

OOPS CONCEPT



PRESENTED BY-
ANJALI SONA
DEPARTMENT OF COMPUTER SCIENCE



INTRODUCTION

➤ As the name suggests, Object-Oriented Programming or OOPs refers to languages that use objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism, etc in programming. 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.

➤ **OOPs Concepts:**

Class

Objects

Data Abstraction

Encapsulation

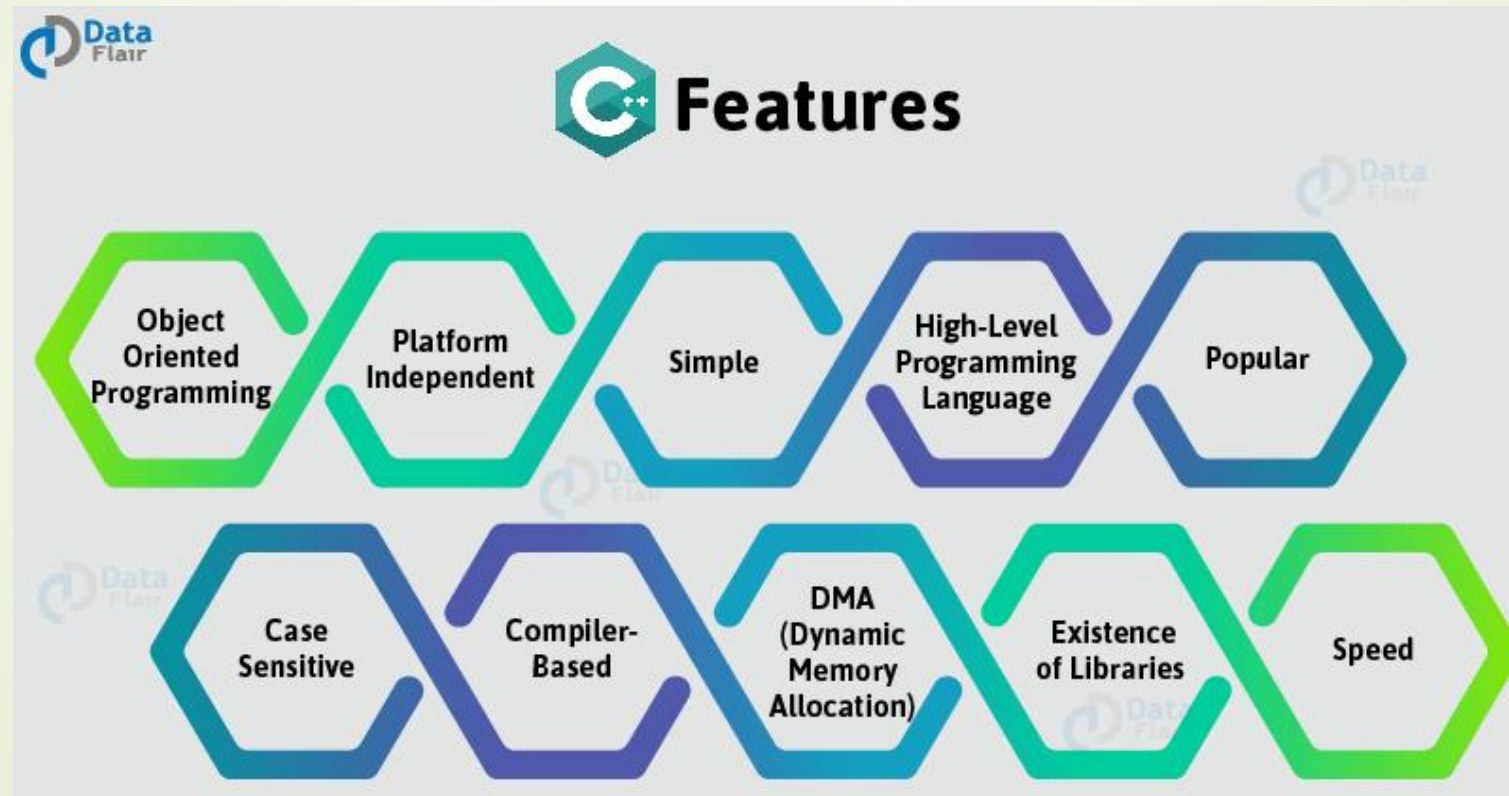
Inheritance

Polymorphism

Dynamic Binding

Message Passing

FEATURES OF OOPS



DIFFERENCE BETWEEN OOP AND POP

OOP	POP
<p>programs are divided into parts known as objects</p> <p>2. The main focus of oop is on the data</p> <p>3.It follows bottom-up approach</p> <p>4.It has access specifiers such as public,private etc</p> <p>5.It provides data hiding,data associated with the program.so security is provide</p> <p>6.Ease of Modification</p> <p>7.Examples:Java,Perl</p>	<p>1.In this programs are divided into functions</p> <p>2.The main focus of POP is on the procedures</p> <p>3. It follows top-down approach</p> <p>4.In this most functions use global data</p> <p>5.POP does not have any access specifiers</p> <p>6.POP does not provide any data security</p> <p>7.Modification is difficult.</p> <p>8.Examples: C,COBOL etc..</p>

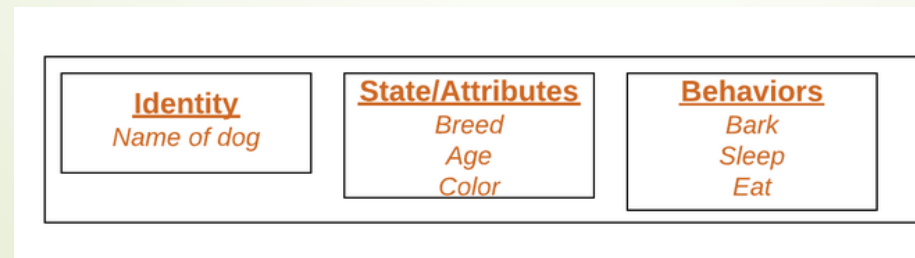


1. Class:

- ▶ A class is a user-defined data type. It consists of data members and member functions, which can be accessed and used by creating an instance of that class. It represents the set of properties or methods that are common to all objects of one type. A class is like a blueprint for an object.
- ▶ **For Example:** Consider the Class of Cars. There may be many cars with different names and brands but all of them will share some common properties like all of them will have 4 wheels, Speed Limit, Mileage range, etc. So here, Car is the class, and wheels, speed limits, mileage are their properties.

