MCA-102: Object Oriented programming Concepts

Why do we need object oriented programming?

Object oriented programming was discovered because limitations were found in earlier procedural programming .

Drawbacks

- Every function has complete access to global data, the new programmer can easily corrupt the data.
- We can access the data of one function from other since there is no protection.
- In large program it is very difficult to identify what data is being used by the function.
- If a new data is to be added, all functions need to be modified
- Does not model real world problem very well

Procedural Programming Language

Division into Functions:

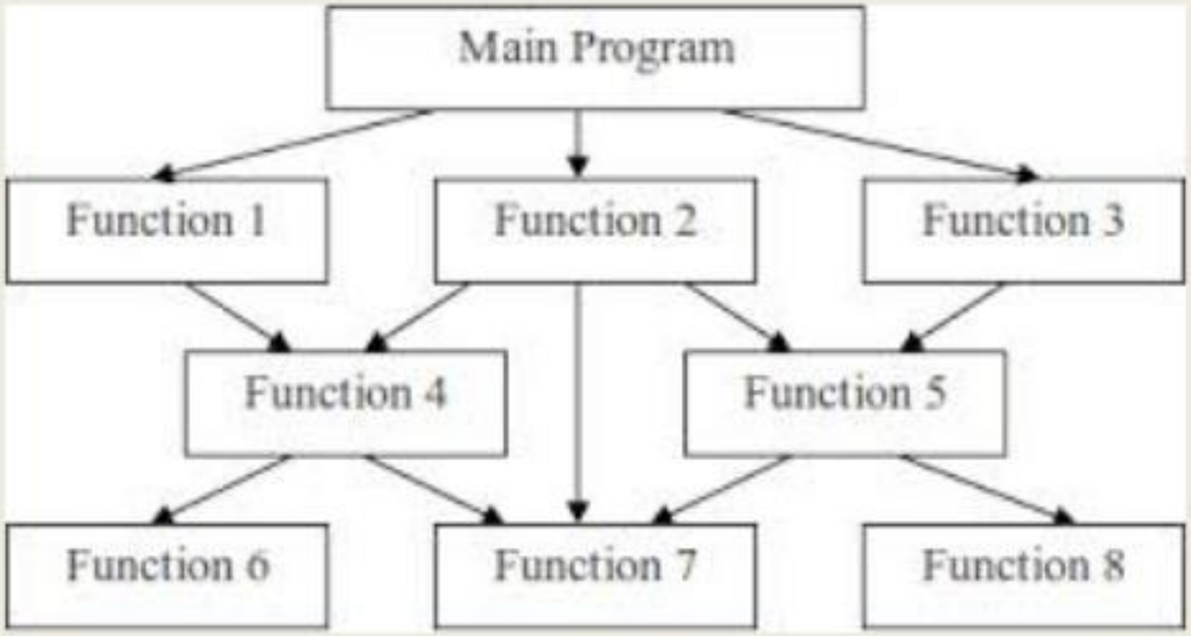
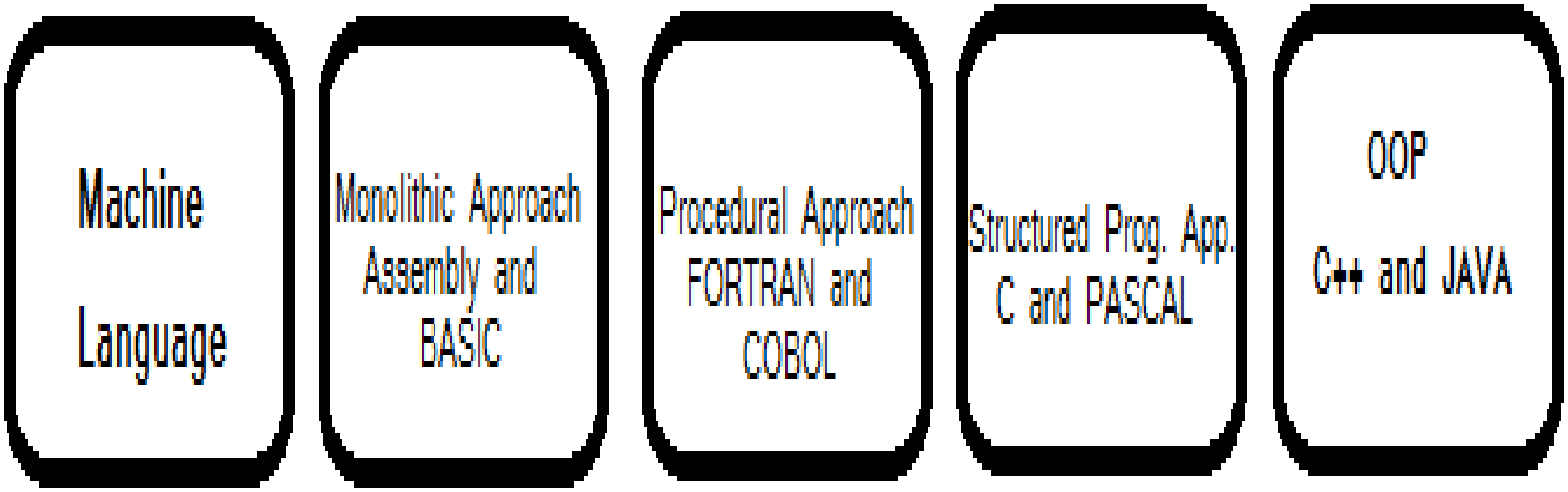


