Introduction to OOPs

- Languages like Pascal, C, FORTRAN, and COBOL are called procedure-oriented programming languages. Since in these languages, a programmer uses procedures or functions to perform a task. When the programmer wants to write a program, he will first divide the task into separate sub tasks, each of which is expressed as functions/ procedures. This approach is called ***procedure-oriented approach.***
- The languages like C++ and Java use classes and object in their programs and are called Object Oriented Programming languages. The main task is divided into several modules and these are represented as classes. Each class can perform some tasks for which several methods are written in a class. This approach is called "***Object Oriented approach***".

OOP Vs POP

	OOP	POP
Definition	OOP stands for Object-oriented programming it focuses on data rather than the algorithm.	POP stands for Procedure-oriented programming, focuses on procedural abstractions.
Programs	Divided into small chunks called objects which are instances of classes.	Divided into small parts based on the functions.
Accessing Mode	Four accessing modes are used in OOP to access attributes or methods – ‘Private’, ‘Public’, ‘default’ , and ‘Protected’	No such accessing mode is required
Execution	Various methods can work simultaneously	Follows a systematic step-by-step approach to execute functions.
Security	It is of high secure because of it data hiding feature	There is no such way of data hiding in POP, thus making it less secure.

Features of OOP

- There are mainly ‘4’ features of OOP’s are there, which are listed below.

1. Encapsulation
2. Inheritance
3. Polymorphism
4. Abstraction

Understanding Encapsulation:

- Generally there are two forms of encapsulation in OOP.

First Form:

- Encapsulation is a technique that packages related data and behaviors into a single unit.
- Here, the common characteristics and behaviors of a student are packaged into a single unit: the Studentclass.
- This is the process of encapsulation.
- Encapsulation hides implementation details of the Student class from other objects.

Second Form:

- Encapsulation is the technique of making the fields in a class private and providing access to the fields via methods.
- If a field is declared private, it cannot be accessed by anyone outside the class.
- Such that we provide security to the data from outside world without misusing it, which is commonly known as ‘Information Hiding’ or ‘Data Hiding’.
- In process of Information Hiding, the other objects cannot access the data directly. Instead, they have to invoke the getters which are designed to protect the data from misuse or unwanted changes.

What is the need for Encapsulation?

- **Flexibility**: It’s more flexible and easier to change the encapsulated code with new requirements.
- **Reusability**: Encapsulated code can be reused throughout the application or across multiple applications.
- **Maintainability**: If an application is encapsulated in separate units (classes, interfaces, methods, setters, getters, etc) then it’s easy to change or update a part of the application without affecting other parts, which reduces the time of maintenance.