Unit 5
Object Oriented Programming (OOP)

1. Programming Paradigms

Paradigms is a programming model or strategy that has distinct features, framework, patterns, style or way of writing a program. It is a way to solve a particular problem using different methods. Some of the programming paradigms are given below.

A. Structured Programming:

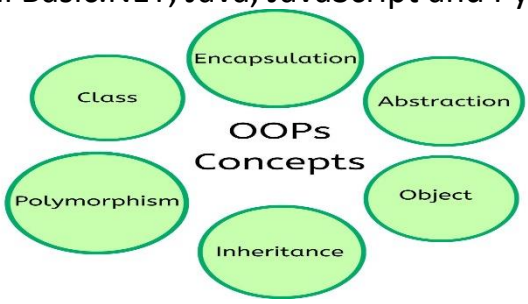
It is a programming paradigm that uses structured control flow. It has three way of making program as sequencing, selection and iteration. BASIC and C are the example of structured programming.

B. Procedural programming:

Procedural programming allows splitting instructions into procedure. It is derived from structured programming language. It is based on the concept of procedure call which is specially known as function. We can call any procedure ar any point during program's execution.

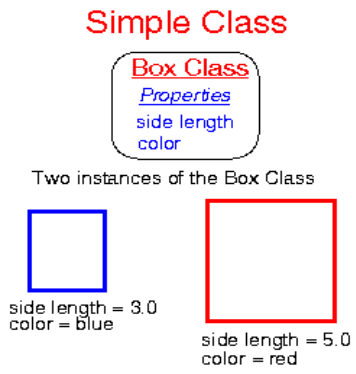
C. Object Oriented Programming (OOP)

Object Oriented programming (OOP) is a programming model that depend on the concept of **classes** and **objects**. It includes several techniques like polymorphism, encapsulation, inheritance etc. C++, Visual Basic.NET, Java, JavaScript and Python are some examples of OOP.



I. **Class:** A class is a collection or group of similar objects that have same properties, common behavior and relationship. It is a blue print or plan that describe the details of an object. When we define a class it just creates template so, no memory is created.

II. **Object:** An object is an instance of class which is declared as many time as per requirement. It determine the behaviour of the class.



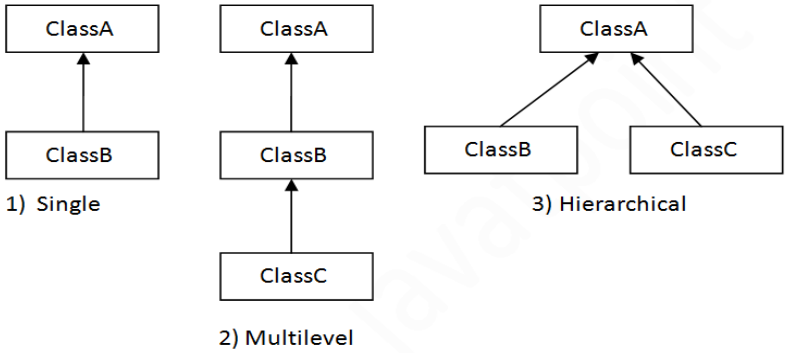
2. Features of OOP

a. Inheritance

he capability of a class to derive properties and characteristics from another class is called **Inheritance**. Inheritance is one of the most important feature 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.



Some Types Inheritance:

Single Inheritance : A class is derived from another single base class.
Multi Level Inheritance: A class is derived from another derived class.
Multiple Inheritance: A class is derived from more than one base class.
Hierarchical Inheritance: More than one derived class created from a single base class.

b. Encapsulation

The process of hiding data is called encapsulation. Encapsulation is an Object Oriented Programming concept that binds data and functions together that manipulate the data, and that keeps both safe from outside interference and misuse. It protect data from outsider.

c. Polymorphism

It is derived from a Latin word poly & morph where poly means many and morph means forms. Polymorphism enables the same function to behave differently on different classes. In other word we can say that polymorphism means that same function behave differently on different classes.

d. Abstraction

It is a process of hiding internal data and allowing only important data to be used is called Abstraction. **Abstraction** is the concept of object-oriented programming that "shows" only essential attributes and "hides" unnecessary information. The main purpose of abstraction is hiding the unnecessary details from the users.

3. Advantages of OOP

- Data hiding concept help to make secure program.
- Inheritance features can be used to eliminate redundant code.
- The feature of one class can be reused in different class.
- Programming is easier because of many features.
- It is newer and improved programming concept.
- Reuse of programs makes program more faster.
- Lower cost of development.
- Higher quality software

4. Disadvantages of OOP

- Larger Program size
- Slower Programs
- Not Suitable for all types of problems

5. Application of OOP

- Real-Time System Design. (flight Control System, Air Bags Control)
- Computer Animation and Gaming
- Client Server System
- Office Automation System (Word, Excel, PowerPoint)
- AI Expert System:
- Decision Support System:

6. Difference between procedural and OOP