Fig 1: Structure of Procedure-Oriented Programming

Evolution of oops

- ➢Initially for designing small and simple programs, the **machine language** was used.
- ➢Next came the **Assembly Language** which was used for designing larger programs.
- ➢Both machine and Assembly languages are machine dependent. Next came **Procedural Programming Approach** which enabled us to write larger and hundred lines of code.
- ➢Then in 1970, a new programming approach called **Structured Programming Approach** was developed for designing medium sized programs.
- ➢In 1980's the size of programs kept increasing so a new approach known as **OOP** was invented.

Evolution of oops


Machine Language
Monolithic Approach Assembly and BASIC
Procedural Approach FORTRAN and COBOL
Structured Prog. App. C and PASCAL
OOP C++ and JAVA

Object Oriented Programming

- Oops was introduced to overcome the flaws of procedural programming language such as lack of reusability and maintainability.

- Fundamental idea behind oops language is to combine data and functions that operate on that data into a single unit and that unit is known as object.

In other words,

- Object oriented programming is a way of solving complex problems by breaking them into smaller problems using objects.

- In Object oriented programming we write programs using classes and objects utilising features of OOPs such as **abstraction**, **encapsulation**, **inheritance** and **polymorphism**

Object Oriented Programming

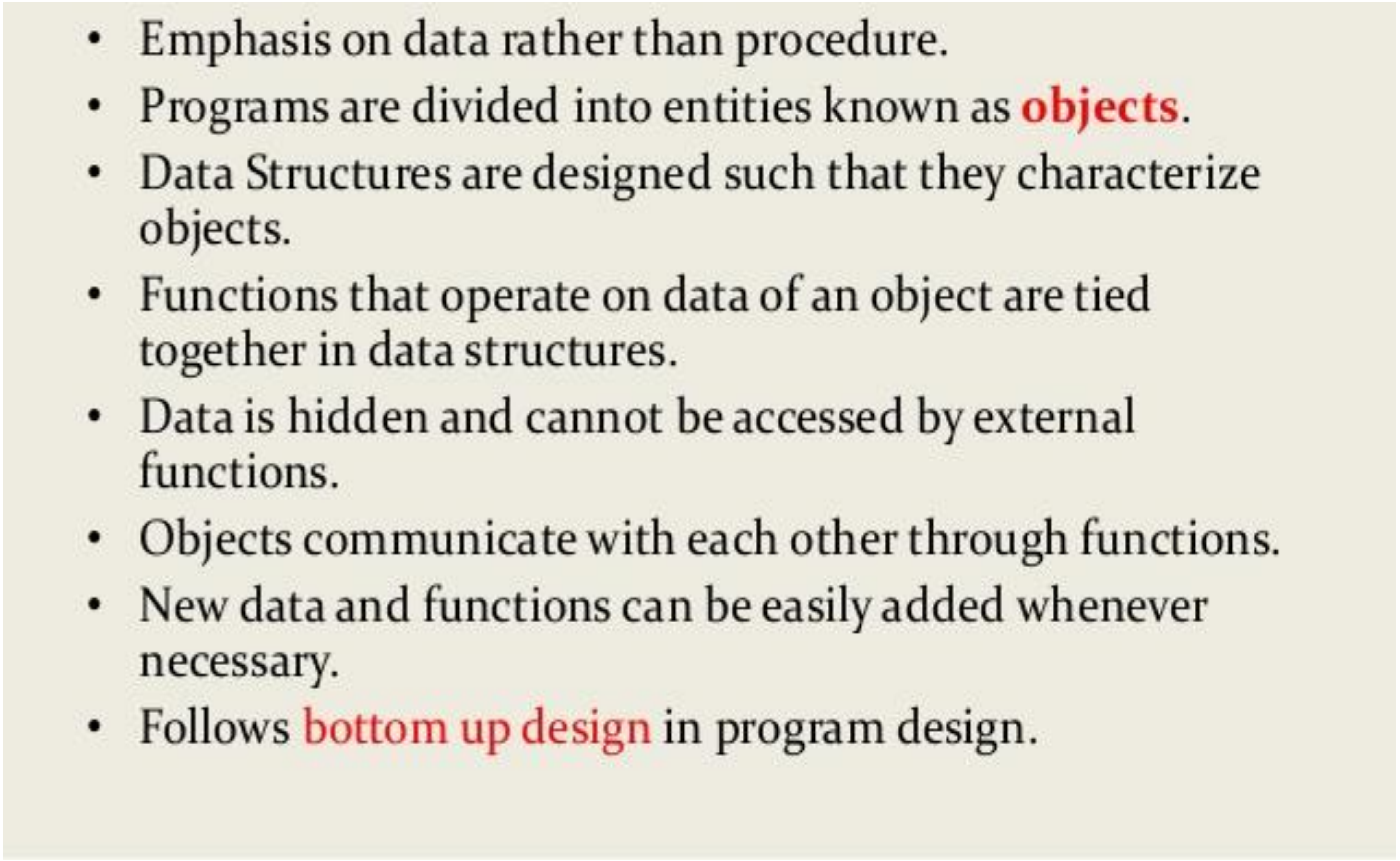
- ➢In oops problem is divided into number of entities called objects and then build data and functions around these objects.

