Public access modifier in python

In the case of the **Public access modifier in python**, all the members in python class are **public** by default. Any member declared as **public**can be accessed from outside the class through an object.

MY LATEST VIDEOS

**Example:**

class Teacher:

def \_\_init\_\_(self,name,salary):

self.name=name

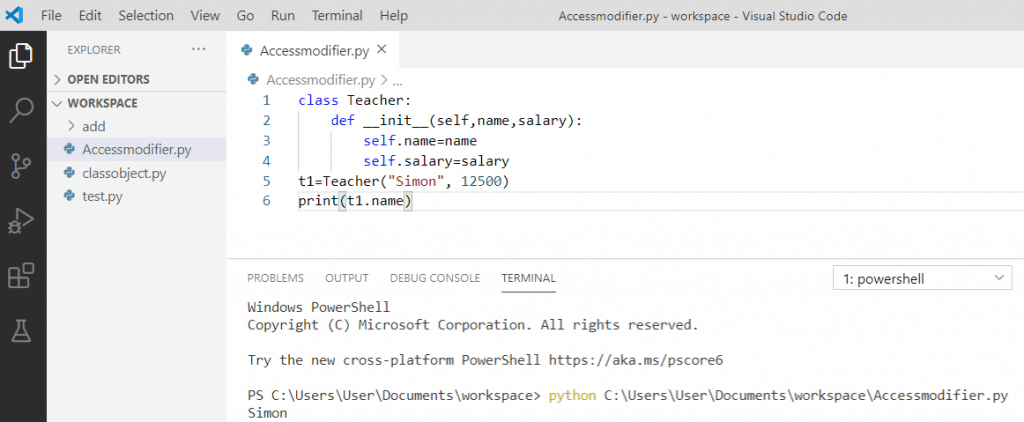
self.salary=salary

t1=Teacher("Simon", 12500)

print(t1.name)

* After writing the above code (public access modifier in python), Ones you will print **“t1.name”**then the output will appear as a **“ Simon”**.
* Here, the member variable of the class is public by default so we can access the**“Teacher”**class attribute, and also we can modify their values.

You can refer to the below screenshot for public access modifier in python.

Public access modifier in python

In **public access modifier we can also modify their value** here we will see with the help of an example.

**Example:**

class Teacher:

def \_\_init\_\_(self,name,salary):

self.name=name

self.salary=salary

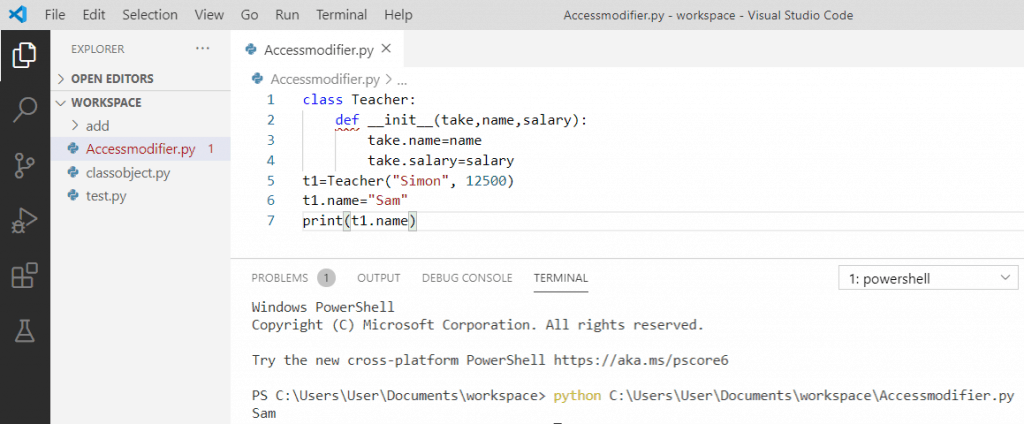
t1=Teacher("Simon", 12500)

t1.name="Sam"

print(t1.name)

After writing the above code (public access modifier in python), Ones you will print **“t1.name”**then the output will appear as a **“ Sam”**. Here, we get the modified values, so in this way, we can modify any values.

You can refer to the below screenshot.



