

# UNIT - I

## Object Oriented Concepts

&

## Java

# Object Oriented Concepts

---

- Background
- Programming Paradigms
- Concepts of OOPL
- Major and Minor elements
- Class, Object and relationships among objects
- Encapsulation
- Polymorphism
- Inheritance
- Message passing
- Difference between OOP and other conventional programming
- Advantages, Disadvantages and Applications

# Object Oriented Concepts

## Background:

- The process problem solving using a computer is complicated process requiring careful planning, logical precision, persistence and attention to detail.
- A programmers primary task is to write software to solve a problem.
- Many programming models have evolved to help programmers in being more effective such as Modular, Top-down, Bottom-up, Structured, Object-Oriented programs etc.

# Object Oriented Concepts

## Programming Paradigms:

- All computer programs consists of two elements:  
**Code and  
Data**
- A program can be conceptually organised around its code or around its data.
- *Major Programming Paradigms are:*  
**Procedure Oriented  
Object Oriented**

# Object Oriented Concepts

## Programming Paradigms: ... POP

- The interdependent functions cannot be used in other programs. As a result, even for a similar task across programs, the entire function has to be recoded. This made program development a more complex task.
- A change means rewriting huge portions of the code. As a result of this, software maintenance costs are very high.
- This approach failed to show the desired result in terms bug free, easy-to-maintain and reusable programs.

# Object Oriented Concepts

## Programming Paradigms: ... OOP

- An approach that provides a way of modularizing programs by creating partitioned memory area for both data and code that can be used as templates for creating copies of such modules on demand.
- OOP is a programming methodology that helps organizing complex programs through the use of the three principles – *Encapsulation, Polymorphism and Inheritance*.
- OOP enables you to consider a real-world entity as an object.
- OOP combines user-defined data and instructions into a single entity called an *Object*.

# Object Oriented Concepts

Programming Paradigms: ... OOP...

*It is a well suited paradigm for the following:*

- Modelling the real world problem as close as possible to the perspective of the user.
- Constructing reusable software components and easily extendable libraries.
- Easily modifying and extending implementations of components without having to recode everything from scratch.

# Object Oriented Concepts

## Concepts of OOP Languages

- Classes
- Objects
- Encapsulation
- Polymorphism
- Inheritance
- Dynamic Binding
- Message Communication



# Object Oriented Concepts

## Concepts of OOP Languages...

**Classes:** “*An user-defined data type/ADT*”

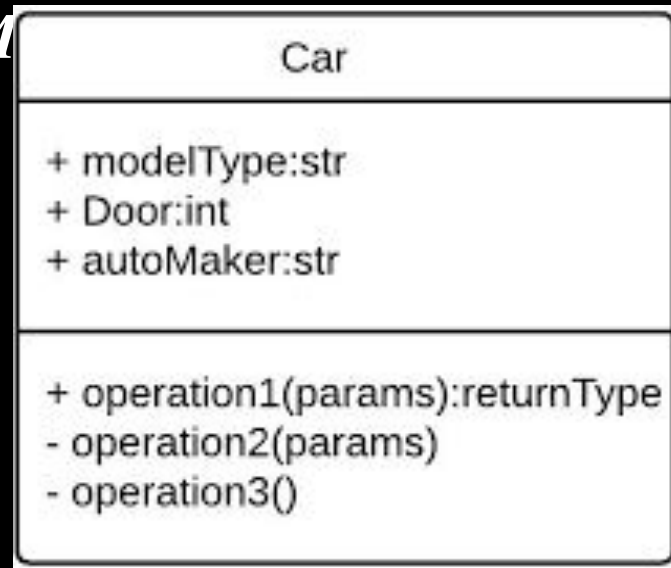
- A blue print used to instantiate many objects.
- Defines set of data members (States/Attributes) and set of methods (Behaviors).
- A *template* for an object / Collection of objects of similar type.
- No memory is allocated when a class is created. Memory is allocated only when an object is created, i.e., when an instance of a class is created.