- ➢It ties data more closely to the functions and avoid any modification.

- ➢Data of an object can be accessed by that function only associated with that object.

- ➢Communication among objects is done through objects.

Advantages of oops



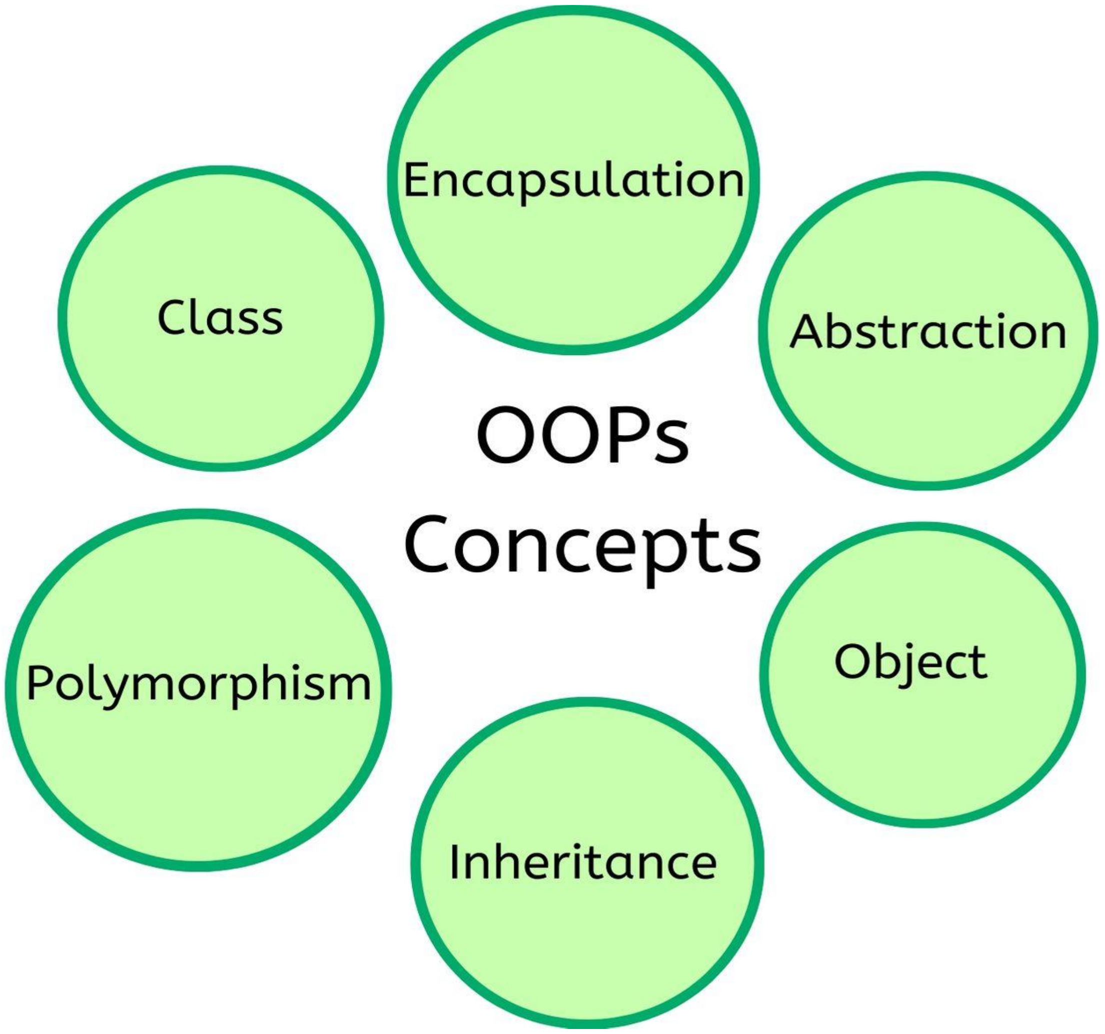
- Emphasis on data rather than procedure.
- Programs are divided into entities known as **objects**.
- Data Structures are designed such that they characterize objects.
- Functions that operate on data of an object are tied together in data structures.
- Data is hidden and cannot be accessed by external functions.
- Objects communicate with each other through functions.
- New data and functions can be easily added whenever necessary.
- Follows bottom up design in program design.

