

Sample Class in Python

In below example, the class Employee has three instance variables name, age, and salary. The `__init__` method is a special method in Python that is automatically called when an object of the class is created. This method sets the initial values of the instance variables. The `display_employee_info` method prints the values of the instance variables.

```
class Employee:
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary

    def display_employee_info(self):
        print("Name:", self.name)
        print("Age:", self.age)
        print("Salary:", self.salary)
```

To create an object of the class Employee and use its methods, you can do the following:

```
employee = Employee("John Doe", 30, 50000)
employee.display_employee_info()
```

Below is another example

```
class Car:
    def __init__(self, make, model, year):
        self.make = make
        self.model = model
        self.year = year

    def get_make_and_model(self):
        return f'{self.make} {self.model}'

car = Car("Toyota", "Camry", 2020)
print(car.get_make_and_model())
# Output: Toyota Camry
```

Below is more advanced case:

```
class Car:
    def __init__(self, make, model, year, speed=0):
        self.make = make
        self.model = model
        self.year = year
        self._speed = speed

    def __repr__(self):
        return f"{self.year} {self.make} {self.model}"

    def __str__(self):
        return f"{self.year} {self.make} {self.model} (speed: {self._speed} mph)"

    def accelerate(self, delta):
        self._speed += delta

    def brake(self, delta):
        self._speed = max(0, self._speed - delta)

    def speed(self):
        return self._speed

my_car = Car("Toyota", "Camry", 2020)
print(my_car)
my_car.accelerate(20)
print(my_car.speed())
my_car.brake(10)
print(my_car.speed())
```

The result will be like below:

```
2020 Toyota Camry (speed: 0 mph)
20
10
```

Below is another example:

```
class sample(object):

    # Static variable for object number
    objectNo = 0

    def __init__(self, name1):

        # variable to hold name
        self.name = name1

        # Increment static variable for each object
        sample.objectNo = sample.objectNo + 1

        # each object's unique number that can be
        # considered as ID
        self.objNumber = sample.objectNo

    def myFunc(self):
        print("My name is ", self.name,
              "from object ", self.objNumber)

    def alterIt(self, newName):
        self.name = newName

    def myFunc2():
        print("I am not a bound method !!!")

# creating first instance of class sample
samp1 = sample("A")
samp1.myFunc()

# unhide the line below to see the error
# samp1.myFunc2() #-----> error line

# creating second instance of class sample
samp2 = sample("B")
samp2.myFunc()
samp2.alterIt("C")
samp2.myFunc()
samp1.myFunc()
```