PYTHON MODULE -6

1. create two classes Cat and Dog. They share a similar structure and have the same method names info() and make_sound() pack these two different objects into a tuple and iterate through it using a common animal variable. It is possible due to polymorphism.

For example:

Input	Result
kitty 2.5 fluffy 4	Meow I am a cat. My name is kitty. I am 2.5 years old. Meow Bark I am a dog. My name is fluffy. I am 4 years old. Bark

PROGRAM:

```
class Cat:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def make_sound(self):
        print("Meow")
    def info(self):
        print(f"I am a cat. My name is {self.name}. I am {self.age} years old.")



class Dog:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def make_sound(self):
        print("Bark")
    def info(self):
        print(f"I am a dog. My name is {self.name}. I am {self.age} years old.")

cat1 = Cat(input(),input())
```

```
dog1 = Dog(input(),input())


for animal in (cat1, dog1):
    animal.make_sound()
    animal.info()
    animal.make_sound()
```

RESULT:

Input	Expected	Got
kitty 2.5 fluffy 4	Meow I am a cat. My name is kitty. I am 2.5 years old. Meow Bark I am a dog. My name is fluffy. I am 4 years old. Bark	Meow I am a cat. My name is kitty. I am 2.5 years old. Meow Bark I am a dog. My name is fluffy. I am 4 years old. Bark
jack 3 shiro 2.3	Meow I am a cat. My name is jack. I am 3 years old. Meow Bark I am a dog. My name is shiro. I am 2.3 years old. Bark	Meow I am a cat. My name is jack. I am 3 years old. Meow Bark I am a dog. My name is shiro. I am 2.3 years old. Bark

d all tests! ✔

2. created two classes Tiger and Elephant. They have the same instance method names color() and nature(). However, we have not linked both the classes nor have we used inheritance. Pack two different objects into a tuple and iterate through it using a car variable.

For example:

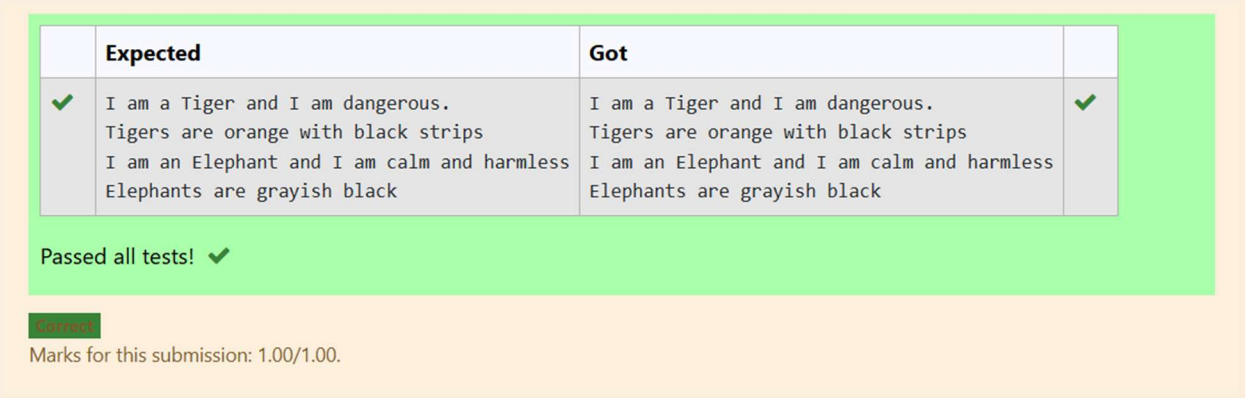
Result
I am a Tiger and I am dangerous. Tigers are orange with black strips I am an Elephant and I am calm and harmless Elephants are grayish black

PROGRAM:

```
class Tiger():
   def nature(self):
      print('I am a Tiger and I am dangerous.')


   def color(self):
      print('Tigers are orange with black strips')


class Elephant():
   def nature(self):
      print("I am an Elephant and I am calm and harmless")
   def color(self):
      print("Elephants are grayish black")
obj1 = Tiger()
```

```
obj2 = Elephant()

for animal in (obj1, obj2):
    animal.nature()
    animal.color()
```

RESULT:



	Expected	Got	
✔	I am a Tiger and I am dangerous. Tigers are orange with black strips I am an Elephant and I am calm and harmless Elephants are grayish black	I am a Tiger and I am dangerous. Tigers are orange with black strips I am an Elephant and I am calm and harmless Elephants are grayish black	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

3. write a python program to overload less than operator

class name should be A

ob1 = A(2)

ob2 = A(3)

