OBJECT ORIENTED PROGRAMMING

Presented by

Smt Reetanjali Panda

Lecturer,UCPES

Programming Languages

- A program is a set of instructions that tell a computer what to do. We execute a program to carry out the instruction listed by that program. Instructions are written using programming languages.
- Programming languages are classified as
- Object Oriented Programming Languages Ex. C++, Java
- High-Level Languages Ex. C, , FORTRAN, and COBOL.
- MiddleLevel Language Ex Assembly Language
- finally the lowest level as the Machine Language.
- The lowest level Machine Language consists of binary codes to write the programs which is directly understandable by the machine.
- MiddleLevel Language(Assembly Language) consists of mnemonic codes and symbolic names to write down the programs.

Programming Languages(Cont.)

High-level programming languages are programming languages using English like language that are rather natural for people to write. It provides Top Down approach to design a software.

These are of two types:

1. Structured or Procedure oriented : The program is divided in to modules. It is function oriented, provides functional abstraction and data moves freely from one function to another.
2. Unstructured : The most primitive of all programming languages having sequential flow of control. They do not contain any procedures or functions.

Using OOP concept, programs are written in terms of objects rather than functions as we perceive the real world in terms of objects. The basic concept of OOPs is to create objects, re-use them throughout the program, and manipulate these objects to get results. Data & actions are combined together in to a single unit called class. It provides Bottom Up approach to design a software.

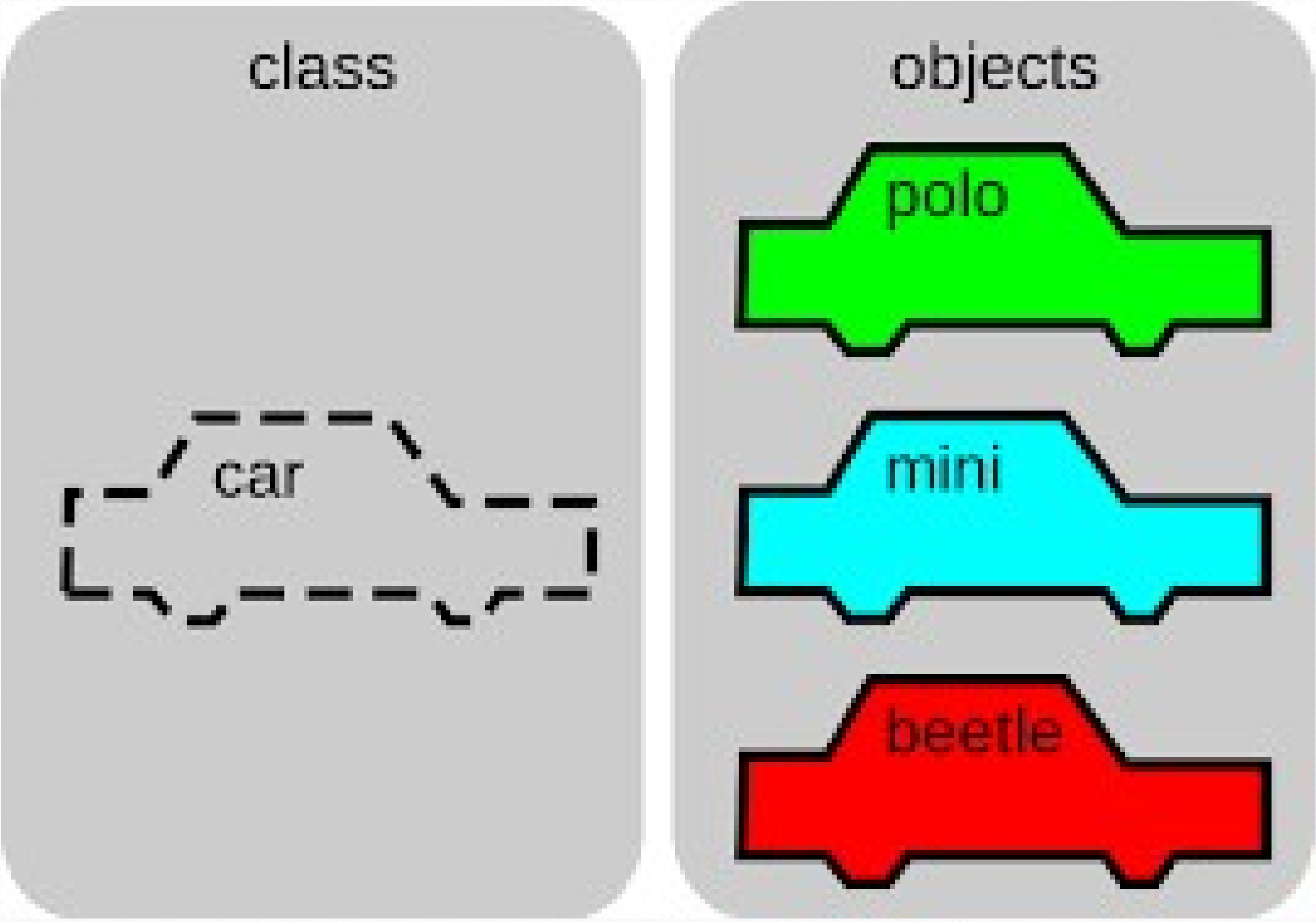
Object Oriented Programming Concepts

- OBJECT ORIENTED PROGRAMMING (OOP) is a programming concept that allows users to write programs in terms of objects as we perceive the real world in terms of objects. It provides Bottom Up approach to design a software, to create the objects that they want and then, create methods to handle those objects, works on the principles of abstraction, encapsulation, inheritance, and polymorphism. The basic concept of OOPs is to create objects, re-use them throughout the program, and manipulate these objects to get results.

<u>Features of OOPS concepts:</u>

1) Class

The class is a group of similar entities. It is only an logical component and not the physical entity. For Example: Book can be considered as a class having different data to define it, such as name, title , publication, the accession number, cost, borrower, date of issue etc, called as Data members. The different methods it can contain   required to process a book can be , open a book, close a book , issue a book ,return a book etc called Member methods.


class
objects
polo
car
mini
beetle

Object Oriented Programming Concepts(Cont.)

