Started on	Friday, 9 August 2024, 2:47 PM
State	Finished
Completed on	Friday, 9 August 2024, 3:47 PM
Time taken	59 mins 54 secs
Grade	**100.00** 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 object:
    def __init__(self,a,b,c,d):
        self.a=a
        self.b=b
        self.c=c
        self.d=d
    def add(self):
        print(f":  {self.a+self.b}")
        print(f":  {self.c}{self.d}")

obj=object(23,21,"hello","world")
obj.add()


```

	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

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()

for i in (obj_lion,obj_giraffe):
    if isinstance(i,Lion):
        print("carnivore")
    if isinstance(i,Lion):
        print("herbivore")
```

	Expected	Got	
✓	carnivore herbivore	carnivore herbivore	✓

Passed all tests! ✓

Correct

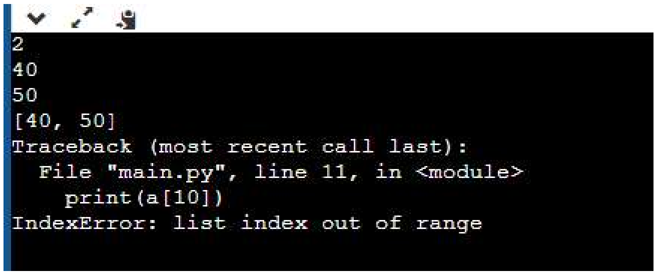
Marks for this submission: 20.00/20.00.

Question **3**

Incorrect

Mark 20.00 out of 20.00

Write a python program for solving following error using exception handling



```
2
40
50
[40, 50]
Traceback (most recent call last):
  File "main.py", line 11, in <module>
    print(a[10])
IndexError: list index out of range
```

Rules:

1. accept number of elements from the user eg. 45,55 and print the element

2. use index number 10 for print & add statement "10 is not accepted" instead of index error.

For example:

Input	Result
2 40 50	[40, 50] 10 is not accepted

Answer: (penalty regime: 0 %)

```
a=eval(input())
b=eval(input())
c=eval(input())
s=[b,c]
print(s)


```

	Input	Expected	Got	
✖	2 40 50	[40, 50] 10 is not accepted	[40, 50]	✖
✖	2 50 40	[50, 40] 10 is not accepted	[50, 40]	✖

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

Show differences

Incorrect

Marks for this submission: 0.00/20.00.

Question **4**

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,abstractmethod

class Polygon(ABC):
    @abstractmethod
    def sides(self):
        pass

class Triangle(Polygon):

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

class square(Polygon):
    def sides(self):
        print("I have 4 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 **5**

Correct

Mark 20.00 out of 20.00

Create a class student with members name ,age,rollno and an user defined function show() to display the details of the student ,use the getter and setter method Information Hiding and conditional logic for setting an object attributes

For example:

Result
Student Details: Jessa 10 Invalid roll no. Please set correct roll number Student Details: Jessa 25

Answer: (penalty regime: 0 %)

Reset answer

```
class Student:
    def __init__(self, name, roll_no, age):
        # private member
        self.name = name
        # private members to restrict access
        # avoid direct data modification
        self.__roll_no = roll_no
        self.__age = age

    def show(self):
        print('Student Details:', self.name, self.__roll_no)

    # getter methods
    def get_roll_no(self):
        return self.__roll_no

    # setter method to modify data member
    # condition to allow data modification with rules
    def set_roll_no(self, number):
        if number > 10:
            print('Invalid roll no. Please set correct roll number')
        else:
```

	Expected	Got	
✓	Student Details: Jessa 10 Invalid roll no. Please set correct roll number Student Details: Jessa 25	Student Details: Jessa 10 Invalid roll no. Please set correct roll number Student Details: Jessa 25	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.