

◆ What is OOP?

- A programming paradigm that mimics **real-world objects**
 - Helps organize code using:
 - **Class** (Blueprint)
 - **Object** (Instance)
 - **Attributes** (Data)
 - **Methods** (Behavior)
-

◆ 1. Class and Object

✓ Class

A blueprint to create objects

Example:

```
python
CopyEdit
class House:
    ...
```

✓ Object

An instance of a class

Example:

```
python
CopyEdit
my_house = House("Blue", "Large")
```

◆ 2. The `__init__` Method (Constructor)

- Special method to **initialize attributes**
- Automatically called when object is created

Example:

```
python
CopyEdit
def __init__(self, color, size):
    self.color = color
    self.size = size
```

◆ 3. Attributes

- Characteristics or properties of an object
- Declared inside `__init__` using `self`

Example:

```
python
CopyEdit
self.color = color
self.size = size
```

◆ 4. Methods

- Functions defined **inside a class**
- Describe what an object **can do**

Example:

```
python
CopyEdit
def change_color(self, new_color):
    self.color = new_color
```

🏠 Practical Example: House Class

```
python
CopyEdit
class House:
    def __init__(self, color, size):
        self.color = color
        self.size = size

    def describe(self):
        print(f"The house is {self.color} and {self.size}.")
```

Create Object:

```
python
CopyEdit
my_house = House("Blue", "Large")
my_house.describe()
```

Output:

```
mathematica
CopyEdit
The house is Blue and Large.
```

Changing Attribute with Method

```
python
CopyEdit
my_house.change_color("Red")
print(my_house.color) # Output: Red
```

Real-Life Example: Car Class

```
python
CopyEdit
class Car:
    def __init__(self, color, speed):
        self.color = color
        self.speed = speed

    def accelerate(self):
        self.speed += 10
        print("Speed is now:", self.speed)

python
CopyEdit
my_car = Car("Red", 120)
my_car.accelerate() # Output: Speed is now: 130
```

Summary Table

Concept	Meaning
Class	Blueprint/template for objects
Object	Instance of class with unique attribute data
Attribute	Object's data (e.g., color, size)

Concept

Meaning

Method Object's action/function (e.g., drive())



Final Thoughts

- Use classes to model real-world objects
- Keep attributes in `__init__`
- Use methods to modify or display object behavior
- Instantiate as many objects as needed using the same class