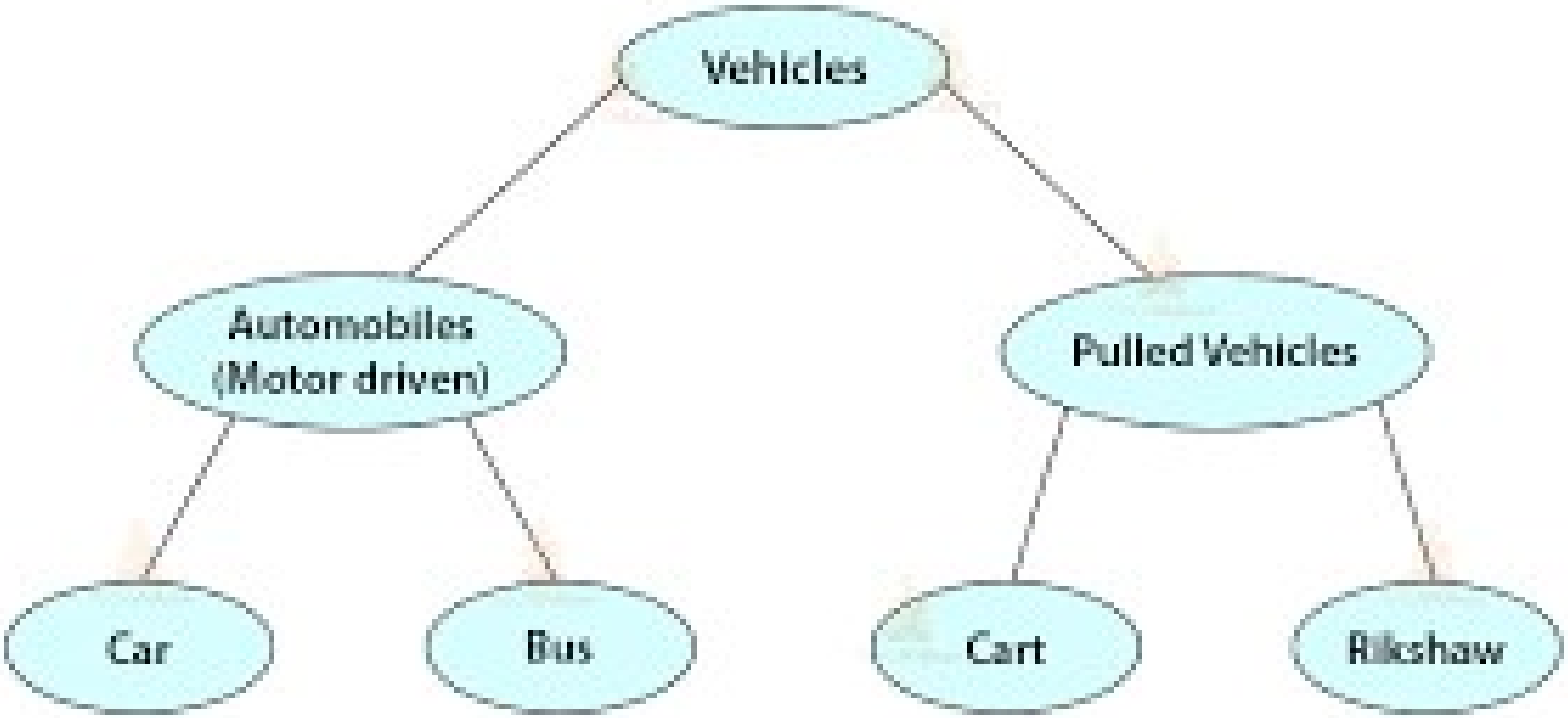
2) Object

An object can be defined as an instance of a class, and there can be multiple instances of a class in a program. They are the real world entities about which want to record data and manipulate. An Object contains both the data and the function, which operates on the data. For example - chair, bike, student,teacher, pen, table, car, etc.

3) Inheritance(Is- A Relationship)

Inheritance is an OOPS concept in which one object acquires the properties and behaviours of the parent object. It creates a parent-child relationship between two classes. It offers robust and natural mechanism for organizing and structure any software. It provides the benefit of reusability. The new class is called as the derived class and the existing class is called as the base class. For example: Books can be of two types Text Book or Reference Book. So these two classes are derived from the base class Book, where to the existing features of the base class Book some extra features are added in the derived classes. Similarly classes Automobiles and Pulled Vehicles can be derived from the class Vehicle from which car ,bus, cart and rickshaw classes can be derived subsequently as described in the figure.

Inheritance in Java



Vehicles

Automobiles (Motor driven)

Pulled Vehicles

Car

Bus

Cart

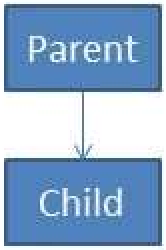
Rikshaw

Object Oriented Programming Concepts(Cont.)

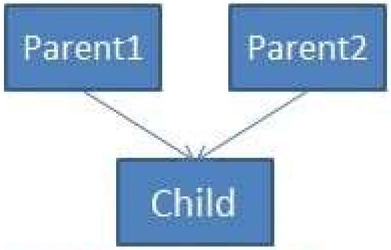
- Types of Inheritance: There are five types of inheritance.

1. Single Inheritance: A single derived class is derived from a single base class.
2. Multiple Inheritance: A single derived class is derived from more than one base class. This type of inheritance is not found in Java.
3. Multilevel Inheritance: There is multiple levels of inheritance. A single derived class is derived from a single base class from which another class is derived.
4. Hierarchical Inheritance: From a single base class, more than one classes are derived. It forms an inverted tree like structure.
5. Hybrid Inheritance: Combination of more than one type of inheritance.