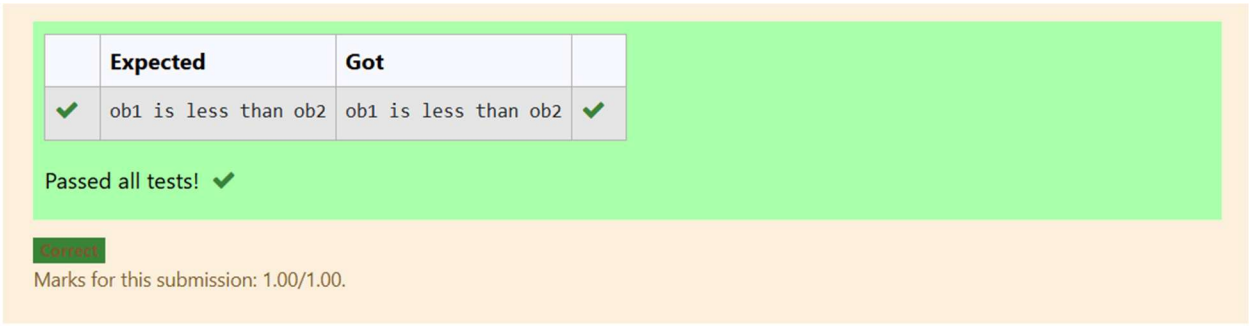
For example:

Result
ob1 is less than ob2

PROGRAM:

```
class A:
   def __init__(self,a):
      self.a=a
   def __lt__(self,z):
      return self.a < z.a
ob1=A(2)
ob2=A(3)
if ob1 < ob2:
   print("ob1 is less than ob2")
```

RESULT:



	Expected	Got	
✔	ob1 is less than ob2	ob1 is less than ob2	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

4. Create an abstract class Invoice, and abstract method invoice () under def statement also has two child class derived from Invoice and does the functionality. Then, using an Object ‘aa’, the methods are invoked.

For example:

Result
paycheque of- 6500 Purchase of the product- 6500 True pay through card of- 2600 Purchase of the product- 2600 True

PROGRAM:

```
from abc import ABC, abstractmethod
class Invoice(ABC):
    def final_bill(self, pay):
        print('Purchase of the product- ', pay)
    @abstractmethod
    def Invoice(self, pay):
        pass
class paycheque(Invoice):
    def Invoice(self,pay):
```

```
        print("paycheque of- ",pay)
class CardPayment(Invoice):
    def Invoice(self,pay):
        print("pay through card of- ",pay)
aa = paycheque()
aa.Invoice(6500)
aa.final_bill(6500)
print(isinstance(aa,Invoice))
aa = CardPayment()
aa.Invoice(2600)
aa.final_bill(2600)
print(isinstance(aa,Invoice))
```

RESULT:

	Expected	Got	
✔	paycheque of- 6500 Purchase of the product- 6500 True pay through card of- 2600 Purchase of the product- 2600 True	paycheque of- 6500 Purchase of the product- 6500 True pay through card of- 2600 Purchase of the product- 2600 True	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

5. 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

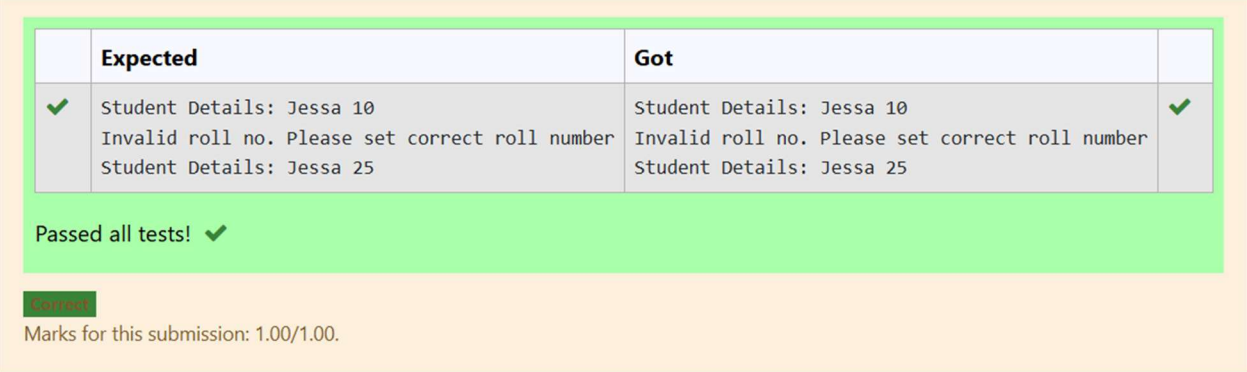
PROGRAM:

```
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 > 50:
            print('Invalid roll no. Please set correct roll number')
        else:
            self.__roll_no = number

jessa = Student('Jessa', 10, 15)

# before Modify
jessa.show()# call show()
jessa.set_roll_no(120)
#changing roll number as 120 using setter



jessa.set_roll_no(25)
```

jessa.show()

RESULT:



	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: 1.00/1.00.