Programming Skills - Python

Learn the basics of Python Programming

Day - 9

Object-Oriented Programming - 2

Agenda

9.1. Brief Recap

A brief recap of what we discussed yesterday.

9.2. Encapsulation

Defining and understanding Encapsulation.

9.3. Inheritance

Defining and understanding Inheritance.

9.4. Polymorphism

Defining and understanding Polymorphism.

Agenda

9.5. Method Overloading

Learn more about methods.

9.6. Method Overriding

Learn and understand what this means.

9.7. Special Methods

Learning with examples.

9.1. Brief Recap

9.2. Encapsulation

Encapsulation

The idea of wrapping data and the methods that work on data within one unit. This restricts access to variables only where necessary.

This is achieved through private attributes and public methods.

9.3. Inheritance

Inheritance

Inheritance is the capability of one class to derive or inherit the properties from another class.

Syntax:

```
Class BaseClass:
    # body
Class DerivedClass(BaseClass):
    # body
```

Single Inheritance

When a child class inherits only from one class, it is called single inheritance.

Syntax:

```
Class BaseClass:
    # body
Class DerivedClass(BaseClass):
    # body
```

Multiple Inheritance

When a child has more than one parent class, it is called multiple inheritance.

Syntax:

```
Class Base1:
  # body of the class

Class Base2:
  # body of the class

Class Derived(Base1, Base2):
  # body of the class
```

Multi-Level Inheritance

When a child class inherits from another child class, it is referred to as multi-level inheritance.

Syntax:

```
Class Base:
  # body of the class

Class Child(Base):
  # body of the class

Class GrandChild(Child):
  # body of the class
```

9.4. Polymorphism

Polymorphism

The word polymorphism means having many forms.
In programming, it means the same function name being used for different types.

Syntax:

```
class Dog:
  def speak(self):
    return "Woof!"

class Cat:
  def speak(self):
    return "Meow!"
```

9.5. Method Overloading

Method Overloading

Two or more methods have the same name but different numbers of parameters or different types of parameters, or both.

```
def product(a, b):
  p = a * b
  print(p)

def product(a, b, c):
  p = a * b * c
  print(p)
```

This is not supported by Python.

9.6. Method Overriding

Method Overriding

When a method in a child class has the same name, the same parameters, and the same return type as a method in its parent-class, then the method in the child is said to **override** the method in the parent.

Example

```
class Bird:
  def fly(self):
    return "Bird flies"

class Penguin(Bird):
  def fly(self):
    return "Penguin can't fly"
```

This is supported by Python.

9.7. Special Methods

Special/Dunder Methods

Dunder - Double Underscore

Special or Dunder methods are methods that start and end with two underscores (__).

These can be used to override default Python operations.

Examples:

```
__str__ → Defines behavior for
of the str() function which is
used to cast other datatypes to
strings.

__len__ → Defines behavior of
the len() function, which is
used to find out the length of
a list.
```

Q&A

Thank You!