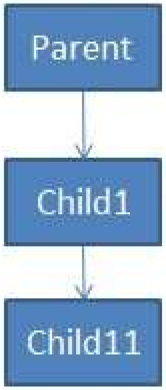
Types of Inheritance


Parent
Child

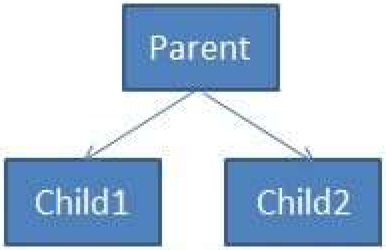
1. Single Inheritance


Parent1
Parent2
Child

2. Multiple Inheritance

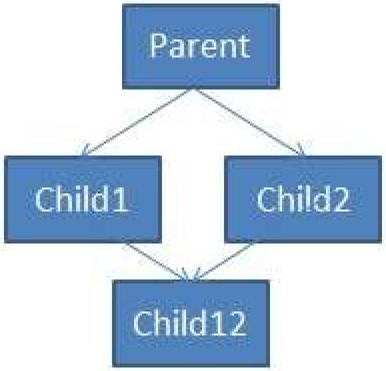

Parent
Child1
Child11

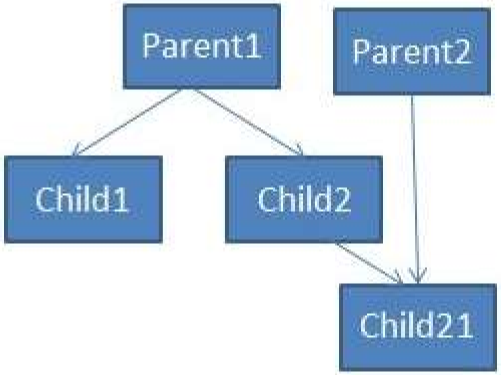
3. Multi-Level Inheritance


Parent
Child1
Child2

4. Hierarchical Inheritan

5. Hybrid Inheritance Variations (Mix of Single & Multiple Inheritance)


Parent
Child1
Child2
Child12


Parent1
Parent2
Child1
Child2
Child21

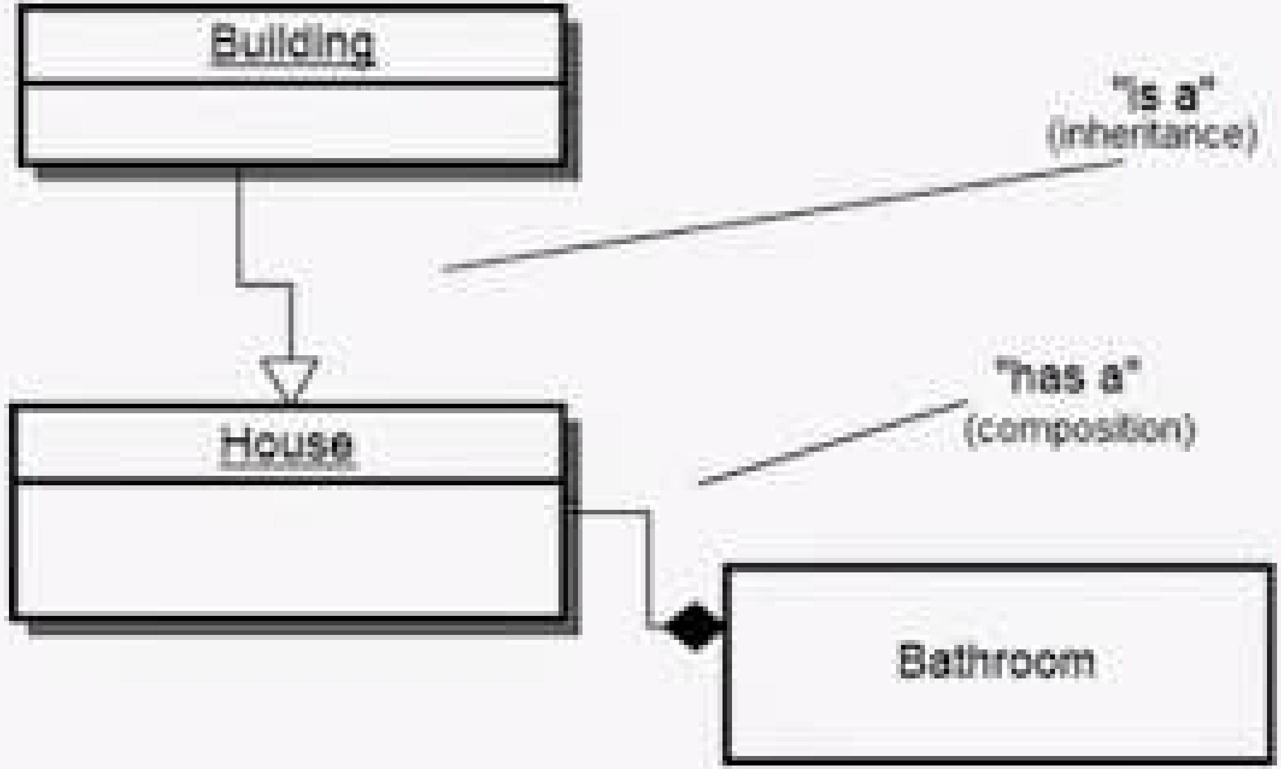
Note: The ones marked in red are not supported by Java

Object Oriented Programming Concepts(Cont.)

4) Composition (Has- A Relationship)

The composition is the strong type of association. An association is said to be a composition if an Object owns another object and another object cannot exist without the owner object. Let's take an example of House and rooms. Any house can have several rooms. One room can't become part of two different houses. So, if you delete the house, room will also be deleted.

