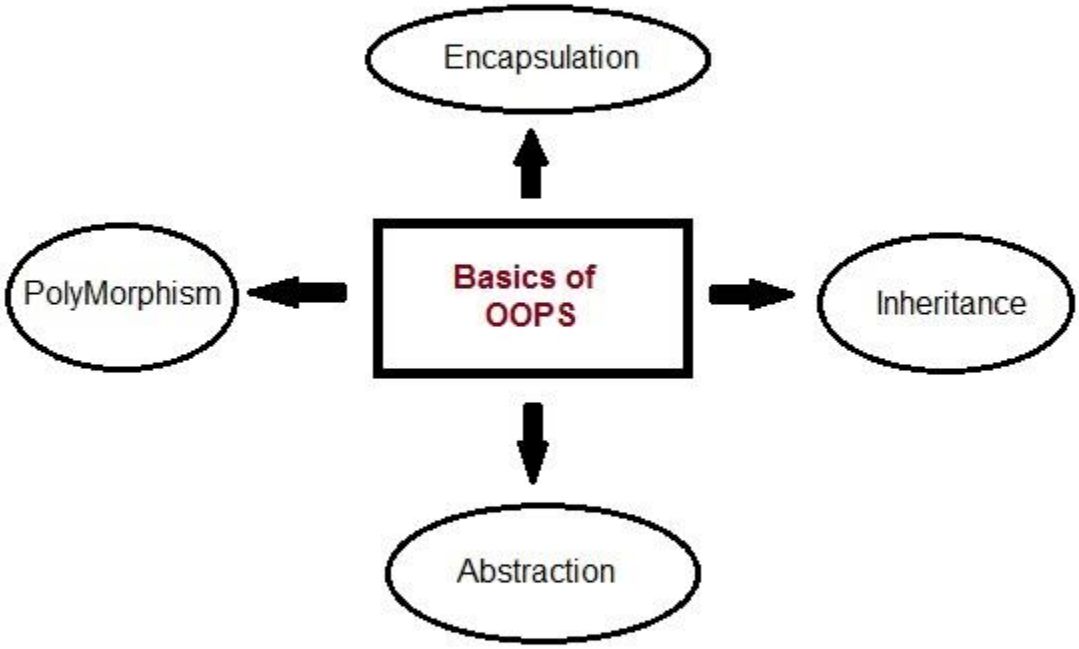
# Object Oriented Programming Important Concepts

Object Oriented programming is a programming style that is associated with the concept of OBJECTS, having data fields and related member functions.

Objects are instances of classes and are used to interact amongst each other to create applications. Instance means, the object of class on which we are currently working. C++ can be said to be as C language with classes. In C++ everything revolves around object of class, which have their methods & data members.

C++ can be said to be as C language with classes. In C++ everything revolves around object of class, which have their methods & data members.



For Example: We consider human body as a class, we do have multiple objects of this class, with variable as color, hair etc. and methods as walking, speaking etc.

Now, let us discuss some of the main features of object oriented programming which you will be using in C++.

Objects, Classes, Abstraction, Encapsulation, Inheritance, Overloading, Exception Handling

**Objects:**
Objects are the basic unit of OOP. They are instances of class, which have data members and uses various member functions to perform tasks.

**Class:**
It is similar to structures in C language. Class can also be defined as user defined data type but it also contains functions in it. So, class is basically a blueprint for object. It declares & defines what data variables the object will have and what operations can be performed on the class's object.

**Abstraction:**
Abstraction refers to showing only the essential features of the application and hiding the details. In C++ classes provide methods to the outside world to access & use the data variables, but the variables are hidden from direct access.

**Encapsulation:**
It can also be said data binding. Encapsulation is all about binding the data variables and functions together in class.

**Inheritance:**
Inheritance is a way to reuse once written code again and again. The class which is inherited is called base class & the class inherits is called derived class. So when, a derived class inherits a base class, the derived class can use all the functions which are defined in base class, hence making code reusable.

**Polymorphism:**
Polymorphism makes the code more readable. It is a feature, which lets is create functions with same name but different arguments, which will perform differently. That is function with same name, functioning in different.

**Overloading:**
Overloading is a part of polymorphism. Where a function or operator is made & defined many times, to perform different functions they are said to be overloaded.

**Exception Handling:**
Exception Handling is a feature of OOP, to handle unresolved exceptions or errors produced at runtime.

# Object-Oriented Programming Important Terms

## What Is an Object?

An object is a software bundle of related state and behavior. Software objects are often used to model the real-world objects that you find in everyday life. This lesson explains how state and behavior are represented within an object, introduces the concept of data encapsulation, and explains the benefits of designing your software in this manner.

## What Is a Class?

A class is a blueprint or prototype from which objects are created. This section defines a class that models the state and behavior of a real-world object. It intentionally focuses on the basics, showing how even a simple class can cleanly model state and behavior.

## What Is Inheritance?

Inheritance provides a powerful and natural mechanism for organizing and structuring your software. This section explains how classes inherit state and behavior from their super classes, and explains how to derive one class from another using the simple syntax provided by the Java programming language.

## What Is an Interface?

An interface is a contract between a class and the outside world. When a class implements an interface, it promises to provide the behavior published by that interface. This section defines a simple interface and explains the necessary changes for any class that implements it.

## What Is a Package?

A package is a namespace for organizing classes and interfaces in a logical manner. Placing your code into packages makes large software projects easier to manage. This section explains why this is useful, and introduces you to the Application Programming Interface (API) provided by the Java platform.