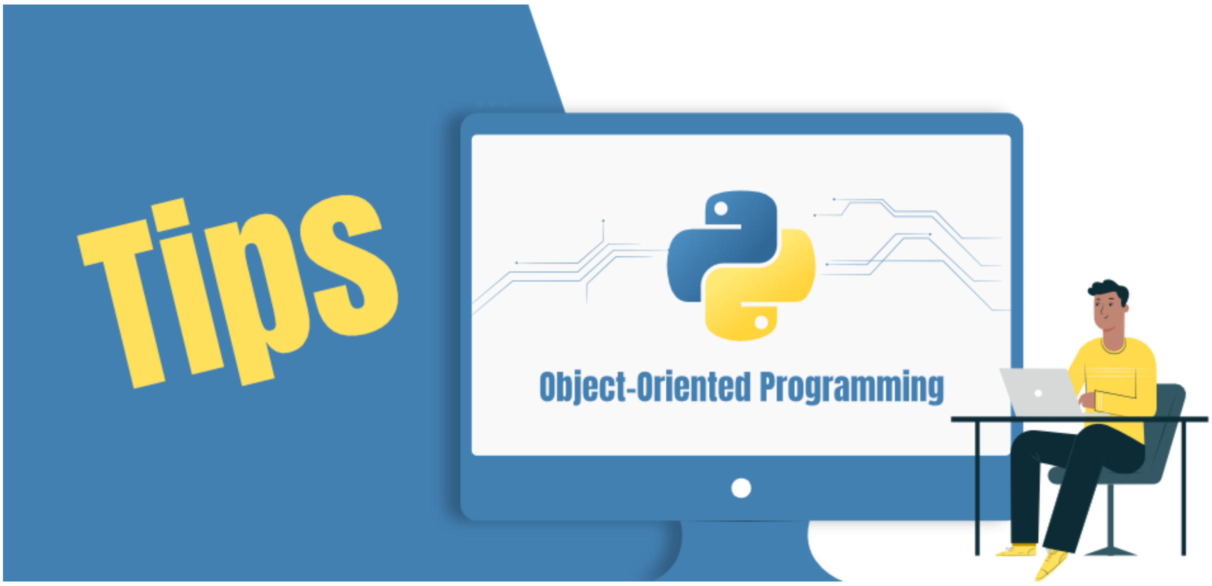
8 Tips For Object-Oriented Programming in Python

OOP or Object-Oriented Programming is a programming paradigm that organizes software design around data or objects and relies on the concept of classes and objects, rather than functions and logic. **Object-oriented programming ensures code reusability and prevents Redundancy**, and hence has become very popular even in the fields outside software engineering like **Machine Learning, Artificial Intelligence, Data Science**, etc. There are many object-oriented programming languages like Java, JavaScript, C++, Python, and many others.



Basics of Object-Oriented Programming in Python

As stated earlier, **OOP is a programming paradigm that uses objects and classes and aims to implement real-world entities in programming.** The main concepts of OOP are stated below:

1. Inheritance: It is the process in which **one class inherits the methods and attributes from another class**. The class whose properties and methods are inherited is known as the Parent class and the class which inherits from the parent class is called the Child class. Inheritance is the most important aspect of object-oriented programming and provides the reusability of the code.

2. Encapsulation: The word, “encapsulate” means to enclose something and the principle of encapsulation works in the same way. In OOP, **the data and functions which operate on that data are wrapped together into a single unit by encapsulation.** By this method, we can hide private details of a class and can only expose the functionality that is needed for interfacing with it.

3. Polymorphism: Polymorphism is taken from the Greek words Poly and morph which means many and shape respectively. Hence in OOP, **Polymorphism refers to the functions having the same names but having different functionalities.** Polymorphism helps in making programming more intuitive and easier.

4. Data Abstraction: Abstraction is another functionality of OOP in which **we hide the internal details or implementations of a function and show the functionalities only**. In other words, the users know “what the function does” but they don’t know “how it does” because they only get to see the basic implementation of the function, whereas the inner working is hidden.