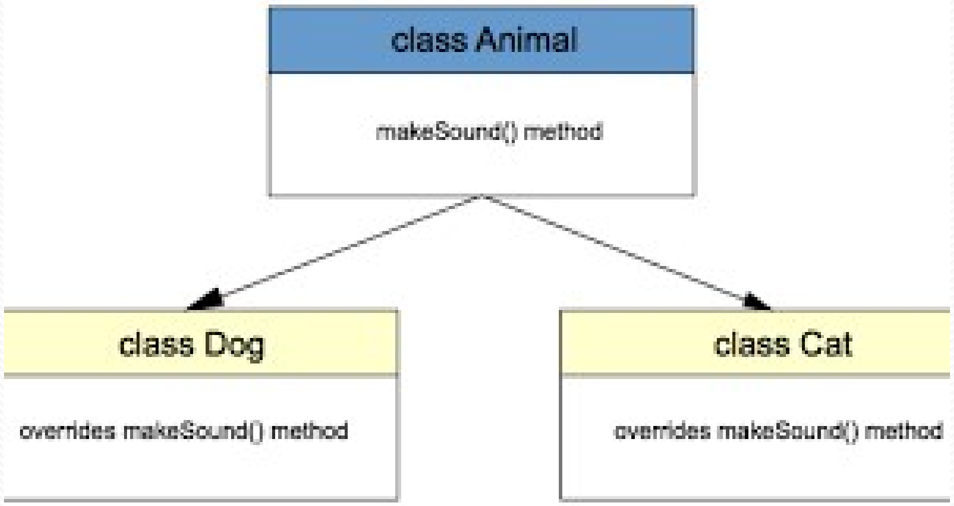
Composition vs. Inheritance



Object Oriented Programming Concepts(Cont.)

5) Polymorphism(Poly means many and Morph means form):

Polymorphism refers to the ability of a variable, object or function to take on multiple forms. For example, in English, the verb *run* has a different meaning if you use it with *a laptop*, *a foot race*, and *business, they are having different meaning according to the context they are used*. Similarly we can write an add method in a class and use it to attain polymorphism for adding two integers, two floating point numbers or concatenate two strings etc. Polymorphism is of two types: Compile time polymorphism and Run time polymorphism based on the decision of invoking a function made at compile time or run time respectively.



class Animal

makeSound() method

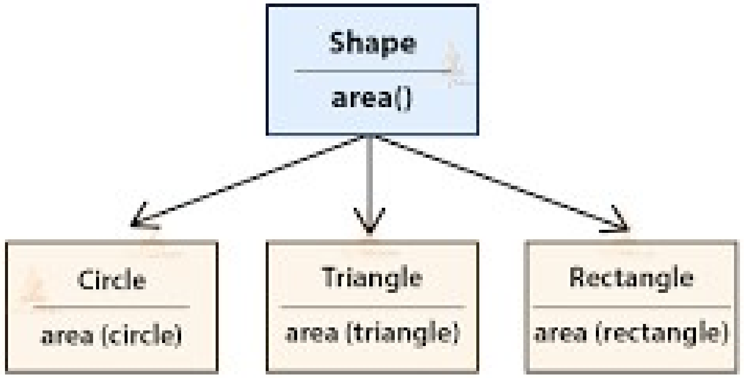
class Dog

overrides makeSound() method

class Cat

overrides makeSound() method

Example of Polymorphism in Java



Shape

area()

Circle

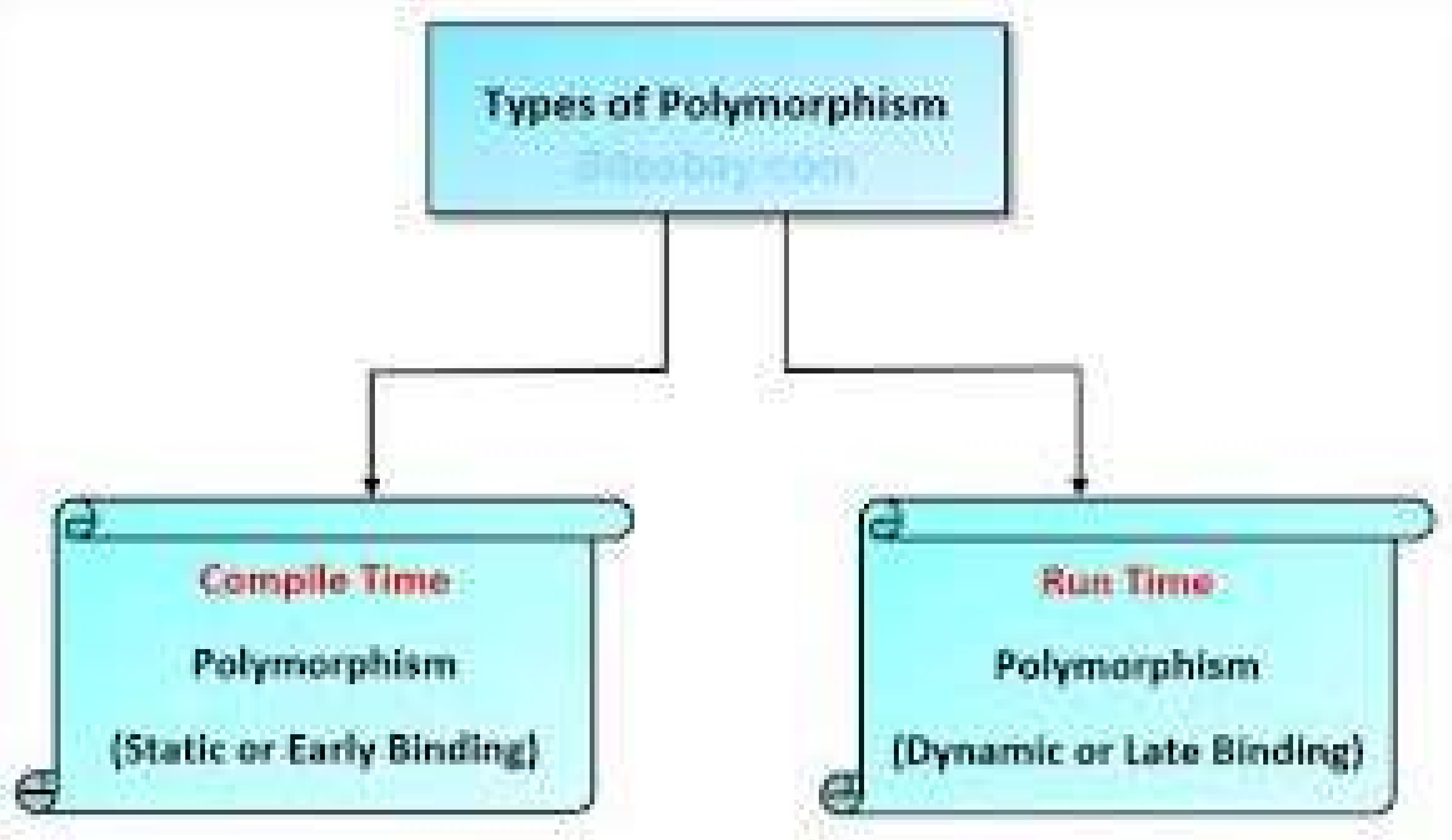
area (circle)

