

Principles of Object-Oriented Programming

Features of Procedure-Oriented programming

- The procedural programming focuses on **processing of instructions (Functions)** in order to perform a desired computation.
- The **top-down approach** is used for execution.
- Program consist of **different functions**.
- Most of the functions share **global data**.
- Functions **transform data** from one form to another.
- Data is **not hidden** and can be **easily** shared.

Features of Object-Oriented programming

- Focus is on the **data** rather than procedure or function.
- Program consists of **different objects**.
- **Data structures** are designed to characterize the **objects**.
- Data is **hidden** and can only be accessed through the **object's member functions**.
- Objects can **pass messages** to each other through **functions**.
- **New data and functions** can be easily added whenever necessary.
- It follows **bottom-top approach**.

Difference between Procedural and Object-Oriented Programming

Procedural Oriented Programming	Object Oriented Programming
Focus of Procedural Programming is on Procedures (Function).	Focus of Object-Oriented Programming is on data .
Large programs are divided into smaller programs known as function .	Programs are divided into objects .
Data is not hidden and can be accessed by external functions.	Data is hidden and can not be accessed by external functions.
Follow top-down approach in program design.	Follow bottom-up approach in program design.
Data may communicate with each other through functions.	Objects may communicate with each other through functions.

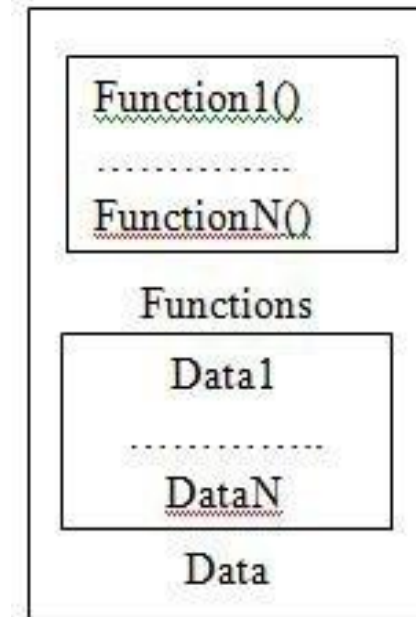
Basic Concepts of OOP's

1. Class
2. Objects
3. Data abstraction
4. Data encapsulation
5. Inheritance
6. Polymorphism

Class

- **Class** is a user defined data types, which consists of **data members and member functions** of an object.
- Classes are declared by using the keyword **class** followed by **class name**.
- Once class is defined, any **number of objects** of that class is created.
- For example : Snacks is a class and wafers, chips, popcorns etc. are the objects.
- Syntax:

```
class class_name  
{  
    Data Members;  
    Member Functions;  
};
```



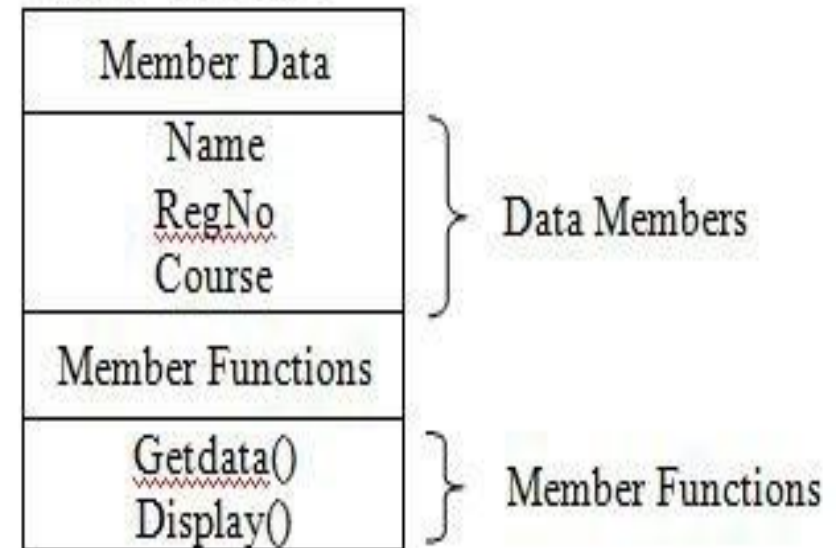
Objects

- Objects are **basic building blocks** for designing programs.
- Objects are **instances of a class** i.e. **multiple objects** can be created for a class which are of same type.
- Each object is identified by a **unique name**.
- For example : Fruit is a class and apple, orange, mango etc. are the objects.
- Syntax:

```
class class_name
{
    Data Members;
    Member Functions;
}object1;
class_name object2;
```

