Course Name - Object Oriented Programming using Java

Lecture 4– Properties of OOP- message passing, inheritance, encapsulation, polymorphism, Data abstraction.

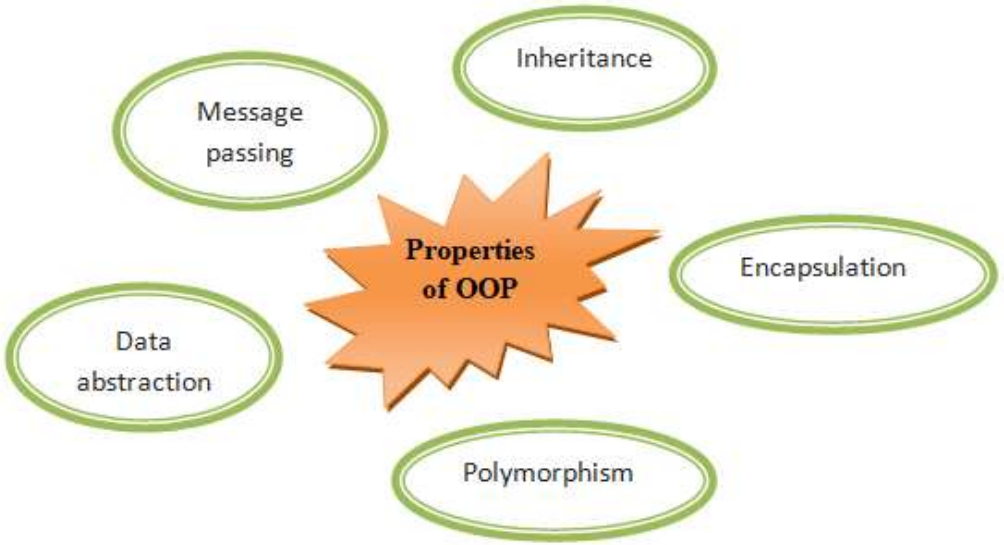
Presented By
Dr. Sudipta Sahana
Associate Professor
Dept. of CSE
UEM - Kolkata

Topic of Interest

- **Properties of OOP**
- **Message passing**
- **Inheritance**
- **Encapsulation**
- **Polymorphism**
- **Data Abstraction**

Properties of OOP

In order for a programming language to be object-oriented, it has to enable working with classes and objects as well as the implementation and use of the fundamental object-oriented principles and concepts: message passing, inheritance, encapsulation, and polymorphism and data abstraction. Let's summarize each of these fundamental principles of OOP:



Message passing

In object oriented languages, you can consider a running program under execution as a pool of objects where objects are created for ‘interaction’ and later destroyed when their job is over. This interaction is based on ‘messages’ which are sent from one object to another asking the recipient object to apply one of its own methods on itself and hence, forcing a change in its state. The objects are made to communicate or interact with each other with the help of a mechanism called message passing.

Inheritance

The process by which one class acquires the properties(data members) and functionalities(methods) of another class is called **inheritance**. The aim of inheritance is to provide the **reusability of code** so that a class has to write only the unique features and rest of the common properties and functionalities can be extended from the another class.