Triangle

area (triangle)

Rectangle

area (rectangle)



Types of Polymorphism

Compile Time

Polymorphism

(Static or Early Binding)

Run Time

Polymorphism

(Dynamic or Late Binding)

Object Oriented Programming Concepts(Cont.)

6) Data Abstraction

- An abstraction is an act of representing essential features without including background details. It is a technique of creating a new data type that is suited for a specific application. For example, while driving a car, you do not have to be concerned with its internal working. Here you just need to concern about the use of the parts like steering wheel, Gears, accelerator, etc. The main purpose of abstraction is hiding the unnecessary details from the users. Abstraction is selecting data from a larger pool to show only relevant details of the object to the user. It helps in reducing programming complexity and efforts. However, the same information once extracted can be used for a wide range of applications. For instance, you can use the same data of customers for a bank application , for hospital application, job portal application, a Government database, etc. with little or no modification. Hence, it becomes your Master Data. This is an advantage of Abstraction.

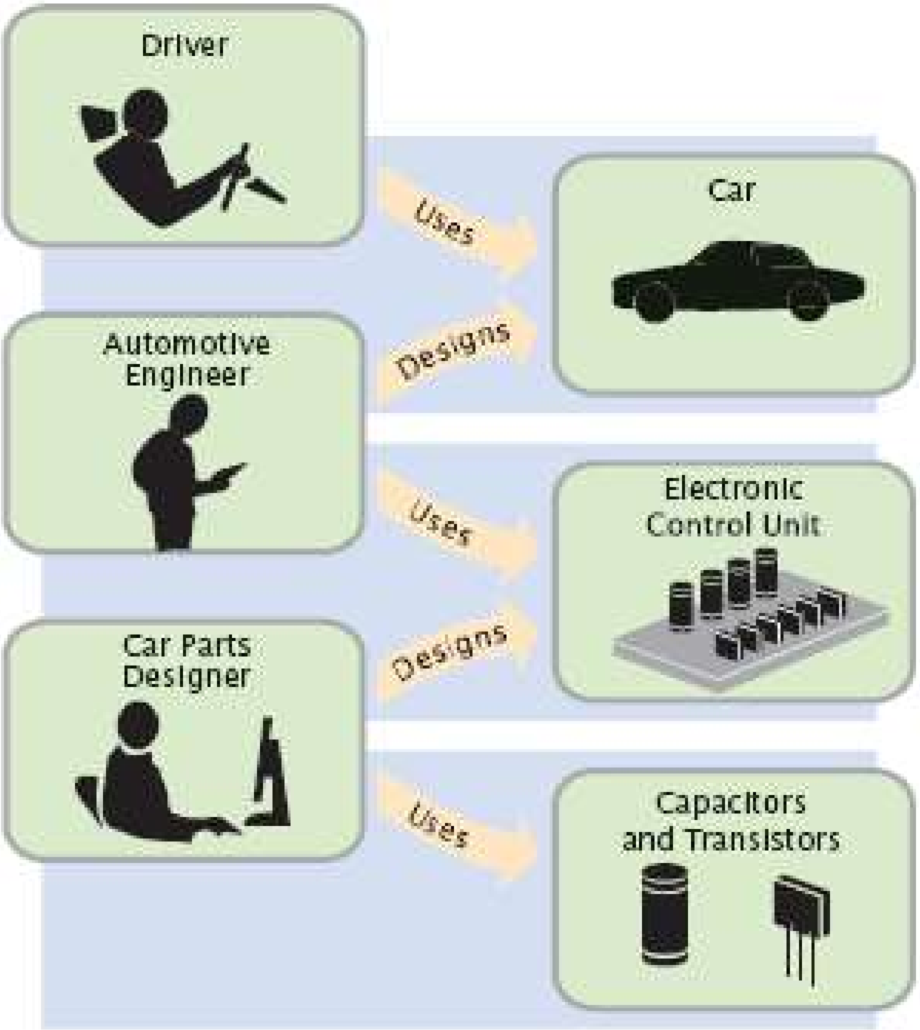


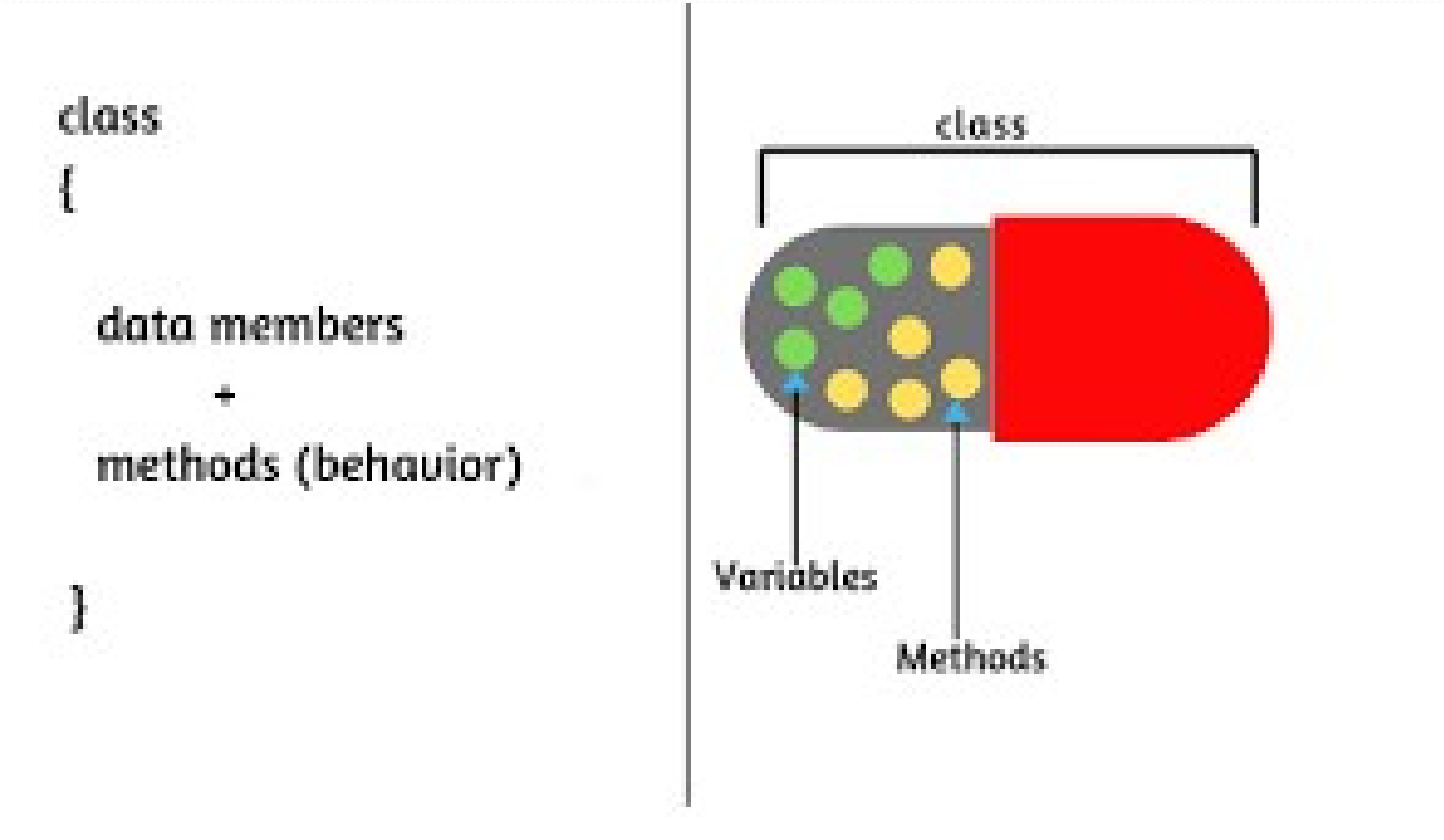
Figure 1

Levels of Abstraction in Automotive Design

Object Oriented Programming Concepts(Cont.)

