

Part-7

Python Classes and Objects Overview

Python is an object-oriented programming language, which means it provides features that support object-oriented programming (OOP) concepts. In Python, everything is an object, and classes are the blueprints for creating these objects. This set of notes will cover the fundamental concepts of classes and objects in Python, including:

- Defining classes and creating objects
- Using the `__init__` method (constructor)
- Accessing class attributes and methods
- Inheritance (single-level, multi-level, hierarchical, and multiple)
- Polymorphism
- Encapsulation
- Abstraction

Creating Lists and Applying Functions

- We can create a list of values, such as `[2, 5, 10, 6, 4, 12]`
- To find the cube of each value in the list, we can use the `map()` function along with a lambda function.
- The `map()` function applies the lambda function to each element in the list and returns a map object.
- The lambda function `pow(x, 3)` raises each element `x` to the power of 3.
- The resulting list of cubes is `[8, 125, 1000, 216, 64, 1728]`

Using the `reduce()` Function

- The `reduce()` function from the `functools` module applies a cumulative calculation to all items in a list.
- To find the sum of all elements in a list, we can use `reduce()` with a lambda function that adds two elements (`x + y`).
- The result of applying `reduce()` to the list `[2, 5, 10, 6, 4, 12]` is 39

- To find the product of all elements in the list, we can use the multiplication operator `*` in the lambda function instead of addition.

Defining a User-Defined Function with a Lambda

- We can create a user-defined function that uses a lambda function to calculate the value of a quadratic equation.
- The user-defined function `quadratic()` takes three parameters: `a`, `b`, and `c`.
- Inside the function, we return a lambda function that calculates the value of the quadratic equation $ax^2 + bx + c$.
- We can then call the `quadratic()` function with specific values for `a`, `b`, and `c`, and pass an `x` value to the returned lambda function to get the result.

Classes in Python

- A class is a virtual entity or blueprint for creating objects.
- A class provides a way to bundle data and functionality together.
- Classes are created using the `class` keyword, followed by the class name and a colon.
- The first string inside the class is called a docstring and provides a brief description of the class.

Creating Objects

- An object is an instance of a class, and the process of creating an object is called instantiation.
- To create an object, you call the class name like a function, using parentheses.
- The object can then access the attributes and methods defined in the class.

The `__init__` Method (Constructor)

- The `__init__` method is similar to a constructor in other programming languages.
- It is automatically executed when a new object is created.
- You can use the `__init__` method to assign values to object properties or perform other necessary operations when the object is created.

Accessing Class Attributes and Methods

- You can access class attributes and methods using the object name followed by a dot and the attribute or method name.
- Class attributes are variables defined within the class, and class methods are functions defined within the class.

- The `self` parameter in class methods refers to the current object instance and is used to access class attributes.

Inheritance

- Inheritance is a mechanism that allows a new class to be based on an existing class.
- The new class inherits the attributes and methods of the existing class, which is called the base or parent class.
- The new class is called the derived or child class.
- Python supports different types of inheritance, including single-level, multi-level, hierarchical, and multiple inheritance.

Single-Level Inheritance

- In single-level inheritance, a child class inherits from a single parent class.

Multi-Level Inheritance

- In multi-level inheritance, a child class inherits from a parent class, and that parent class inherits from a grandparent class.

Hierarchical Inheritance

- In hierarchical inheritance, multiple child classes inherit from a single parent class.

Multiple Inheritance

- In multiple inheritance, a child class inherits from multiple parent classes.

Polymorphism

- Polymorphism is the ability of an object to take on many forms.
- It allows the same method to be used for different types of objects.
- Polymorphism is achieved through method overriding, where a child class provides its own implementation of a method that is already defined in the parent class.

Encapsulation

- Encapsulation is the process of hiding the internal details of an object from the outside world.
- In Python, you can use the `__` (double underscore) prefix to make a variable or method private, which means it can only be accessed within the class.

- You can provide public methods to interact with the private variables, allowing you to control the access and modification of the internal data.

Abstraction

- Abstraction is the process of hiding the implementation details and showing only the essential features of an object.
- Abstract classes in Python are created using the abc (abstract base class) module.
- Abstract classes cannot be instantiated, but they can define abstract methods that must be implemented by the child classes.

Key Concepts

- **Class:** A blueprint or template for creating objects.
- **Object:** An instance of a class.
- **Attribute:** A variable defined within a class.
- **Method:** A function defined within a class.
- **Self:** The first parameter in a class method, referring to the current object instance.
- **Inheritance:** The ability of a new class to be based on an existing class.
- **Polymorphism:** The ability of an object to take on many forms.
- **Encapsulation:** The process of hiding the internal details of an object.
- **Abstraction:** The process of hiding the implementation details and showing only the essential features of an object.

Table of Inheritance Types

Inheritance Type	Description
Single-Level	A child class inherits from a single parent class.
Multi-Level	A child class inherits from a parent class, which inherits from a grandparent class.
Hierarchical	Multiple child classes inherit from a single parent class.
Multiple	A child class inherits from multiple parent classes.