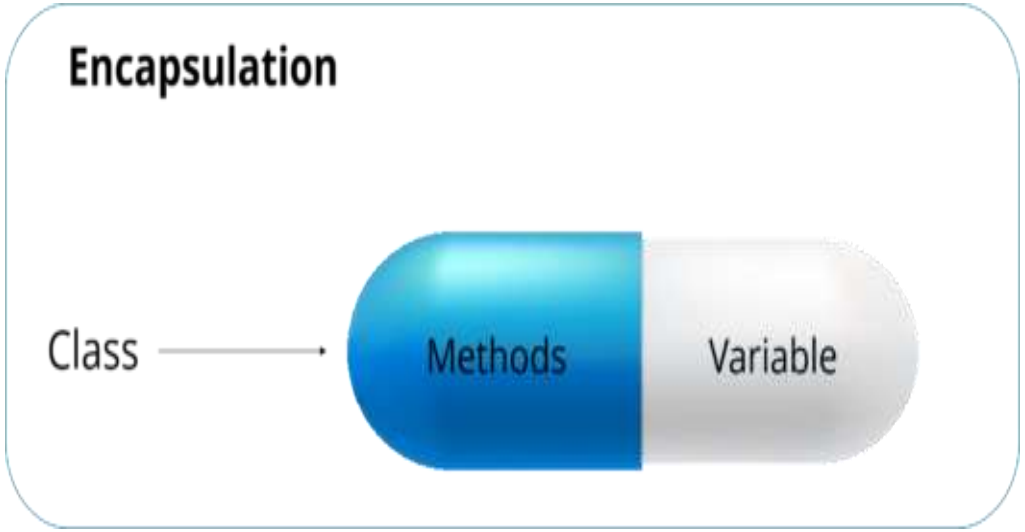
Child Class:
The class that extends the features of another class is known as child class, sub class or derived class.

Parent Class:
The class whose properties and functionalities are used(inherited) by another class is known as parent class, super class or Base class.

Encapsulation

Hiding internal state and requiring all interaction to be performed through an object's methods is known as data encapsulation — a fundamental principle of object-oriented programming. The encapsulation is the process of grouping or wrapping up of data and functions to perform actions on the data into the single unit. The single unit is called a class. Encapsulation is like enclosing in a capsule. That is enclosing the related operations and data related to an object into that object. It keeps the data and the code safe from external interference.

The main purpose or use of the encapsulation is to provide the security to the data of a class.

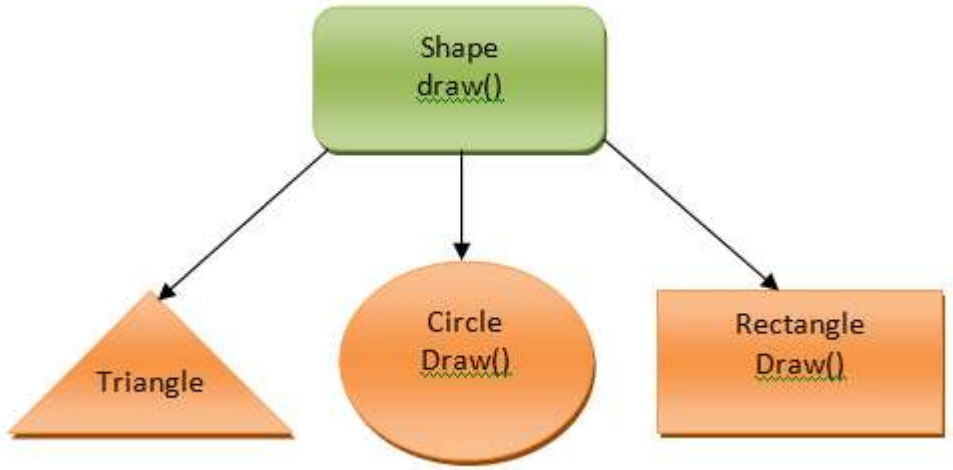


Real-Life Example of Encapsulation

Polymorphism

Polymorphism means taking many forms, where ‘poly’ means many and ‘morph’ means forms. It is the ability of a variable, function or object to take on multiple forms. In other words, polymorphism allows you define one interface or method and have multiple implementations.

Let’s understand this by taking a real-life example and how this concept fits into Object oriented programming.



Data Abstraction

Data abstraction refers to the quality of dealing with ideas rather than events. It basically deals with hiding the details and showing the essential things to the user. If you look at the image here, whenever we get a call, we get an option to either pick it up or just reject it. But in reality, there is a lot of code that runs in the background. So you don't know the internal processing of how a call is generated, that's the beauty of abstraction. Therefore, abstraction helps to reduce complexity.

UEM
UNIVERSITY OF ENGINEERING & MANAGEMENT
Good Education, Good Jobs

Thank You