

Creating Classes and Instance Objects in Python

Class in Python

A class in Python is like a blueprint or a template for creating objects. Objects are instances of classes that can hold data and have behaviors (functions /methods).

Think of a class like a recipe for making a cake – it tells you what ingredients and steps are needed, but the actual cake is made later (that’s the object).

Why Use Classes?

- To group related data (attributes) and functions (methods) together.
- To model real-world things, concepts, or entities.
- To write reusable and organized code using object-oriented programming (OOP).

Syntax of a Class

To create a class, use the keyword “class”:

```
class ClassName:  
    # class variable  
    # methods (functions inside a class)
```

Example: Creating a Class

```
class Student:  
    # attribute (variable)  
    school = "Abhyas Online"  
  
    # method (function)  
    def greet(self):  
        print("Hello! Welcome to", self.school)
```

Explanation:

- class Student: → defines a class named Student.
 - school → is a class variable (common for all students).
 - greet(self) → is a method (function inside the class).
- The self keyword represents the **current object** calling the method.

Creating Instance Objects

An **object (or instance)** is a **real-world example** created from a class.

When you create an object, you’re using the class’s blueprint to make something real.

We create an object using the class name followed by parentheses:

```
object_name = ClassName()
```

CBSE

ICSE

NTSE

Banking & Insurance

Central Govt. Service

State Govt. Services

LAW Entrance

MBA Entrance

Railways & Metro Services

...many more

abhyasonline.in

Course
&
Test Series

Creating Classes and Instance Objects in Python

Creating a Single Object in a Class

Example:

```
class Student:
    def show(self):
        print("Hello, I am a Student!")
```

Creating ONE object

```
s1 = Student()
s1.show()
```

◆ Output:

Hello, I am a Student!

👉 Here, s1 is the only object created from class Student.

Creating Multiple Objects in a Class

Example:

```
class Car:
    def details(self):
        print("This is a car.")
```

Creating MULTIPLE objects

```
car1 = Car()
car2 = Car()
```

```
car1.details()
car2.details()
```

Output:

This is a car.
This is a car.

👉 Many objects can be created from one class, each working independently.

Example: Creating Objects and Using Them

```
class Student:
    # class variable
    school = "Abhyas Online"

    # method
    def greet(self):
        print("Hello! Welcome to", self.school)
```

creating objects (instances)

-  CBSE
-  ICSE
-  NTSE
-  Banking & Insurance
-  Central Govt. Service
-  State Govt. Services
-  LAW Entrance
-  MBA Entrance
-  Railways & Metro Services
- ...many more

abhyasonline.in

Course
&
Test Series

Creating Classes and Instance Objects in Python

```
student1 = Student()  
student2 = Student()
```

```
# calling the method using objects  
student1.greet()  
student2.greet()
```

Output:

Hello! Welcome to Abhyas Online
Hello! Welcome to Abhyas Online

Explanation:

- student1 and student2 are two **instances** (objects) of the Student class.
- Each object can **access class variables and methods**.
- The self parameter automatically refers to the object that called the method.

Deleting Objects in Python (del keyword)

The del statement in Python is used to remove a reference to an object.

Once deleted:

- You cannot access that object again
- If that was the last reference, Python removes the object from memory automatically

pass Statement in Python

The pass statement is a placeholder in Python.

It is used when a statement is required by Python syntax, but you do not want to write any code yet.

In simple words: "pass tells Python to do nothing."

Where do we use pass?

We use pass when we want to create structure first and add the code later, for example:

- Inside an empty function.
- Inside an empty loop
- Inside an empty class
- Inside conditional blocks (if/else) where no action is needed

Deleting a Single Object

Example:

```
class Student:  
    pass
```

CBSE

ICSE

NTSE

Banking &
Insurance

Central Govt.
Service

State Govt.
Services

LAW
Entrance

MBA
Entrance

Railways & Metro
Services

...many more

abhyasonline.in

Course
&
Test Series

Creating Classes and Instance Objects in Python

```
# Create object  
s1 = Student()
```



CBSE

```
# Delete object  
del s1
```



ICSE

```
# Trying to access deleted object  
print(s1) # ✗ This will give an error
```

Output:

```
NameError: name 's1' is not defined
```



NTSE

Explanation:

- s1 was pointing to a Student object
- del s1 removed that reference
- So Python does not know s1 anymore



Banking &
Insurance

Deleting Multiple Objects

Example:

```
class Car:  
    pass
```



Central Govt.
Service

```
car1 = Car()  
car2 = Car()  
car3 = Car()
```



State Govt.
Services

```
# Deleting multiple references  
del car1, car2
```



LAW
Entrance

```
print(car3) # ✓ Still exists  
print(car1) # ✗ Error
```

Output:

```
<__main__.Car object at 0x000...> # car3 exists  
NameError: name 'car1' is not defined
```



MBA
Entrance

Explanation:

- Only car3 remains because we didn't delete it
- car1 and car2 were deleted together



Railways & Metro
Services

...many more

abhyasonline.in