7) Encapsulation

Encapsulation is an OOP technique of wrapping the data members and member functions into a single unit called class. In this OOPS concept, the data members of a class are always hidden from other classes. It can only be accessed using the methods of their current class. It is the process of hiding information details and protecting data and behaviour of the object from the outside world . In Function oriented programming , the global data was shared among the functions where as in Object Oriented Programming data are distributed among the objects and are localised inside a class. The objects can communicate using member functions in the class.



```
class
{

  data members
          +
  methods (behavior)

 }
```

Difference between Abstraction and Encapsulation

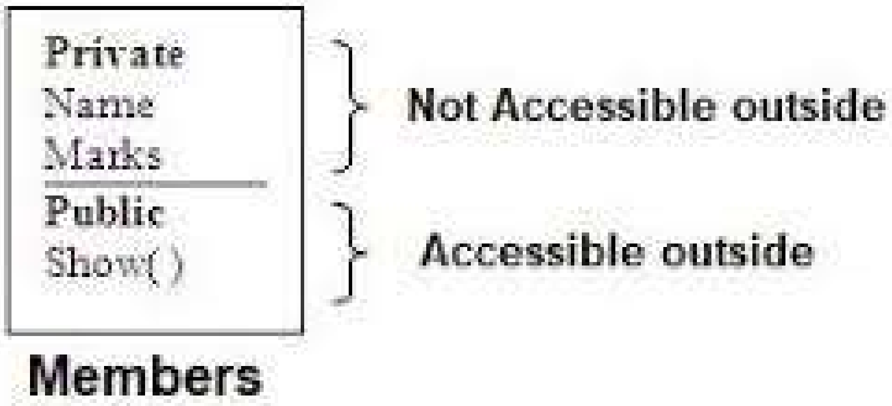
Abstraction	**Encapsulation**
Abstraction solves the issues at the design level.	Encapsulation solves it implementation level.
Abstraction is about hiding unwanted details while showing most essential information.	Encapsulation means binding the code and data into a single unit.
Abstraction allows focussing on what the information object must contain	Encapsulation means hiding the internal details or mechanics of how an object does something for security reasons.
Abstraction is more about "What" a class can do.	Encapsulation is more about "How" to achieve a functionality