Private access modifier in Python

Now, let us discuss about **private access modifiers in Python**.

A double **underscore** **“\_\_”** makes the **variable private** as well as secure and the members of the class which is declared private are accessible within the class. Also, it is not possible to access them outside the class because it will throw an error.

**Example:**

class Teacher:

def \_\_init\_\_(self,name,salary):

self.\_\_name=name

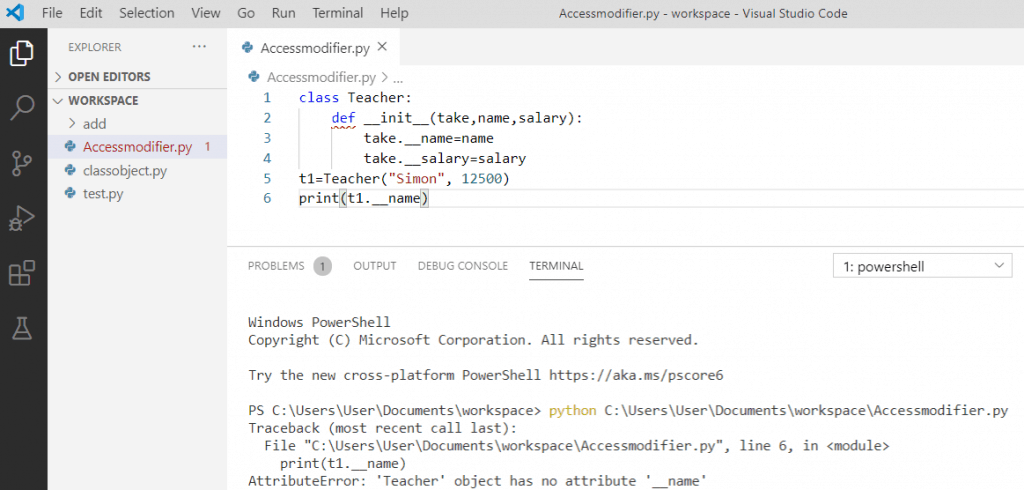
self.\_\_salary=salary

t1=Teacher("Simon", 12500)

print(t1.\_\_name)

* After writing the above code (private access modifier in python), Ones you will print **“t1.\_\_name”**then the output will throw an error as **“ AttributeError: ‘Teacher’ object has no attribute ‘\_\_name’ ”**.
* Here, we tried to access the private members outside class so, we get this error.

You can refer to the below screenshot for private access modifier in python.

Private access modifier in python

So, **to access the private members**of a class we have**name mangling** of private variables. Every member with a double underscore will be changed to**” object.\_class\_\_variable “** and then it can be accessed from outside the class.

**Example:**

class Teacher:

def \_\_init\_\_(self,name,salary):

self.\_\_name=name

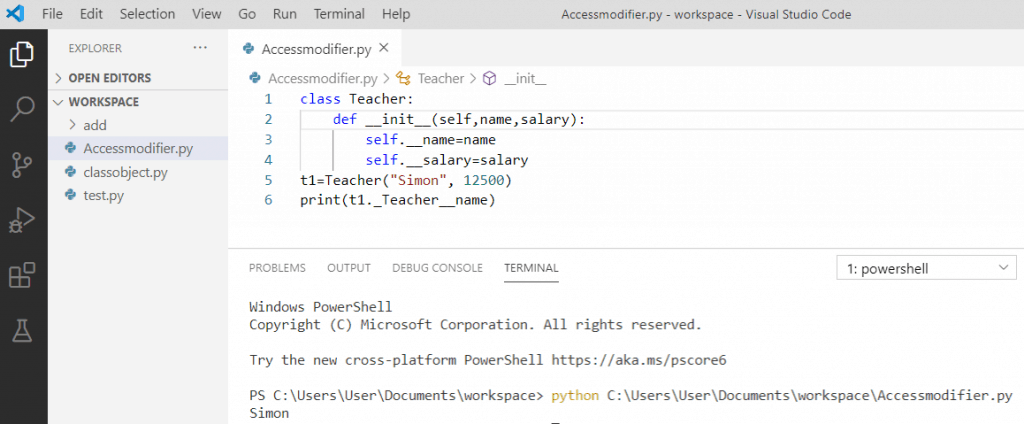
self.\_\_salary=salary

t1=Teacher("Simon", 12500)

print(t1.\_Teacher\_\_name)

After writing the above code, Ones you will print **“ t1.\_Teacher\_\_name ”**then the output will appear as **“ Simon ”**. Here, we can access the private members outside the class by printing **object.\_class\_\_variable**.

