

Object Oriented Programming using Java 1

Object Oriented Paradigm

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University

Structured Programming

- It aims at improving the clarity, quality, and development time of a computer program by making extensive use of the structured control flow constructs of selection (if/then/else) and repetition (while and for), block structures, and subroutines
- It states that three ways of combining programs - sequencing, selection, and iteration - are sufficient to express any computable function

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects
Class
Data Abstraction
Encapsulation
Inheritance
Polymorphism

Applications of OOP

Procedure Oriented Programming

Procedure Oriented Programming

- It can be derived from structured programming, based upon the concept of calling procedure
- Procedures (or routines), subroutines or functions, simply consists of a series of computational steps to be carried out
- During a program's execution, any given procedure might be called at any point, including by other procedures or itself
- Ex: FORTRAN, COBOL, BASIC, Pascal, C
- In the Procedure oriented approach, the problem is viewed as a sequence of things to be done such as reading, calculating, and printing
- A number of functions are written to accomplish the task
- The primary focus is on function
- The Typical structure of POP (Procedure Oriented Programming) is like hierarchical structure

Characteristics of Procedure Oriented Programming

- Procedure Oriented Programming follows a top-down approach
- Functions are more important than data in a program
- Emphasis is on doing things (algorithms)
- Large programs are divided into smaller programs known as functions
- Most of the functions share global data
- Data moves openly around the system from function to function
- Functions transform data from one form to another

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Advantages

- It promotes a modular approach to software development
- They are known for their efficiency in terms of memory usage and execution speed
- In a multi-function program, many important data items are placed as global so that they may be accessed by all the functions. Each function may have its local data

Disadvantages

- They can sometimes lack code reusability
- As programs grow in size and complexity, managing and understanding procedural-based code can be challenging
- They have limited support for abstraction
- Encapsulation is not well supported in procedural-based languages
- Global data are more vulnerable to an inadvertent change by a function

Chittaranjan Pradhan

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Object Oriented Programming

- Object-oriented programming (OOP) is a programming paradigm based on the concept of **objects**, which can contain data and code: **data** in the form of fields (often known as attributes or properties), and **code** in the form of procedures (often known as methods)
- In OOP, programmers define not only the data type of a data structure, but also the types of operations (functions) that can be applied to the data structure
- The major motivating factor in the invention of Object oriented approach is to remove some of flaws encountered in Procedure Oriented Programming
- OOP treats data as a critical element in the program development and does not allow it to flow freely around the system. It ties data more closely to the functions that operate on it, and protects it from accidental modification from outside functions
- OOP allows decomposition of a problem into number of entities called objects and then builds data and functions

Chittaranjan Pradhan

Structure Oriented Programming

Procedure Oriented Programming

Object Oriented Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Characteristics of Object Oriented Programming

- Emphasis is on data rather than procedure
- Programs are divided into what are known as objects
- Data structures are designed such that they characterize the objects
- Functions that operate on the data of an object are tied together in the data structure
- Data is hidden and cannot be accessed by external functions
- Object may communicate with each other through functions
- New data and functions can be easily added whenever necessary
- Follows Bottom-up approach in program design
- *OOP is an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand*

Chittaranjan Pradhan

Structure Oriented Programming

Procedure Oriented Programming

Object Oriented Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Advantages

- OOP divides complex systems into smaller components, making it easier to create and maintain
- Inheritance allows code reuse in saving time
- OOP protects data integrity and privacy by restricting direct access
- OOP encourages the development of reusable code elements, saving time and improving system reliability
- OOP models real-world systems

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Disadvantages

- Increased complexity
- Overhead in multi-class projects
- Difficulty in testing
- Performance overhead

POP vs. OOP

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects
Class
Data Abstraction
Encapsulation
Inheritance
Polymorphism