Comparison between Functional programming and oops

S.No.	Functional Programming	Object-Oriented Programming
1.	The functional programming language supports immutable data.(values and data can't be changed)	OOP uses mutable data.(values and Data can be changed)
2.	Functional Programming supports the Declarative Programming Model.	OOP supports the imperative Programming Model.
	*Declarative programming is a programming paradigm that expresses logic of computation without describing its control flow. Ex- Declarative Programming is like asking your friend to fix your car. You don't care how to fix it, that's up to her.	Imperative is a programming paradigm that uses statements that changes a program state. Ex- Imperative Programming is like your friend calling your father that tells her how to fix your car step by step.
3.	Functional Programming focuses on the "**What we are doing".**	OOP focuses on the "**How we are doing**".
4.	The methods of Functional Programming will not produce any side-effects.	Methods of the OOP can produce the side-effects.
5.	Functional Programming follows parallel programming.	OOP does not work on parallel programming.
6.	For the flow control, we do function calls & function calls with recursion.	Object-Oriented Programming supports the use of the loops and conditional statements for the flow control.

7.	For the iteration of the data collection, functional programming uses the "Recursion" concept.	Object-Oriented Programming uses the "Loop" concept for the iteration of Data collection. For example, For-each loop in Java
8.	For functional programming, the execution of statements in the order is not so important.	It is essential for the oop programming to execute the statements in order is very important.
9.	Functional Programming supports "Abstraction over Data" and "Abstraction over Behavior".	OOP supports only "Abstraction over Data".