2. Object:

- ▶ It is a basic unit of Object-Oriented Programming and represents the real-life entities. An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated. An object has an identity, state, and behavior. Each object contains data and code to manipulate the data. Objects can interact without having to know details of each other's data or code, it is sufficient to know the type of message accepted and type of response returned by the objects.
- ▶ For example "Dog" is a real-life Object, which has some characteristics like color, Breed, Bark, Sleep, and Eats.



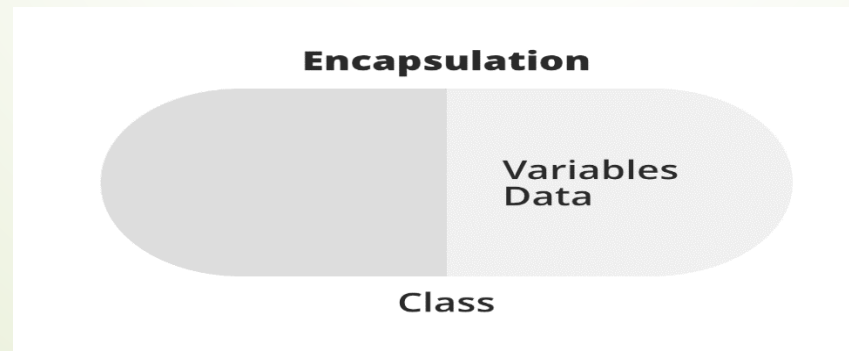


3. Data Abstraction:

- Data abstraction is one of the most essential and important features of object-oriented programming. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation. Consider a real-life example of a man driving a car. The man only knows that pressing the accelerators will increase the speed of the car or applying brakes will stop the car, but he does not know about how on pressing the accelerator the speed is increasing, he does not know about the inner mechanism of the car or the implementation of the accelerator, brakes, etc in the car. This is what abstraction is.

4. Encapsulation

- Encapsulation is defined as the wrapping up of data under a single unit. It is the mechanism that binds together code and the data it manipulates. In Encapsulation, the variables or data of a class are hidden from any other class and can be accessed only through any member function of their class in which they are declared. As in encapsulation, the data in a class is hidden from other classes, so it is also known as **data-hiding**.





