss	✔

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

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
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**

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

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

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

    def __Age__(self):

print("Name: ", obj.name)

obj.Age()
```

Syntax Error(s)

```
Sorry: IndentationError: expected an indented block (__tester__.python3, line 10)
```

Incorrect

Marks for this submission: 0.00/20.00.

Question **4**

Correct

Mark 20.00 out of 20.00

Write a python program to read one string and integer and print the values.

For example:

Input	Result
1 saveetha	1 saveetha

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
a=int(input())
b=str(input())
print(a,b)
```

	Input	Expected	Got	
✔	1 saveetha	1 saveetha	1 saveetha	✔
✔	101 computer science	101 computer science	101 computer science	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

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

class name : fruits

object name : apple, mango, a and b

For example:

Input	Result
100 200 banana orange	apple and mango mixed: 300 fruit mix: bananaorange

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
class Accessories:
    def __init__(self,a,b):
        self.a=a
        self.b=b
    def __mul__(self,z):
        return self.a+z.a,self.b+z.b

a=int(input())
b=int(input())
print("apple and mango mixed:" ,a+b)
c=str(input())
d=str(input())
print("fruit mix: " ,c+d)
```

	Input	Expected	Got	
✔	100 200 banana orange	apple and mango mixed: 300 fruit mix: bananaorange	apple and mango mixed: 300 fruit mix: bananaorange	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.