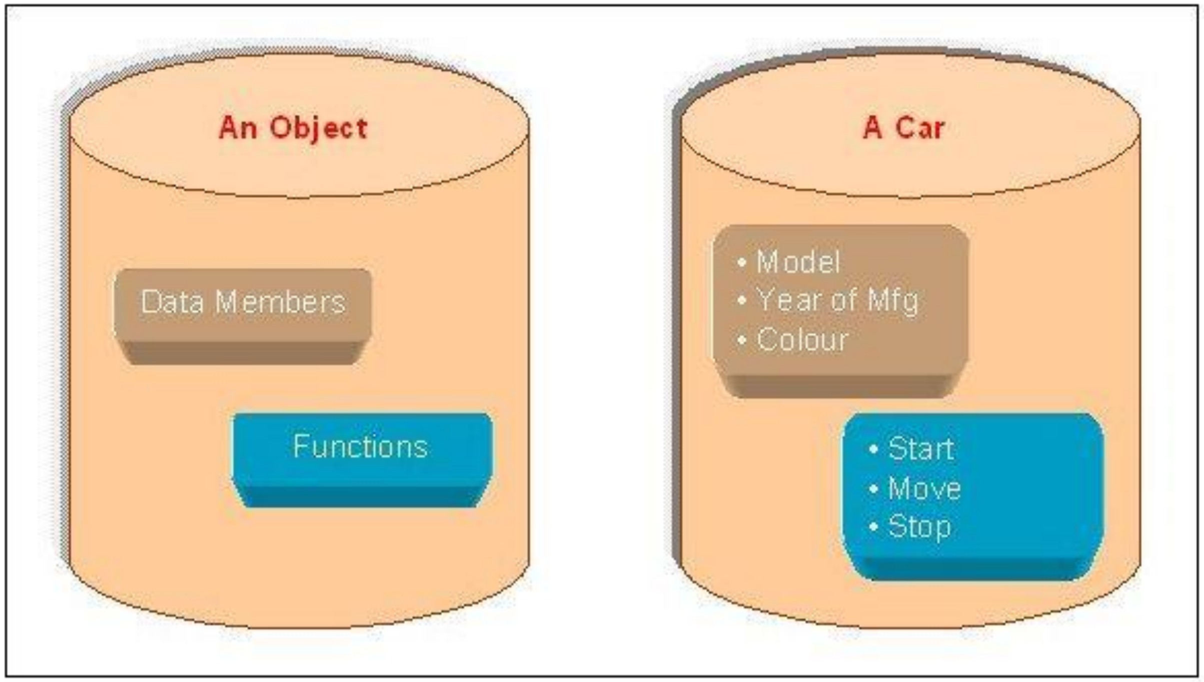
Characteristics of oops



Characteristics of oops

Objects:- An Object is an identifiable entity with some characteristics and behaviour. **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.

Objects are the basic run time entities of any object oriented system.