Prepared by Deepali Sonawane, MIT-WPU, Pune

Object - Student

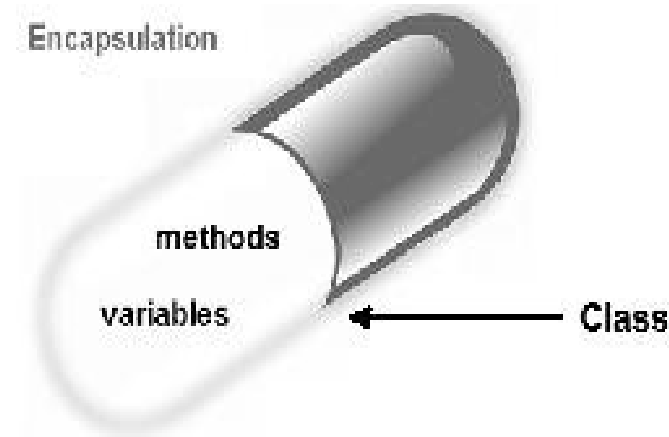


Data Abstraction

- **Data Abstraction** refers to the process of **representing essential features** without including background details or explanations.

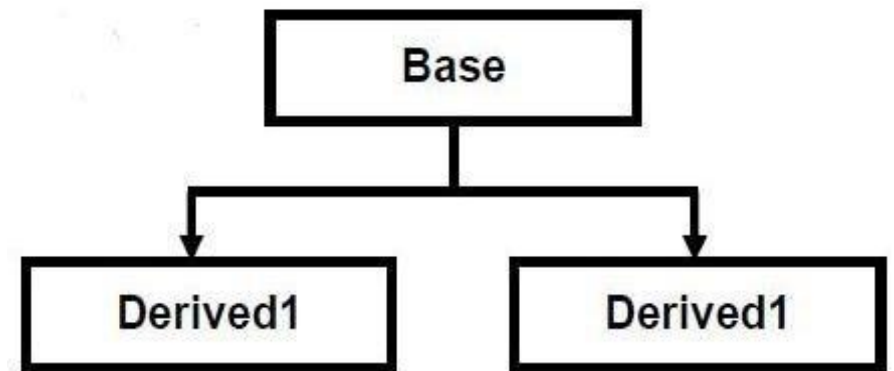
Data Encapsulation

The **binding of data and functions** into a single unit (class) is called data **encapsulation**.



Inheritance

- **Inheritance** is the process by which one object can acquire and use the properties of another object.
- It is the ability to derive a **new class** from an existing one.
- The existing class is known as *base class or super class or parent class*.
- The new class is known as *derived class or sub class or child class*.
- The derived class shares some of the properties of the base class. Therefore, a code from a base class can be **reused by** a derived class.



Polymorphism

- Poly – many and morphism-Forms
- It is used to express the fact that the **same message** can be sent to **many different objects** and interpreted in different ways by each object.
- Overloading allows objects to have **different meaning** depending upon context.
- There are two types of overloading
 - Operator Overloading
 - Function Overloading
- When an existing operator operates on new data type is called **operator overloading**.
- **Function overloading** means two or more function have **same name**, but differ in the **number of arguments** or **data type** of arguments.

Benefits of OOP

- The programs are modularized based on the principles of **classes** and **objects**.
- Linking code & object allows related objects to share common code. This **reduces code duplication** and **code reusability**.
- Creation and implementation of OOP code is **easy and reduces** software development time.
- The concept of **data abstraction** separates object specification and object implementation.
- **Data encapsulated** along with functions. Therefore, external non - member function cannot access or modify data, thus proving **data security**.
- Easier to develop **complex software**, because complexity can be minimized through **inheritance**.

Object Oriented Lanaguage

- Object Oriented programming is the **principle of design and development** of programs using **modular approach**.
- Object Oriented programming approach provides advantages in **creation and development of software** for real life application.
- The basic element of object-oriented programming is **data**.
- The programs are built by combining **data and functions** that operate on the data.
- Some of the OOP's languages are CPP, JAVA,C#, Pearl, Python.

Applications of OOP

- CAD/CAM software
- Object-oriented databases
- User-Interface design such as windows
- Real-time systems like Air Traffic Control Systems, Networked Multimedia Systems etc.
- Simulation and Modeling
- Artificial intelligence and expert systems
- Client-Server Systems
- Office automation System

Thank you