# Object Oriented Concepts

## Concepts of OOP Languages...

### Classes: ...

- **Eg:** A Class of **Cars** under which Santro Xing, Alto and WaganR represents individual Objects. In this context each Car Object will have its own, *Model, Year of Manufacture, Colour, Top Speed, Engine Power etc., which form Properties of the Car class* and the associated actions i.e., object functions like *Start, Move, Stop form the M*



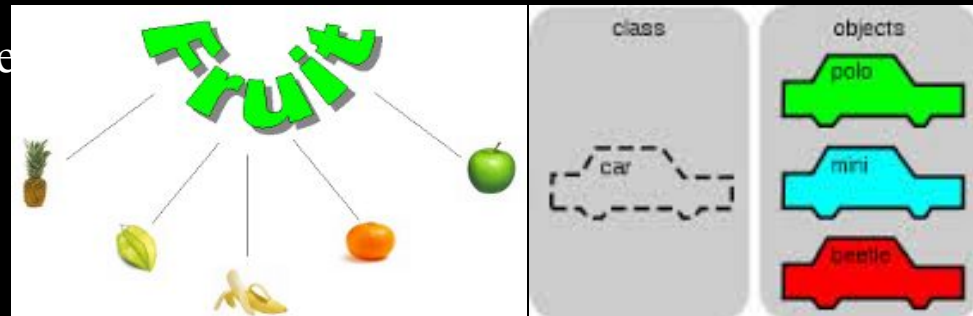
# Object Oriented Concepts

## Concepts of OOP Languages...

### Objects: “Instances”

- The basic Building Blocks/Run-time Entities having two characteristics: *State and Behavior*.
- An Object is a collection of data members and associated member functions also known as methods.
- Each instance of an object can hold its own relevant data.
- Memory is allocated only when an object is created, i.e., when an instance of a class is created.
- **Eg:** `Employee emp1; emp1 = new Employee();` *OR*

`Employee emp1 = new Employee`

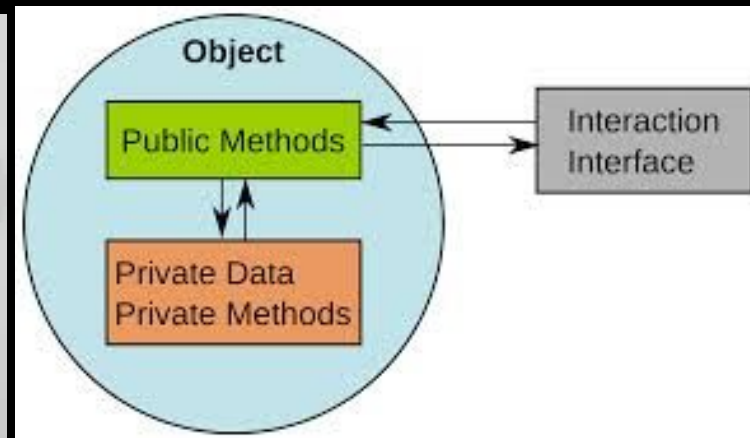
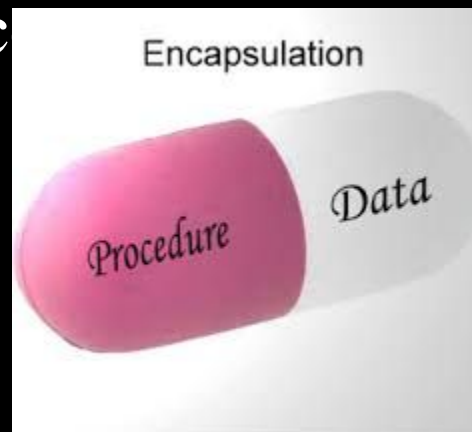


# Object Oriented Concepts

## Concepts of OOP Languages...

### Encapsulation: “*Wraps Data and Method*”

- The mechanism that binds together Code and the Data it manipulates.
- Keeps both safe from outside interference and misuse.
- The wrapping up of data and methods into a single unit called *class*, access is tightly controlled through a well defined interface



# Object Oriented Concepts

## Concepts of OOP Languages...

**Polymorphism:** *“ability to take more than one form”*

- A feature that allows one interface to be used for a general class of actions.
- It allows an object to have different meanings, depending on its context.
- *“One Interface, Multiple Methods”* to design a generic interface to a group of related activities.
- **Eg:** Coffee day vending machine