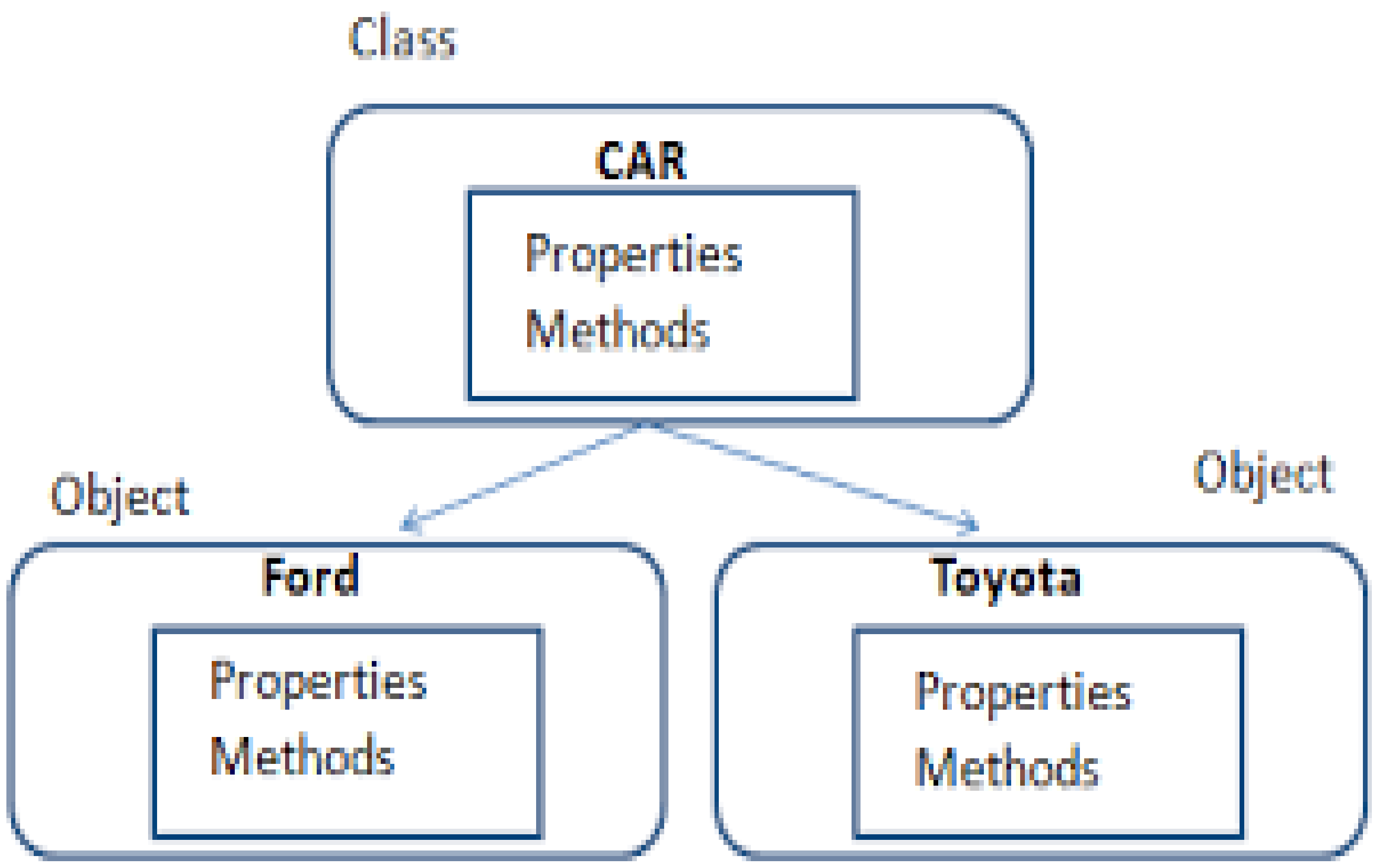
Characteristics of oops

Class:- The building block of C++ that leads to Object-Oriented programming is a Class. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. **A class is like a blueprint for an object.**

A Class is a user-defined data-type which has data members and member functions.

Data members are the data variables and member functions are the functions used to manipulate these variables and together these data members and member functions define the properties and behaviour of the objects in a Class.