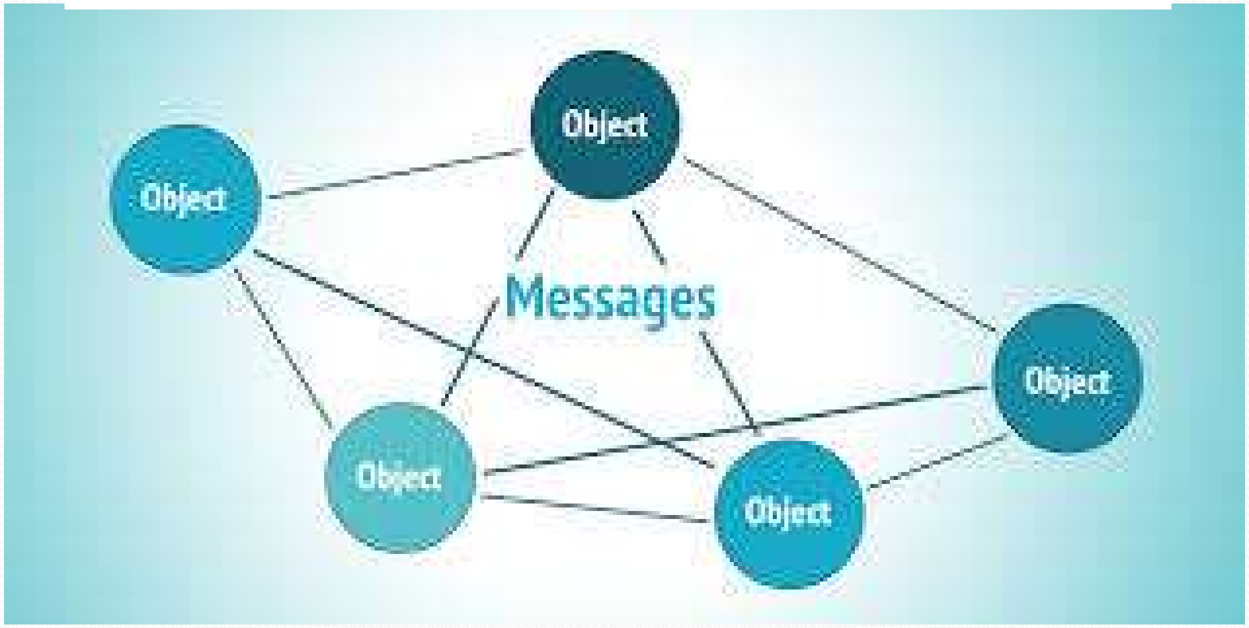
- 8) **Data Hiding :**
- Data Hiding is hiding the variables of a class from other classes. It can only be accessed through the method of their current class. It hides the implementation details from the users. But more than data hiding, it is meant for better management or grouping of related data.

- 9) **Message Passing:**
- Objects can communicate among themselves by invoking the member functions through a dot (.) operator as follows:

  object name . Member method()

Data Hiding

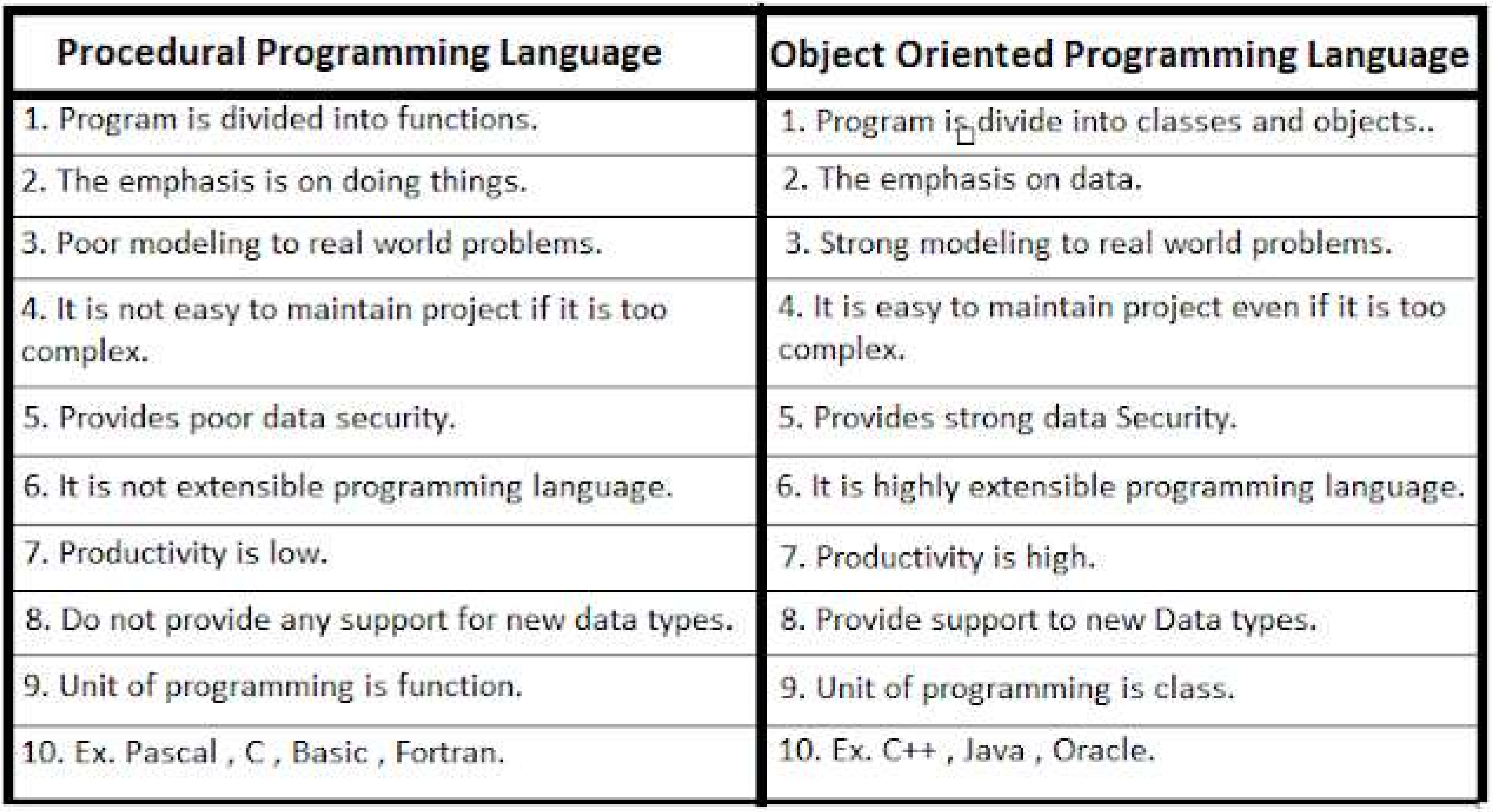

Private
Name
Marks
Public
Show( )
Not Accessible outside
Accessible outside
Members


