

OOPS

Abstraction: Hiding complex things behind the procedure which makes those things look simple.

Eg: Button click e photo getting captured

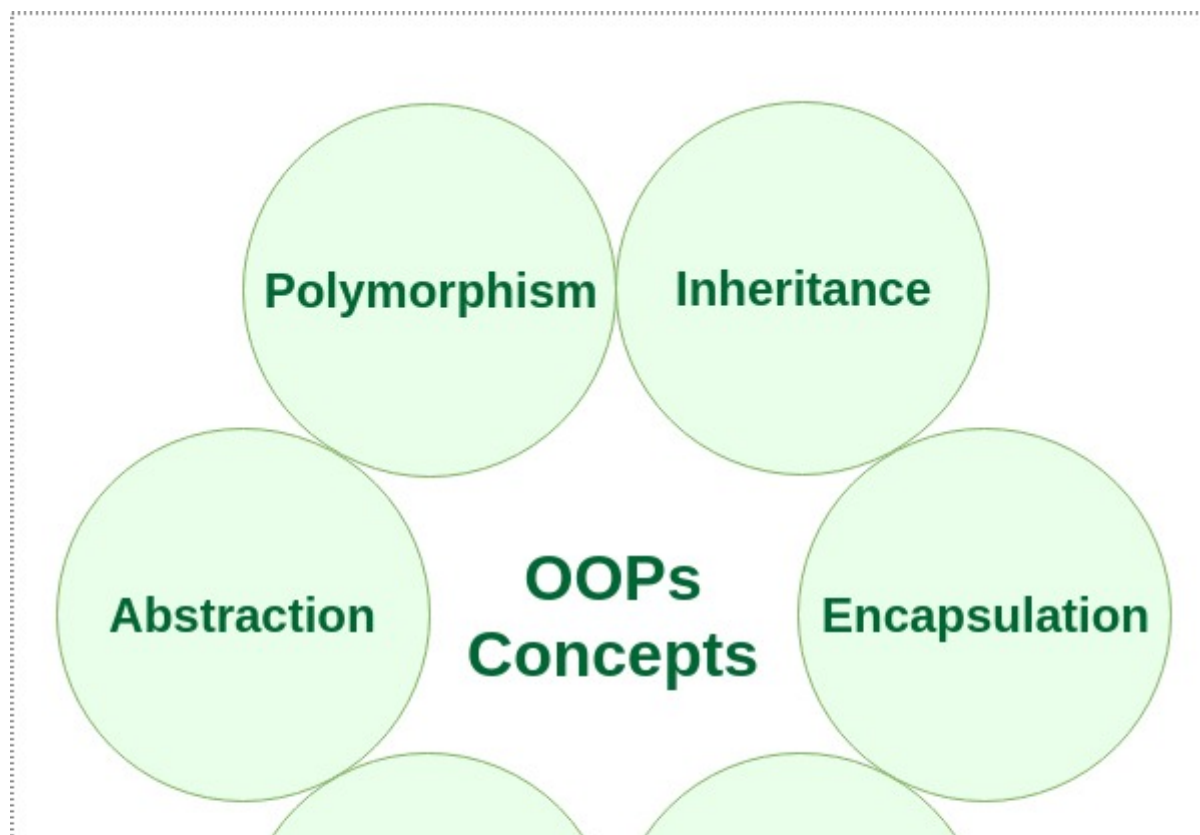
```
class AbstractEmployee{
virtual void AskForPromotion() =0 ;
}
```

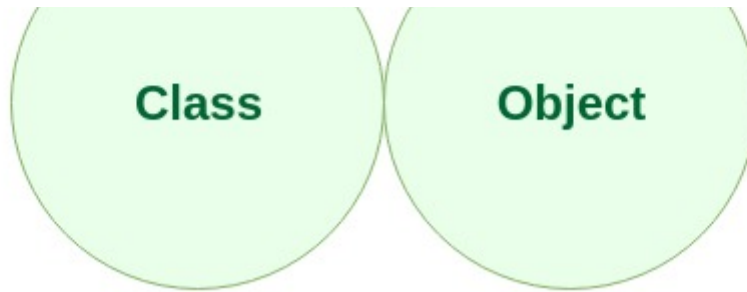
In programming, we cannot have variables, functions, etc with the same name. So to avoid those conflicts we use **namespaces**.

