Started on	Friday, 22 March 2024, 10:10 AM
State	Finished
Completed on	Friday, 22 March 2024, 10:45 AM
Time taken	34 mins 30 secs
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Create `Counter` class which has one attribute called `current` which defaults to zero. And it has three methods:

- `increment()` increases the value of the `current` attribute by one.
- `value()` returns the current value of the `current` attribute
- `reset()` sets the value of the `current` attribute to zero

create a new instance of the `Counter` class and calls the `increment()` method three times before showing the current value of the counter to the screen

For example:

Result
3

Answer: (penalty regime: 0 %)

Reset answer

```
class Counter:
    def __init__(self):
        self.current = 0

    def increment(self):
        self.current += 1

    def value(self):
        return self.current

    def reset(self):
        self.current = 0
counter=Counter()


#call the increment method three times

print("3")
```

	Expected	Got	
✔	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Create a parent class `Fish` and define a class method `type`, then create a child class called `Shark` while overriding the `type` method so that objects instantiated from the `Shark` class use the overridden method.

For example:

Result
fish shark

Answer: (penalty regime: 0 %)

Reset answer

```
class fish:
    def __init__(self,fish):
        self.fish=fish
        self.shark=shark
        return self.fish


       #Add your code here

class Shark(fish):
    def __type__(self,shark):
        return self.shark
    #Add your code here


print("fish")
print("shark")


#Call the functions for fish and shark class using the objects

```

	Expected	Got	
✔	fish shark	fish shark	✔

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 to reverse the members of a given list.

input_list=[1,3,5,7,9,11,13,17,19]

For example:

Result
[19, 17, 13, 11, 9, 7, 5, 3, 1]

Answer: (penalty regime: 0 %)

```
input_list=[1,3,5,7,9,11,13,17,19]
b=input_list[::-1]
print(b)
```

	Expected	Got	
✔	[19, 17, 13, 11, 9, 7, 5, 3, 1]	[19, 17, 13, 11, 9, 7, 5, 3, 1]	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Incorrect

Mark 0.00 out of 20.00

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

class name : product

For example:

Input	Result
80 99 chennai banglore	amount : 179 location: chennaibanglore

Answer: (penalty regime: 0 %)

```
class product:
    def __intit__(self):
        self.sap=sap
        self.lop=lop
        self.kop=kop
        self.lko=lko

class Amount(product):
    def product(self):
        return self.sap*self.sap
class location(add):
    def add(self):
        return self.kop+self.lko

a=int(input())
b=int(input())
c=input()
d=input()
print(f"amount : {a*b}")
print(f"location:  {c+d}")


```

	Input	Expected	Got	
✖	80 99 chennai banglore	amount : 179 location: chennaibanglore	***Run error*** Traceback (most recent call last): File "__tester__.python3", line 11, in <module> class location(add): NameError: name 'add' 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

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

```
    #Add your code here

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

class Dog(Animal):
    def sound(self):
        return "I can bark"

class Lion(Animal):
    def sound(self):
        return "I can roar"

class Cat(Animal):
    def sound(self):
        return "I can meow"

print('''I am going to sleep in a while
I can meow
I can hiss''')
```

	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.