Object
Object
Messages
Object
Object
Object

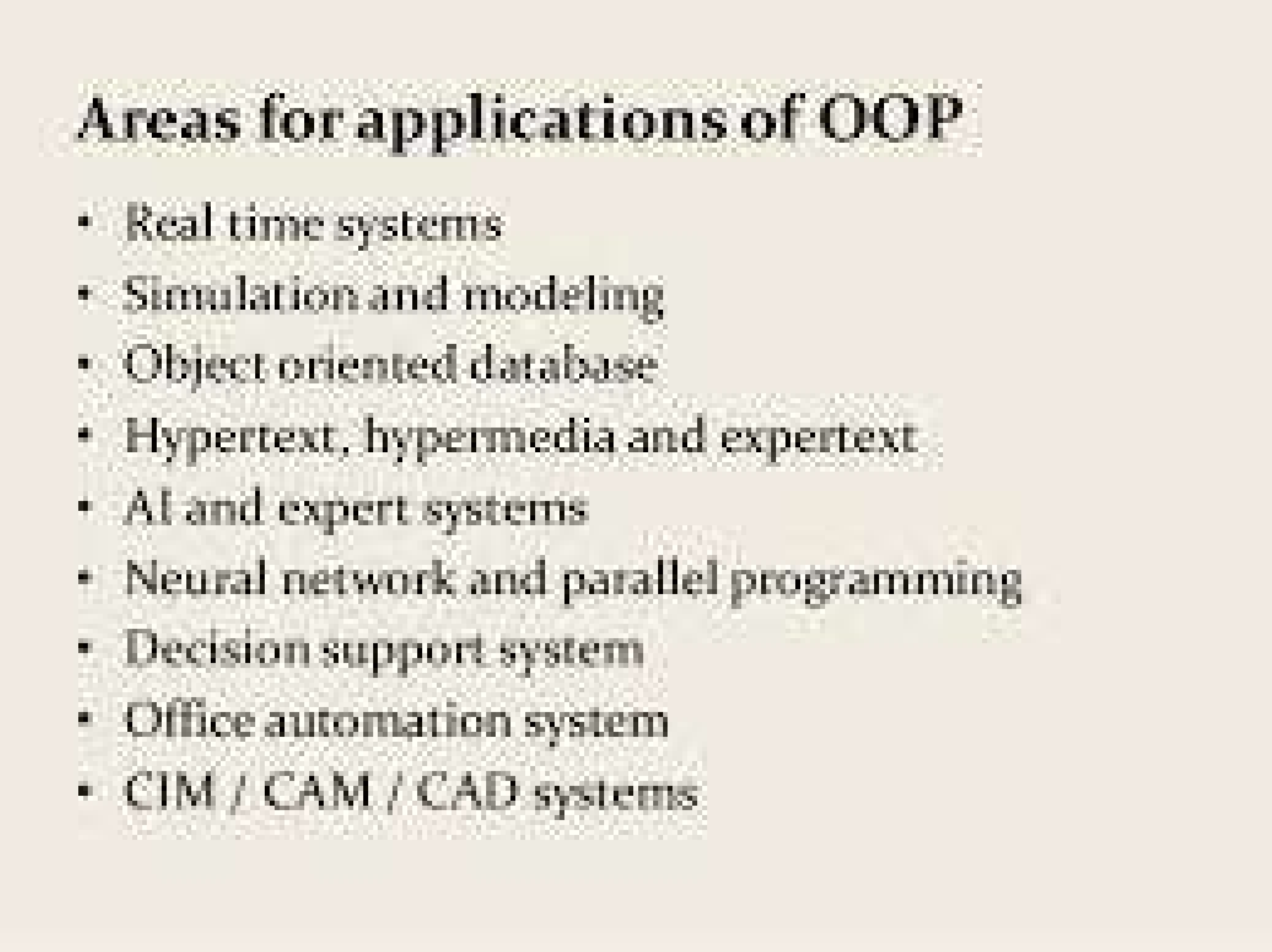
Advantages of OOP

1. OOP offers easy to understand and a clear modular structure for programs.
2. Objects created for Object-Oriented Programs can be reused in other programs. Thus it saves significant development cost.
3. Large programs are difficult to write, but if the development and designing team follow OOPS concept then they can better design with minimum flaws.
4. It also enhances program modularity because every object exists independently.
5. The data is more secured.
6. Provides easy maintainability and High productivity

Difference between Procedure Oriented and Object Oriented



Procedural Programming Language	Object Oriented Programming Language
1. Program is divided into functions.	1. Program is divide into classes and objects..
2. The emphasis is on doing things.	2. The emphasis on data.
3. Poor modeling to real world problems.	3. Strong modeling to real world problems.
4. It is not easy to maintain project if it is too complex.	4. It is easy to maintain project even if it is too complex.
5. Provides poor data security.	5. Provides strong data Security.
6. It is not extensible programming language.	6. It is highly extensible programming language.
7. Productivity is low.	7. Productivity is high.
8. Do not provide any support for new data types.	8. Provide support to new Data types.
9. Unit of programming is function.	9. Unit of programming is class.
10. Ex. Pascal , C , Basic , Fortran.	10. Ex. C++ , Java , Oracle.



Areas for applications of OOP

- Real time systems
- Simulation and modeling
- Object oriented database
- Hypertext, hypermedia and expertext
- AI and expert systems
- Neural network and parallel programming
- Decision support system
- Office automation system
- CIM / CAM / CAD systems

THANK YOU