using namespace std means we use the namespace named std. "std" is an abbreviation for standard. So that means we use all the things with in "std" namespace. If we don't want to use this line of code, we can use the things in this namespace like this. `std::cout`, `std::endl`.

why you need both the header file and the namespace to run a simple c++ program, because computer needs to know the definition of the code of the functionalities. It is defined in the header file. So header file needs to be included. namespace is needed because if a functionalities like `cout` is used, but not defined in the current scope computer needs to know where to check. so namespace needs to be included. Because we are writing the code outside the std namespace.

The **main aim of OOP** is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.





Encapsulation is defined as binding together the data and the functions that manipulate them. Encapsulation also leads to data abstraction or hiding. As using encapsulation also hides the data.

Eg: Sales report cannot be accessed by developers. To access it they need to contact sales employees. Sales data and employees who can manipulate them are encapsulated.

Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

Eg: One click to capture photo, internally what happens we don't know

Common forms of Abstraction:

Abstraction using Classes

Abstraction using Header files

Difference between encapsulation and abstraction is in abstraction although the internal code base or data is hidden but we can access it from outside but in-case of encapsulation we always need member function's help to access.

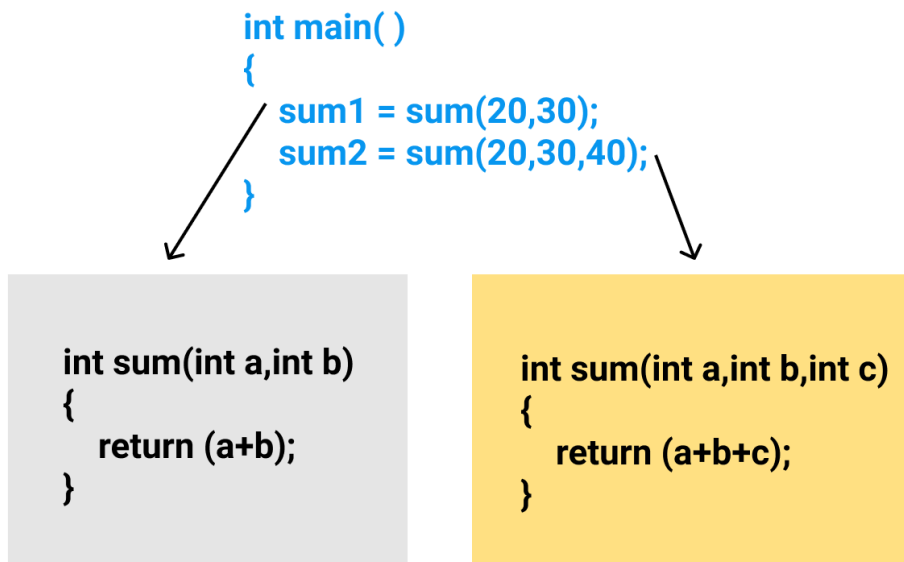
Advantages of Data Abstraction:

- Helps the user to avoid writing the low level code
- Avoids code duplication and increases reusability.
- Can change internal implementation of class independently without affecting the user.
- Helps to increase security of an application or program as only important details are provided to the user.

In simple words, we can define **polymorphism** as the ability of a message to be displayed in more than one form. An operation may exhibit different behaviours in different instances. The behaviour depends upon the types of data used in the operation.

C++ supports operator overloading and function overloading.

- **Operator Overloading:** The process of making an operator to exhibit different behaviours in different instances is known as operator overloading.
 - **Function Overloading:** Function overloading is using a single function name to perform different types of tasks.
- Polymorphism is extensively used in implementing inheritance.



The capability of a class to derive properties and characteristics from another class is called **Inheritance**.

Dynamic Binding: In dynamic binding, the code to be executed in response to function call is decided at runtime. C++ has **virtual functions** to support this.

Message Passing: Objects communicate with one another by sending and receiving information to each other.