# Object Oriented Concepts

## Concepts of OOP Languages...

### Polymorphism: ...

Car



Ford



Honda



Move

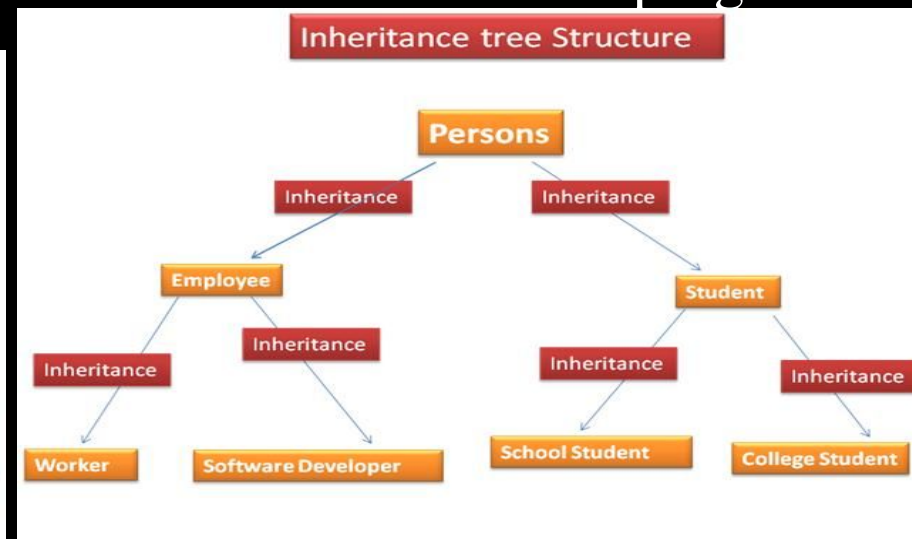
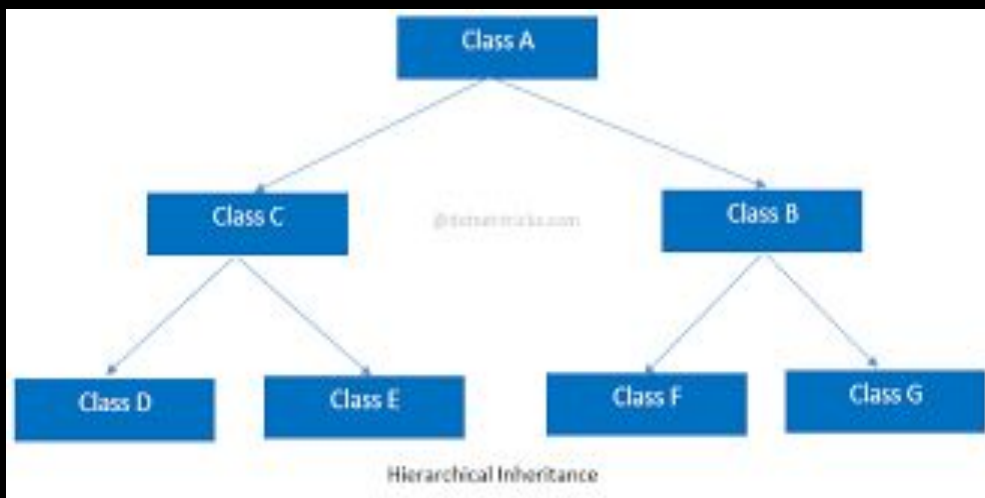
- Car uses normal engine to move
- Ford uses V engine to move
- Honda uses i-vtec technology to move

# Object Oriented Concepts

## Concepts of OOP Languages...

### Inheritance: *“The Idea of Reusability”*

- The process by which one object acquires the properties of another object **OR** the process of forming a new class from an existing class or base class.
- Supports the concepts of hierarchical classifications.
- Inheritance helps in reducing the overall code size of the program.



# Object Oriented Concepts

## Concepts of OOP Languages...

**Dynamic Binding:** *“Associated with the concept of Inheritance and Polymorphism”*

- Binding means **link** between **procedure** call and **code** to be execute.
- It is the **process** of linking of a function call to the actual code of the function at **run-time**.
- That is, in dynamic binding, the actual code to be executed is not known to the compiler until run-time.
- It is also known **late binding**.

