



**Department of Computer Science and Engineering
PES University, Bangalore, India**

**Lecture Notes
Python for Computational Problem Solving
UE23CS151A**

**Lecture #98
*Introduction to Object Oriented Programming***

**By,
Prof. Sindhu R Pai,
Anchor, PCPS - 2023
Assistant Professor
Dept. of CSE, PESU
&
Dr. Ramya C
Associate Professor
Dept. of CSE, PESU**

**Many Thanks to
Dr. Shylaja S S (Director, CCBD and CDSAML Research Centers, Former
Chairperson, CSE, PES University)
Prof. Chitra G M (Asst. Prof, Dept. of CSE, PCPS Anchor – 2022)**

Introduction

Three main streams under OO philosophy -> A) **Object Oriented Analysis**

B) **Object Oriented Design**

C) **Object Oriented Programming**

A) and B) together dealt in detail in OOAD or OOMD course. In this semester, emphasis is more on **OOP - Object Oriented Programming**

OOP is style of programming in which the **main focus is on the data and the operations that manipulates the data**. Data are organized into **classes, attributes and methods**. OOP is mainly useful to develop big and complex projects carried out by large teams consisting of many developers. Some of the Object Oriented Programming languages are **Java, C#, C++, Python** etc. In Object Oriented approach, both the data and the behavior(operations) are grouped together into classes.

In OOP, **objects are used to interact with real world entities**. Objects are created from templates called "classes", which define the attributes/properties and behavior of the objects they create. OOP allows you to create reusable code and model real-world concepts more closely, making it a **popular choice for many software projects**.

For example, an object could represent a person with **properties** like a name, age, and address and **behaviors** such as walking, talking, breathing, and running. It could represent an email with properties like a recipient list, subject, and body and behaviors like adding attachments and sending. Object-oriented programming is an **approach for modeling concrete, real-world things**, like cars, as well as relations between things, like companies and employees or students and teachers. OOP models real-world entities as software objects that have some data associated with them and can perform certain operations.

The key takeaway is that objects are at the center of object-oriented programming in Python. In other programming paradigms, objects only represent the data. In OOP, they additionally inform the overall structure of the program.

Features of OOP:

OO philosophy emphasizes on below **principles (features)**:

- View everything as an object.
- **Encapsulation:** Binding of data and procedure as a single unit
- **Data Hiding:** Who can access the data. Implemented using access specifiers
- **Abstraction:** The way you view an object

Example: student, library, books

Depending on the application, abstraction has to be implemented.

- **Polymorphism:** poly ... many and morph ... forms

2+3 ... addition,

"2"+"3" ... concatenation,

Based on the data, operator works.

- **Inheritance:** Obtaining or acquiring the property of parent object

Generalization to Specialization

Benefits: Reusability, Information will not be cluttered

Example: Student is a person, Rose is a flower. Tiger is an Animal etc..

Collaboration or Containership/Composition: Object contains another object.

Example: Student has a date_of_birth where in date_of_birth is an object with three attributes day, month, year.

More on classes and objects will be discussed in the next lecture notes (#99)

-END-