To know more about the 4 pillars of Object-oriented programming click here. In this blog, we will discuss the **8 Tips for Object-Oriented Programming in Python**. So Let’s take a look.

1. Difference of Class and Instance Level Data

To under inheritance, **learning to differentiate between class-level and instance-level data is very crucial**, as inheritance is one of the core concepts of OOP. An instance attribute is a Python variable belonging to one object and is unique to it. It is accessible only in the scope of this object and is defined inside the constructor function of the class. Whereas, a class attribute is unique to a class rather than a particular object and is defined outside the constructor function. While writing a code, Instance-level data should be kept separate from class-level data and should not interfere with it.

2. Using Meaningful Names

One of the best practices to follow while writing Object-Oriented Programming is to use meaningful names keeping in mind that the **naming convention must follow camel casing.** There are rules which apply while naming attributes and functions as well. Hence, we should always name the design in such a way that one class, attribute, or function is responsible for one particular task only.

3. Knowing the Use of Static

Static means that the member is bound to a class level rather than to an instance level. A static method can not access or modify the state of the class. By using static, the method doesn’t need access to the class instance. Also, using **static methods improves code readability and saves a lot of memory**.

4. Deciding Between Internal and Private Modifiers

Access specifiers in Python play an important role in securing data from

a private one a double leading underscore __ is used. This practice helps in determining the access control of a specific data member or a member function of a class and tells the users which parts of the code are not public and should not be relied on.

5. Docstrings in Python

Developers are used to writing comments in the code but comments do not always provide the most structured way of workflows. For this inconvenience, **Python structured documentation or docstrings provide a convenient way of associating documentation with Python public modules, functions, classes, methods**, etc. Unlike source code comments, the docstring describes what the function does and not how. Docstrings in python are declared using ”’triple single quotes”’ or “””triple double quotes”””.

6. pep8 Guidelines

Python Enterprise Proposal also called **PEP** are coding conventions for python packages. Besides writing a code with proper logic, many other important factors affect a code's quality. PEP 8 is a document that was written in 2001 by Guido van Rossum, Barry Warsaw, and Nick Coghlan with its main aim to enhance the readability and consistency of code. One should keep in mind three important pep8 guidelines that are, to ensure that **each line of code is limited to 80 characters**, **all libraries should be imported at the beginning, and to eliminate redundant variables or intermediary variables present in the code**.

7. Setting Access to Attributes

Attributes of a class are function objects that are used to implement access controls of the classes and define corresponding methods of its instances. To access and manipulate the attributes of a class, python has some in-built methods. Those are **getattr()**, **hasattr()**, **setattr()** and **delattr()**. The **getattr()** function helps in accessing the attribute of the object. The **hasattr()** function is used in checking if an attribute exists or not. **Setattr()** is used to set an attribute. In case the attribute does not exist, then one would be created, whereas, **delattr()** is used to delete an attribute.

8. Using Abstract Classes

An abstract class provides a **common interface for different implementations of a component**. In object-oriented programming, developers need to reduce the amount of coding, and hence, abstract classes are used. An abstract class is used as it applies to a wide variety of objects and helps in the creation of a set of methods that must be created within any child classes built from that abstract class.

Last Updated : 08 Mar, 2022 24

Previous

Top 10 Python Applications in Real World

Next

10 Skills to Become a Backend Developer in 2023

Similar Reads

- Object Oriented Programming in Python | Set 2 (Data Hiding and Object Printing)
- A Step-by-Step Approach to Learn Object Oriented Programming
- Top 10 Object-Oriented Programming Languages in 2023
- Python | Matplotlib Sub plotting using object oriented API
- Python | Matplotlib Graph plotting using object oriented API
- Object oriented testing in Python
- ChatGPT Tips & Tricks: 5 Secret Tips That Can Boost Your ChatGPT Game to God Mode
- Characteristics of Good Object Oriented Design
- 6 Steps To Approach Object-Oriented Design Questions in Interview
- Tips to reduce Python object size