Applications of OOP

Procedure oriented programming (POP)	Object oriented programming (OOP)
Program is divided into functions	Program is divided into objects
Top-down approach	Bottom-up approach
Inheritance is not allowed	Inheritance property is used
It doesn't use access specifier	It uses access specifier
No data hiding	Encapsulation is used to hide the data
Parts of program are linked through parameter passing	Object functions are linked through message passing
Adding new data and functions is not easy	Adding new data and functions is easy
No code reusability	The existing code can be reused
Not suitable for solving big problems	Used for solving bigger problems
Ex: C, FORTRAN	Ex: Java, Python

Objects

- **Objects** are basic run-time entities in an object-oriented system
- An object has an identity, state, and behavior. Ex: a chair, pen, table, keyboard
 - **State**: represented by the attributes of an object. It also reflects the properties of an object
 - **Behavior**: represented by the methods of an object. It also reflects the response of an object to other objects
 - **Identity**: unique name given to an object that enables it to interact with other objects
- When a program is executed the objects interact by sending messages to one another
- Object take up space in memory and have an associated address like a record in pascal or structure or union in C
- Each object contains data and code to manipulate the data

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Class

Class

- **Class** is a user-defined blueprint or prototype from which objects are created
- It represents the set of properties or methods that are common to all objects of one type
- It consists of data members and member functions, which can be accessed and used by creating an instance of that class
- Using classes, you can create multiple objects with the same behavior instead of writing their code multiple times
- When a class is defined, no memory is allocated but when it is instantiated (i.e. object is created) memory is allocated
- A class may be thought of as a 'data type' and an object as a 'variable' of that data type. Each object is associated with the data of type class with which they are created
- A class is a collection of object of similar type
- Ex: Mango, Apple and Orange are members of class Fruit

Data Abstraction

- **Data abstraction** refers to providing only essential information about the data to the outside world, hiding the background details or implementation
- Data abstraction may also be defined as the process of identifying only the required characteristics of an object, ignoring the irrelevant details
- Ex: Consider a real-life example of a man driving a car. The man only knows that pressing the accelerators will increase the car speed or applying brakes will stop the car, but he does not know how on pressing the accelerator, the speed is actually increasing. He does not know about the inner mechanism of the car or the implementation of the accelerators, brakes etc. in the car. This is what abstraction is

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Encapsulation

- The wrapping up of data and functions into a single unit (called class) is known as **encapsulation**
- It is defined as the wrapping up of data under a single unit. It is the mechanism that binds together the code and the data it manipulates
- It is a protective shield that prevents the data from being accessed by the code outside this shield
- Abstraction refers to the act of representing essential features without including the background details or explanations

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Inheritance

- The capability of a class to derive properties and characteristics from another class is called **Inheritance**
- Inheritance allows the user to reuse the code whenever possible and reduce its redundancy
 - The class whose features are inherited is known as superclass
 - The class that inherits the other class is known as subclass. The subclass can add its own fields and methods in addition to the superclass fields and methods

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Polymorphism

Polymorphism

- **Polymorphism** means ability to take more than one form. Polymorphism is a Greek term, means the ability to take more than one form
- An operation may exhibit different behaviors in different instances. The behavior depends upon the types of data used in the operation
- **Method Overloading:** Also, known as compile-time polymorphism, is the concept of Polymorphism where more than one method share the same name with different signature(Parameters) in a class
- **Method Overriding:** Also, known as run-time polymorphism, is the concept of Polymorphism where method in the child class has the same name, return-type and parameters as in parent class. The child class provides the implementation in the method already written

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP

Applications of OOP

- Real-Time System Design
- Simulation and Modelling System
- Object-Oriented Databases
- Hypertext And Hypermedia
- AI and Expert Systems
- Neural Networking and Parallel Programming
- Office Automation Systems
- CIM/CAD/CAM Systems

Structure Oriented
Programming

Procedure Oriented
Programming

Object Oriented
Programming

Objects

Class

Data Abstraction

Encapsulation

Inheritance

Polymorphism

Applications of OOP