

An object oriented programming is a method of implementation in which programs are organised as a cooperative collection of objects each of which represent an instance of some class and whose classes are all members on a hierarchy of classes united through the property called inheritance.

- Emphasises on data rather than algorithms
- All programs are made up of by designing the objects. objects can be communicated via their interfaces.
- An object is formed data along with operations.

Programming Styles

procedure oriented
object oriented
logic oriented
Rule oriented
Constraint oriented

Abstraction Employed

Algorithmic based
objects, classes
goals oriented (predicate logic)
if-then-else
Invariant relationship.

Object oriented programming language provide the programmer the ability to create class hierarchy, instantiate cooperative objects collectively working on a problem to produce the solution and send messages between objects to process themselves.

The power of the object oriented programming Languages are,

- The programmer can create modular & reusable code to formulate a program by composition and modification of the existing modules.
- Flexibility is gained by being able to change or replace modules without disturbing the other parts of the code.
- Software development speed is gain by reusing and enhancing the existing code.
- programming objects are closest to representation of the realworld objects.

Features :-

Encapsulation

Abstraction

Inheritance

Generality

Resistant Persistence

Containership Delegation

polymorphism

OOP Elements:-

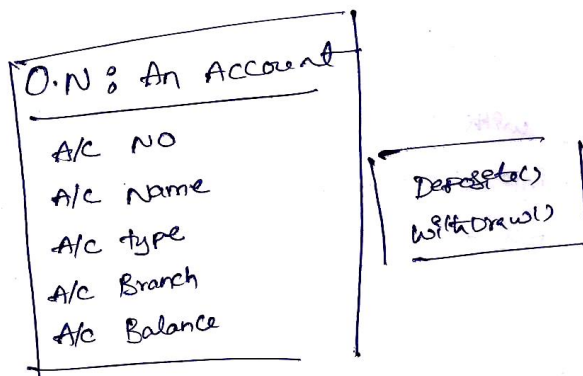
Every OOP must meet the following requirements.

- modelling the real world problem close as possible to the user perspective.
- Interacting easily with computational environment
- constructing
Reusable software components
and
easily extendable libraries.
- Easily modifying and extending implementation components without having to re-code everything from scratch.

Objects:-

An object is an entity which is identified with its attributes.

Ex:- A Person, A Car, A boy, A clock, A book, A tree etc,



Every object will have data structures and its behaviour. These data structures are called attributes and its behaviour is called operations.

An object is basically a runtime entity of the object oriented programming.

Classes :- Grouping of similar objects.

	<u>class</u>	<u>object</u>	<u>Attributes</u>	<u>Operations</u>
1.	person	aboy agiri	name age gender	speak() walk() eat()
2.	vehical	acar avon	modelno colour price wheels	start() stop() accelerate()
3.	Furniture	A Table A bench	length shape	flexibility() move() fixed()
4.	Polygon	Triangle Rectangle	vertices Edges Colour	Draw() Erase() fill colour()

→ The objects with the same data structure and behaviour are grouped into a class. All those objects possessing similar properties are grouped into the same unit.

→ So a class is a template that unites data and operations.

- A class is an abstraction of the realworld entities with similar properties.
- A class is implementation of abstraction type data.

Encapsulation:-

It is a mechanism that associates the code and the data, it implements and keeps them and save them from external interference and misuse.

Data Abstraction:-

Creating new datatypes using encapsulated objects that are well suited to an application to be programmed is known as Abstraction.

Abstraction refers to the act of representing essential features without including background details or explanations. It is a powerful technique creating ADT's and will result in optimal more readable and flexible programs.

Inheritance:-

Inheritance is the process by which one object can acquire the properties of another.

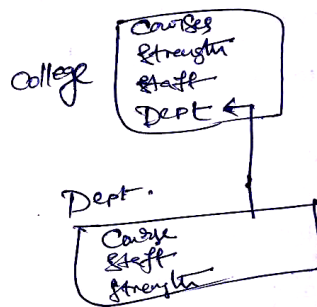
It is for constructing the reusable parts and for extensibility of a module or a program.

The benefits of the inheritance are

1. The code does not have to be rewritten so it leads to increased reliability and decreased maintenance cost.
2. Inheritance will permit the construction of the reusable software components.

Delegation:-

The new approach of object composition takes a view that an object can be a collection of many other objects and the relationship is called the 'has a' relationship.



Polymorphism:-

It allows a single name or an operator to be associated with different operations depending on the type of data passed to it. It has many forms and be implemented as function overloading, operator overloading and dynamic binding.

Message Communication:- polymorphism that allows a single name to be used for more than one related purpose but all these are technically different. This polymorphism which implements one concept called binding.

message communication :-

Communication among the objects occur in the same way as how people exchange the messages between them. Message for an object is interpreted as a request for the execution of a function.

For ex:- In conventional notation to find the square root of 'x' we write $\text{sqrt}(x)$ but where as in 'oop' this is changed to $x.\text{sqrt}()$

<u>Feature :-</u>	<u>Java</u>
Encapsulation (Data hiding)	✓
Single Inheritance	✓
multiple Inheritance	x
polymorphism	✓ (not all)
Binding	late Binding
Concurrency	✓
garbage collection	✓
persistent	x
Generosity	x
Class Library	✓

Disadvantages of OOP:-

1. Run time overhead
2. Compiler overhead
3. Reorientation of software developed to the object-orientation thinking.
4. It requires mastery over the areas on programming methodologies
Software Engineering.
5. Benefits only in long run while managing large software projects.

OOP Definition:-

① An object oriented program is an approach that provides a way of modularising the programs for creating partial memory area by both data and function that can be used as templates for creating copies of such modules on demand.

② Object oriented technology is more than just a way of programming. It is a way of thinking abstractly about a problem using real world concepts rather than computer concepts.