**Object Oriented Programming**

Object Oriented programming is a programming style that is associated with the concept of Class, Objects and various other concepts revolving around these two, like Inheritance, Polymorphism, Abstraction, Encapsulation etc.

In the video below, we have explained the basic concepts of Object Oriented Programming with help of a very easy to understand example. If you want to skip the video, everything is covered below as well.

Let us try to understand a little about all these, through a simple example. Human Beings are living forms, broadly categorized into two types, Male and Female. Right? Its true. Every Human being(Male or Female) has two legs, two hands, two eyes, one nose, one heart etc. There are body parts that are common for Male and Female, but then there are some specific body parts, present in a Male which are not present in a Female, and some body parts present in Female but not in Males.

All Human Beings walk, eat, see, talk, hear etc. Now again, both Male and Female, performs some common functions, but there are some specifics to both, which is not valid for the other. For example : A Female can give birth, while a Male cannot, so this is only for the Female.

Human Anatomy is interesting, isn't it? But let's see how all this is related to C++ and OOPS. Here we will try to explain all the OOPS concepts through this example and later we will have the technical definitons for all this.

**Class**

Here we can take **Human Being** as a class. A class is a blueprint for any functional entity which defines its properties and its functions. Like Human Being, having body parts, and performing various actions.

**Inheritance**

Considering HumanBeing a class, which has properties like hands, legs, eyes etc, and functions like walk, talk, eat, see etc. Male and Female are also classes, but most of the properties and functions are included in HumanBeing, hence they can inherit everything from class HumanBeing using the concept of **Inheritance**.

**Objects**

My name is Abhishek, and I am an **instance/object** of class Male. When we say, Human Being, Male or Female, we just mean a kind, you, your friend, me we are the forms of these classes. We have a physical existence while a class is just a logical definition. We are the objects.

**Abstraction**

Abstraction means, showcasing only the required things to the outside world while hiding the details. Continuing our example, **Human Being's** can talk, walk, hear, eat, but the details are hidden from the outside world. We can take our skin as the Abstraction factor in our case, hiding the inside mechanism.

**Encapsulation**

This concept is a little tricky to explain with our example. Our Legs are binded to help us walk. Our hands, help us hold things. This binding of the properties to functions is called Encapsulation.

**Polymorphism**

Polymorphism is a concept, which allows us to redefine the way something works, by either changing how it is done or by changing the parts using which it is done. Both the ways have different terms for them.

If we walk using our hands, and not legs, here we will change the parts used to perform something. Hence this is called **Overloading**.

And if there is a defined way of walking, but I wish to walk differently, but using my legs, like everyone else. Then I can walk like I want, this will be called as **Overriding**.

**OOPS Concept Definitions**

Now, let us discuss some of the main features of Object Oriented Programming which you will be using in C++(technically).

1. Objects
2. Classes
3. Abstraction
4. Encapsulation
5. Inheritance
6. Overloading
7. 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 declare & 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 can provide methods to the outside world to access & use the data variables, keeping the variables hidden from direct access, or classes can even declare everything accessible to everyone, or maybe just to the classes inheriting it. This can be done using access specifiers.

**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 the **Base** class & the class which inherits is called the **Derived** class. They are also called parent and child 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**

It is a feature, which lets us create functions with same name but different arguments, which will perform different actions. That means, functions with same name, but functioning in different ways. Or, it also allows us to redefine a function to provide it with a completely new definition. You will learn how to do this in details soon in coming lessons.

**Exception Handling**

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

### OOPS Concept With Real Life Example

### OOPS Concept With Real Life Example

This is the most asked Question in a technical interview in any domain.. OOPs Concept is very very important.. Today I will explain OOPs concept with real Life Example that will help you to grasp the concept well and excel in the interviews...

**Objects:**Object is the basic unit of object-oriented programming.Objects are identified by its unique name. An objectrepresents a particular instance of a class. There can be more than one instance of an object. Each instance of an object can hold its own relevant data.

An Object is a collection of data members and associated member functions also known as methods.