Encapsulation

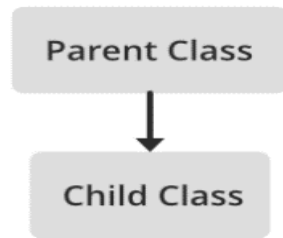
- Consider a real-life example of encapsulation, in a company, there are different sections like the accounts section, finance section, sales section, etc. The finance section handles all the financial transactions and keeps records of all the data related to finance. Similarly, the sales section handles all the sales-related activities and keeps records of all the sales. Now there may arise a situation when for some reason an official from the finance section needs all the data about sales in a particular month. In this case, he is not allowed to directly access the data of the sales section. He will first have to contact some other officer in the sales section and then request him to give the particular data. This is what encapsulation is. Here the data of the sales section and the employees that can manipulate them are wrapped under a single name “sales section”.



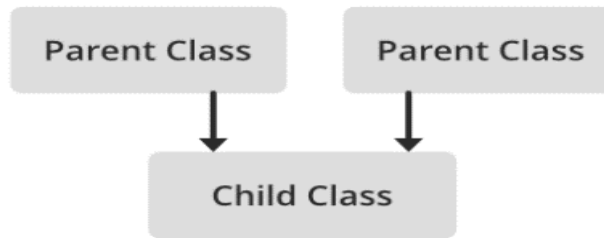
5. Inheritance:

- Inheritance is an important pillar of OOP(Object-Oriented Programming). The capability of a class to derive properties and characteristics from another class is called Inheritance. When we write a class, we inherit properties from other classes. So when we create a class, we do not need to write all the properties and functions again and again, as these can be inherited from another class that possesses it. Inheritance allows the user to reuse the code whenever possible and reduce its redundancy.

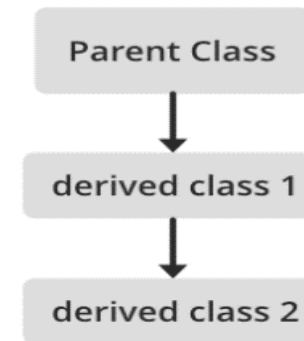
TYPES OF INHERITANCE



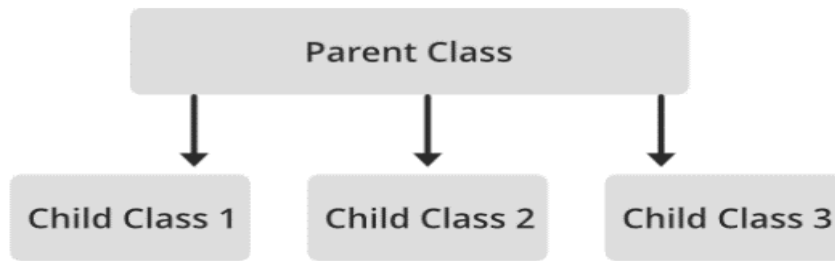
Single inheritance



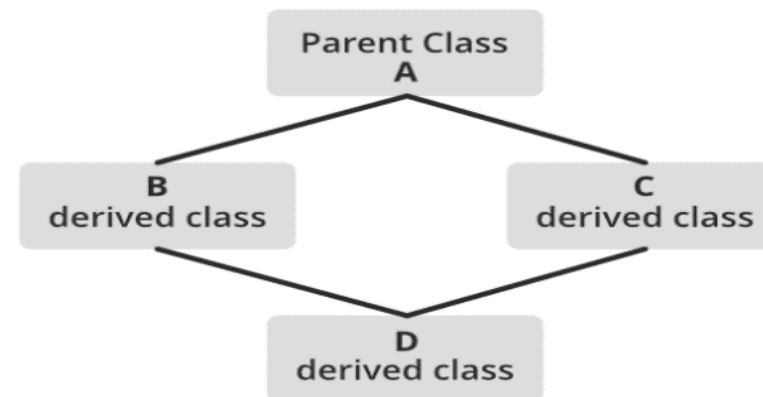
Multiple inheritance



Multilevel inheritance



Hierarchical inheritance





6. Polymorphism:

- ▶ The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form. For example, A person at the same time can have different characteristics. Like a man at the same time is a father, a husband, an employee. So the same person possesses different behavior in different situations. This is called polymorphism.



7. 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.



8.Message Passing:

- Objects communicate with one another by sending and receiving information to each other. A message for an object is a request for execution of a procedure and therefore will invoke a function in the receiving object that generates the desired results. Message passing involves specifying the name of the object, the name of the function and the information to be sent.



Thank you