Class and objects



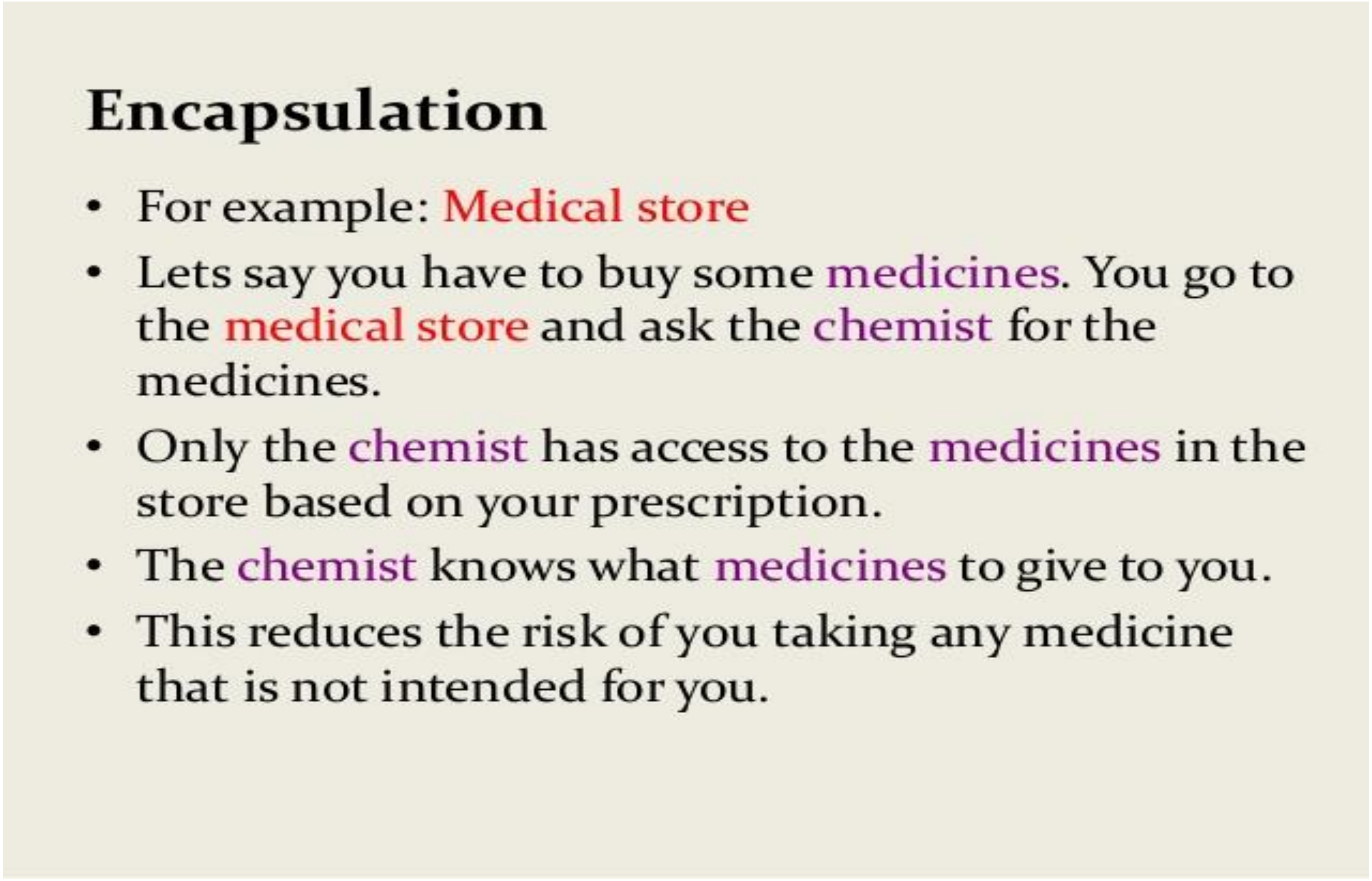
Characteristics of oops

Encapsulation:- Encapsulation is a process of combining data and function into a single unit like capsule. In Object-Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulate them. This is to avoid the access of private data members from outside the class. To achieve encapsulation, we make all data members of class private and create public functions, using them we can get the values from these data members or set the value to these data members.

In simple words:-

Encapsulation is the process of binding data members (variables , properties) and member functions(methods) into a single unit.

Characteristics of oops



Encapsulation

- For example: Medical store
- Lets say you have to buy some medicines. You go to the medical store and ask the chemist for the medicines.
- Only the chemist has access to the medicines in the store based on your prescription.
- The chemist knows what medicines to give to you.
- This reduces the risk of you taking any medicine that is not intended for you.

Characteristics of oops

Abstraction:-

➢Data abstraction is one of the most essential and important features of object-oriented programming in C++.

➢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.

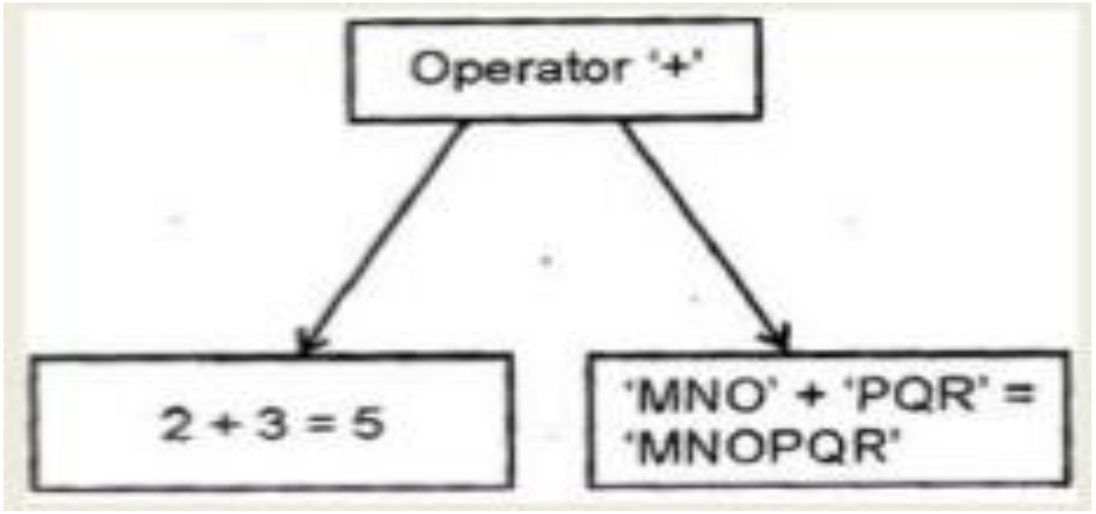
Example- when you a press a key on keyboard the character appears on the screen and you can see only that character but the mechanism involved behind it is what we are not much bothered about.

Characteristics of oops

Polymorphism:-
The word polymorphism is a Greek term which means having more than one form.

In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.

