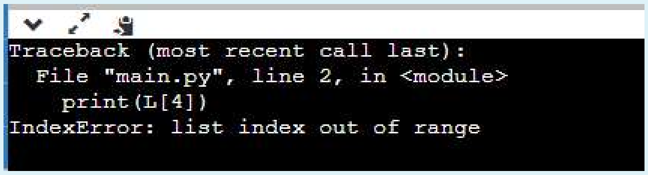
Started on	Tuesday, 30 April 2024, 2:59 PM
State	Finished
Completed on	Tuesday, 30 April 2024, 3:33 PM
Time taken	33 mins 56 secs
Grade	**80.00** out of 100.00

Question **1**

Not answered

Mark 0.00 out of 20.00

Write a python program for solving the following error using Exception Handling



```
Traceback (most recent call last):
  File "main.py", line 2, in <module>
    print(L[4])
IndexError: list index out of range
```

Rules :

assign the variable 'L' with 3 elements 'laptop','mobile','pen'.

use index number '4' for print and clear these error with the statement "check index range" using Exception Handling.

Answer: (penalty regime: 0 %)

Question **2**

Correct

Mark 20.00 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):
        return self.age

obj=pub_mod("Jason",35)

print("Name: ", obj.name)
print("Age: ", obj.Age())

```

	Expected	Got	
✓	Name: Jason Age: 35	Name: Jason Age: 35	✓

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**

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

```
from abc import ABC, abstractmethod
class Car(ABC):
    def mileage(self):
        pass

class Tesla(Car):
    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 ")
n = Tesla()
n.mileage()
t = Renault()
t.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.

Question **5**

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
10 20 can teen	adding integers : 30 adding strings : canteen

Answer: (penalty regime: 0 %)

```
class priya:
    def __init__(self,a,b,c,d):
        self.a=a
        self.b=b
        self.c=c
        self.d=d
    def __add__(self,other):
        return self.a+other.a,self.b+other.b,self.c+other.b,self.c+other
ob1=int(input())
ob2=int(input())
ob3=str(input())
ob4=str(input())
print("adding integers :",ob1+ob2)
print("adding strings :",ob3+ob4)
```

	Input	Expected	Got	
✓	10 20 can teen	adding integers : 30 adding strings : canteen	adding integers : 30 adding strings : canteen	✓
✓	20 30 sky walk	adding integers : 50 adding strings : skywalk	adding integers : 50 adding strings : skywalk	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.