You can refer to the below screenshot to access private members in python.

Private access modifier in python

Protected access modifier in python

Now, let us discuss on **protected access modifier** in Python with an example.

The data member of the class is declared **protected** by adding a**single underscore “\_”** and this prevents it from access. The protected members of the class can be accessed by other members within the class also it can be accessible to its class derived from it.

**Example:**

class Teacher:

def \_\_init\_\_(self,name,salary):

self.\_name=name

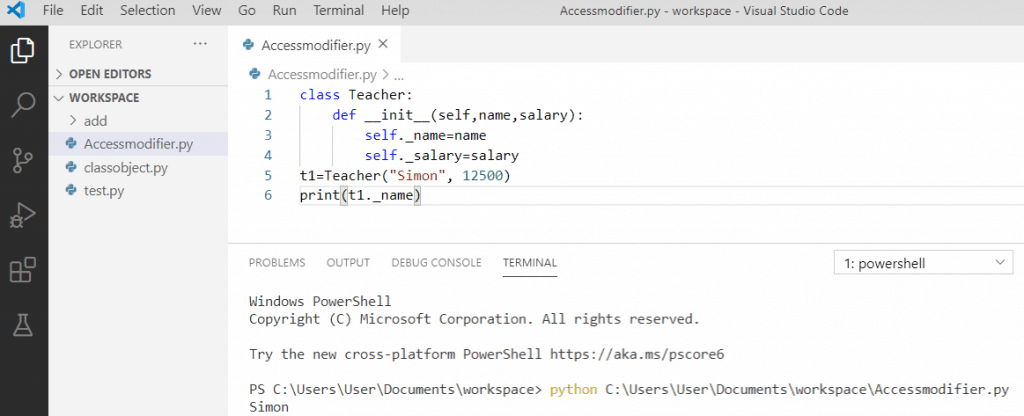
self.\_salary=salary

t1=Teacher("Simon", 12500)

print(t1.\_name)

* After writing the above code (protected access modifier in python), Ones you will print **“t1.\_name”**then the output will appear as **“ Simon ”**.
* Here, we made the class variable **name**and**salary**protected by giving**underscores**. We can access the protected members from the outside class with the help of an object.

You can refer to the below screenshot for protected access modifier in python.

Protected access modifier in python

Python access modifiers program example

Now, let us see an example of **Python access modifiers** with a complete program.

As we have seen all the **Python access modifiers** separately with an example, now let’s combine all the three into the program to see how it works.

**Example:**

class Teacher:

val1 = None

\_val2 = None

\_\_val3 = None

def \_\_init\_\_(self, val1, val2, val3):

self.val1 = val1

self.\_val2 = val2

self.\_\_val3 = val3

def dispPublicMembers(self):

print("This is public member: ", self.val1)

def \_dispProtectedMembers(self):

print("This is protected member: ", self.\_val2)

def \_\_dispPrivateMembers(self):

print("This is private member: ", self.\_\_val3)

def accessPrivateMembers(self):

self.\_\_dispPrivateMembers()

class Child(Teacher):

def \_\_init\_\_(self, val1, val2, val3):

Teacher.\_\_init\_\_(self, val1, val2, val3)

def accessProtectedMembers(self):

self.\_dispProtectedMembers()

obj1 = Child("Hello", "Simon", 100000)

obj1.dispPublicMembers()

obj1.accessProtectedMembers()

obj1.accessPrivateMembers()

* After writing the above code in python, the output will appear as **“ Hello Simon 100000 ”**.
* Here, we have parent class as **“Teacher”**and derived class as **“Child”**, and the private members are accessed by making it public member function **“def accessPrivateMembers”** and it can access private members of the class.
* Also, we have a **“Child”**class and it inherits the properties of the parent class and also it can access the protected member’s function of **“Teacher”** class which is the parent class