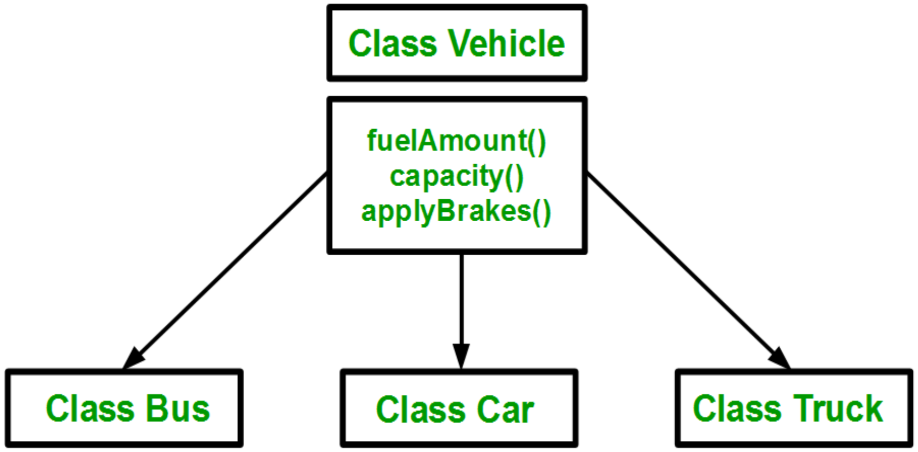
For example:-



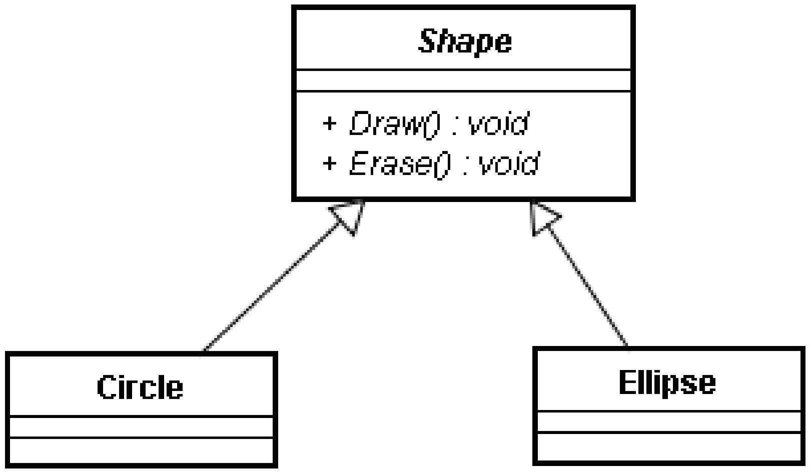
Characteristics of oops

Inheritance:-

1.



2.



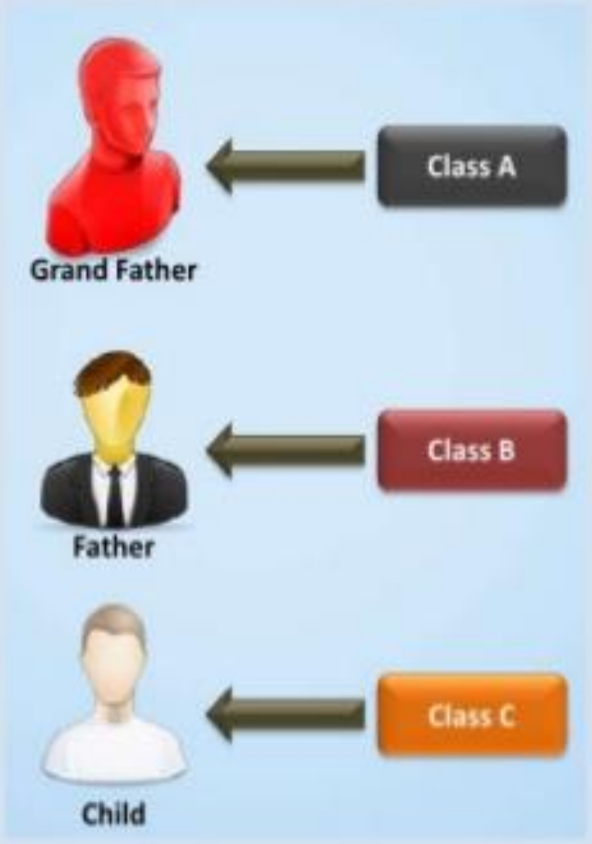
- **The capability of a class to derive properties and characteristics from another class is called Inheritance.**

- Inheritance is one of the most important features of Object-Oriented Programming.

- **Sub Class**: The class that inherits properties from another class is called Sub class or Derived Class.

- **Super Class**: The class whose properties are inherited by sub class is called Base Class or Super class.

- **Reusability**: Inheritance supports the concept of “reusability”, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.



Characteristics of oops

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. 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.