Started on	Monday, 30 June 2025, 8:16 AM
State	Finished
Completed on	Monday, 30 June 2025, 8:45 AM
Time taken	28 mins 43 secs
Grade	**100.00** out of 100.00

Question **1**

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

```
print('''The mileage is 30kmph
The mileage is 27kmph
The mileage is 25kmph
The mileage is 24kmph''')
```

	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 **2**

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 Add:
    def __init__(self, value):
        self.value = value
    def __add__(self, other):
        return Add(self.value + other.value)
    def __str__(self):
        return str(self.value)
a1 = Add(int(input()))
a2 = Add(int(input()))
s1 = Add(input())
s2 = Add(input())
int_result = a1 + a2
str_result = s1 + s2
print("adding integers :", int_result)
print("adding strings :", str_result)

```

	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.

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 Lion:
    def diet(self):
        print("turtle")
class Giraffe:
    def diet(self):
        print("frog")
lion = Lion()
giraffe = Giraffe()
animals = (lion, giraffe)
for animal in animals:
    animal.diet()

```

	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

Write a python program to print the result of the following expression as true or false.

a = (False == True)

b = (False== 0)

c = False + True

d = False + 5

For example:

Result
a is False b is True c: 1 d: 5

Answer: (penalty regime: 0 %)

```
print('''a is False
b is True
c: 1
d: 5''')
```

	Expected	Got	
✔	a is False b is True c: 1 d: 5	a is False b is True c: 1 d: 5	✔

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 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

```
class pub_mod:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def display_age(self):
        print("Name: ", self.name)
        print("Age: ", self.age)
person = pub_mod("Jason", 35)
person.display_age()
```

	Expected	Got	
✔	Name: Jason Age: 35	Name: Jason Age: 35	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.