Complete Tutorials

- OpenAI Python API - Complete Guide
- ChatGPT Tutorial: ChatGPT-3.5 Guide for Beginners
- Pandas AI: The Generative AI Python Library
- Python for Kids - Fun Tutorial to Learn Python Programming
- Data Analysis Tutorial

Article Contributed By :

priyankab14

priyankab14

Follow

Vote for difficulty

Current difficulty : Easy

Easy | Normal | Medium | Hard | Expert

Article Tags : GBlog , Python

Practice Tags : python

Improve Article

Report Issue

GeeksforGeeks
Sanchhaya Education Private Limited
A-143, 9th Floor, Sovereign Corporate Tower, Sector-136, Noida, Uttar Pradesh - 201305

Company

- About Us
- Legal
- Careers
- In Media
- Contact Us
- Advertise with us
- GFG Corporate Solution
- Placement Training Program

Explore

- Job-A-Thon Hiring Challenge
- Hack-A-Thon
- GfG Weekly Contest
- Offline Classes (Delhi/NCR)
- DSA in JAVA/C++
- Master System Design
- Master CP

Languages

- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

DSA

- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- Top 100 DSA Interview Problems
- DSA Roadmap by Sandeep Jain

Data Science & ML

- Data Science With Python
- Data Science For Beginner
- Machine Learning Tutorial
- Maths For Machine Learning
- Pandas Tutorial

HTML & CSS

- HTML
- CSS
- Bootstrap
- Tailwind CSS
- SASS
- LESS
- NodeJS
- Web Design

- Apply for Mentor

- GeeksforGeeks Videos

- All Cheat Sheets

- NumPy Tutorial
- Deep Learning Tutorial

Computer Science

- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

Python

- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- OpenCV Python Tutorial
- Python Interview Question

DevOps

- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Competitive Programming

- Top DS or Algo for CP
- Top 50 Tree
- Top 50 Graph
- Top 50 Array
- Top 50 String
- Top 50 DP
- Top 15 Websites for CP

System Design

- What is System Design
- Monolithic and Distributed SD
- High Level Design or HLD
- Low Level Design or LLD
- Crack System Design Round
- System Design Interview Questions
- Grokking Modern System Design

JavaScript

- TypeScript
- ReactJS
- NextJS
- AngularJS
- NodeJS
- Express.js
- Lodash
- Web Browser

TextBook Solutions/Study Material

- NCERT Solutions for Class 12
- NCERT Solution for Class 11
- NCERT Solutions for Class 10
- NCERT Solutions for Class 9
- NCERT Solutions for Class 8
- Complete Study Material

School Subjects

- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Commerce

- Accountancy
- Business Studies
- Indian Economics
- Macroeconomics
- Microeconimics
- Statistics for Economics

Management & Finance

- Management
- HR Managament
- Income Tax
- Finance
- Economics

SSC/ BANKING

- SSC CGL Syllabus
- SBI PO Syllabus
- SBI Clerk Syllabus
- IBPS PO Syllabus
- IBPS Clerk Syllabus
- SSC CGL Practice Papers

Colleges

- Indian Colleges Admission & Campus Experiences
- Top Engineering Colleges
- Top BCA Colleges
- Top MBA Colleges
- Top Architecture College
- Choose College For Graduation

Companies

Preparation Corner

Exams

More Tutorials

Write & Earn

IT Companies

Software Development Companies

Artificial Intelligence(AI) Companies

CyberSecurity Companies

Service Based Companies

Product Based Companies

PSUs for CS Engineers

Company Wise Preparation

Preparation for SDE

Experienced Interviews

Internship Interviews

Competitive Programming

Aptitude Preparation

Puzzles

JEE Mains

JEE Advanced

GATE CS

NEET

UGC NET

CAT

Software Testing

Software Development

Product Management

SAP

SEO

Linux

Excel

Write an Article

Improve an Article

Pick Topics to Write

Share your Experiences

Internships