**Classes:** Classes are data types based on which objects are created.Objects with similar properties and methods are grouped together to form a Class. Thus a Class represent a set of individual objects. Characteristics of an object are represented in a class as Properties. The actions that can be performed by objects becomes functions of the class and is referred to as Methods.

**Example #1:**

For example consider we have a Class of Cars under which Santro Xing, Alto and WaganR represents individual Objects.In this context each Car Object will have its own, Model,Year of Manufacture, Colour, Top Speed, Engine Power etc.,which form Properties of the Car class and the associated actions i.e., object functions like Start, Move, Stop form the Methods of Car Class.No memory is allocated when a class is created. Memory is  
allocated only when an object is created, i.e., when an instance of a class is created.

**Example #2:**

An architect will have the blueprints for a house....those blueprints will be plans that explain exactly what properties the house will have and how they are all layed out.  However it is just the blueprint, you can't live in it.  Builders will look at the blueprints and use those blueprints to make a physical house.  They can use the same blueprint to make as many houses as they want....each house will have the same layout and properties.  Each house can accommodate it's own families...so one house might have the Smiths live in it, one house might have the Jones live in it.  
  
**The blueprint is the class...the house is the object.  The people living in the house are data stored in the object's properties.**  
  
**Abstraction:** Abstraction means showing essential features and hiding non-essential features to the user.  
  
**For Eg.  Yahoo Mail...**  
  
When you provide the user name and password and click on submit button..It will show Compose,Inbox,Outbox,Sentmails...so and so when you click on compose it will open...but user doesn't  
know what are the actions performed internally....It just Opens....that is essential; User doesn't know internal actions ...that is non-essential things...  
  
**For Eg. Tv Remote..**  
Remote is a interface between user and tv..right. which has buttons like 0 to 10 ,on /of etc but we dont know circuits inside remote...User does not  need to know..Just he is using essential thing that is remote.

**Encapsulation:** Encapsulation means which binds the data and code (or) writing operations and methods in single unit (class).

For Example:  
A car is having multiple parts..like steering,wheels,engine...etc..which binds together to form a single object that is car. So, Here multiple parts of cars encapsulates itself together to form a single object that is Car.  
  
In real time we are using Encapsulation for security purpose...

**Encapsulation = Abstraction + Data Hiding.**

**Inheritance:** Deriving a new class from the existing class,is called Inheritance.  
Derived(sub class) class is getting all the features from Existing (super class\base class) class and also incorporating some new features to the sub class.

For Eg.,

class Address  
{  
String name;  
Srting H.no;  
String Street name;  
}  
class LatestAddress extends Address  
{  
String City;  
String State;  
String Country;  
}  
public class Vishal  
{  
{  
LatestAddress la = new LatestAddress();  
//Assign variable accordingly...  
}  
}

In the above Example class LatestAddress getting all features from the Address class.  
In the LatestAddress class we have total 6 properties..3 are inherited from Address class and 3 properties are  
incorporated. So In the class Vishal we are declaring the object of class LatestAddress and then assign new variables using the properties of the previous base classes... So this is a nice example of inheritance..

**Polymorphism :**

Polymorphism means ability to take more than one form that an operation can exhibit different behavior at different instance depend upon the data passed in the operation.

1>We behave differently in front of elders, and friends. A single person is behaving differently at different time.

2> A software engineer can perform different task at different instance of time depending on the task assigned  to him .He can done coding , testing , analysis and designing depending on the task assign and the requirement.

3> Consider the stadium of common wealth games. Single stadium but it perform multiple task like swimming, lawn tennis etc.

4> If a girl is married and mother of 2 children doing teaching job then  she is a women first ,, teacher in a school when she is in school,,wife of someone at home,, mother of her children,, and obvious daughter of someone & may be girl friend of someone (just kidding) means a woman plays diffent roles at different times dats the polymorphism (many forms).

### Summary:

OOPs have following features:

1. Object             - Instance of Class  
2. Class               - Blue print of Object   
3. Encapsulation    - Protecting our Data  
4. Polymorphism   - Different behaviors at different instances  
5. Abstraction        - Hiding our irrelevant Data  
6. Inheritence        - One property of object is acquiring to another property of object