- At run-time the code matching the object under current reference will be

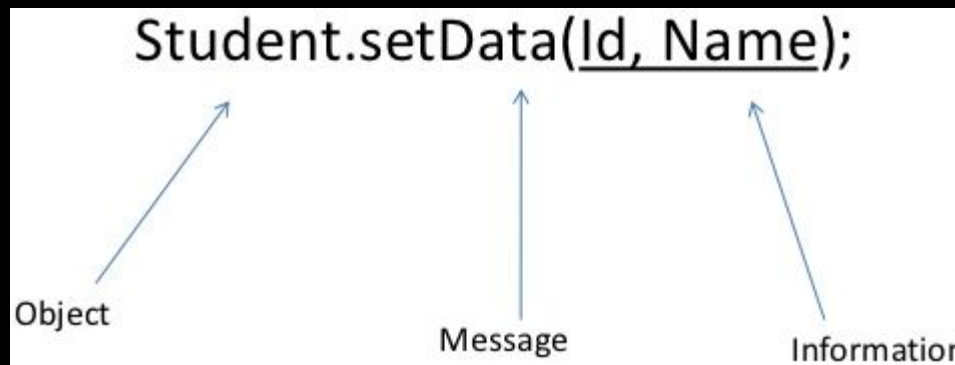


# Object Oriented Concepts

## Concepts of OOP Languages...

**Message Communication:** “A request for execution of a method”

- Objects communicate with one another by sending and receiving information.
- A message of an object is a request for execution of a procedure/method that generates desired result.
- **Eg:** *student1.examResult(4JC10CS901);*



# Object Oriented Concepts

## Difference between OOP and other conventional programming

Procedure Oriented	Object Oriented
Emphasis is on procedure rather than data, characterises a program as a series of linear steps.	Emphasis is on data, data is hidden and cannot be accessed by external functions.
<i>Process-centric approach</i>	<i>Data-centric approach</i>
Programs are written around “What is happening” – Code acting on data.	Programs are written around “Who is being affected” – Data controlling access to code.
Programs are divided into smaller parts called functions/modules.	Programs are divided into objects, may communicate with each other through methods.
Function can call one from another and difficult to separate due to interdependency between modules.	Objects are independent used for different programs.
Follows Top-down approach in program design.	Follows Bottom-up approach.
<b>Eg:</b> Basic, Pascal, COBOL, Fortran, C, etc.	<b>Eg:</b> Smalltalk, C++, Objective C, Ada, Objective Pascal, Java(Pure OOL), etc.

# Object Oriented Concepts

## Advantages and Disadvantages

### Advantages:

**Simplicity:** software objects model *real world objects*, so the complexity is reduced and the program structure is very clear.

**Modularity:** each object forms a separate entity whose internal workings are decoupled from other parts of the system.

**Modifiability:** it is easy to make minor changes in the data representation or the procedures in an OO program. Changes inside a class do not affect any other part of a program, since the only public interface that the external

# Object Oriented Concepts

## Advantages and Disadvantages

### Advantages:...

**Extensibility**(Resilience – System can be allowed to evolve): adding new features or responding to changing operating environments can be solved by introducing a few new objects and modifying some existing ones.

**Maintainability:** objects can be maintained separately, making locating and fixing problems easier.

**Re-usability:** objects can be reused in different programs.

**Design Benefits:** Large programs are very difficult to write. OOPs force designers to go through an extensive planning phase, which makes for better designs with less flaws. In addition, once a program reaches a certain size, Object Oriented Programs are actually easier to program than

# Object Oriented Concepts

## Advantages and Disadvantages...

### Disadvantages:

- **Size:** Programs are much larger than other programs.
- **Effort:** Require a lot of work to create, coders spent more time actually writing the program.
- **Speed:** Slower than other programs, partially because of their size also demand more system resources.
- Not all programs can be modelled accurately by the objects model.
- One programmer's concept of what constitutes an abstract object might not match the vision of another programmer.

*However; many novice programmers do not like Object Oriented Programming because of the great deal of work required to produce minimal results.*

# Object Oriented Concepts

## Applications:

- **User-Interface Design** (CUI/CLI, GUI...WIN), Games, CAD/CAM/CIM systems
- **Real-Time System**
- **Simulation and Modeling**
- **Object-Oriented Database**
- **Artificial Intelligence and Expert Systems**
- **Neural Networks and Parallel Programs**
- **Decision Support and Office Automation System**

*A software that is easy to use hard to build. OOP changes the way software engineers will Think, Analyse, Design and Implement systems in the future*