

Chapter-6

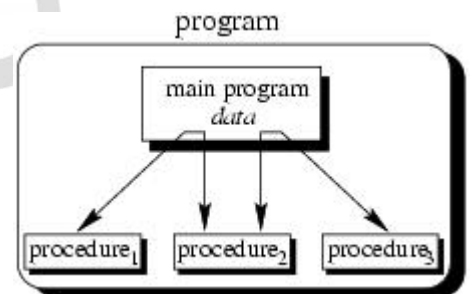
BASIC CONCEPT OF OOP

➤ Introduction:

- Object oriented programming is the principle of design and development of programs using modular approach.
- Object oriented programming approach provides advantages in creation and development of software for real life application.
- The basic element of object oriented programming is the data.
- The programs are built by combining data and functions that operate on the data.
- Some of the OOP's languages are C++, Java, C #, Smalltalk, Perl, and Python.

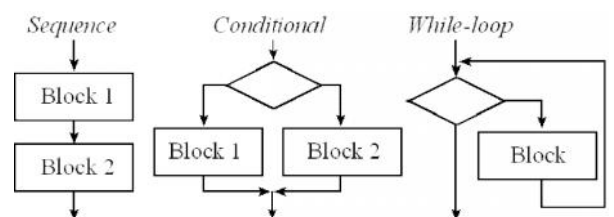
➤ Procedural programming:

- The procedural programming focuses on **processing** of instructions in order to perform a desired computation. Therefore it emphasizes more on doing things like algorithms.
- This programming is lengthy, increases the complexity of program, difficult to understand and modify the program.
- This technique is used in a conventional programming language such as C and Pascal.



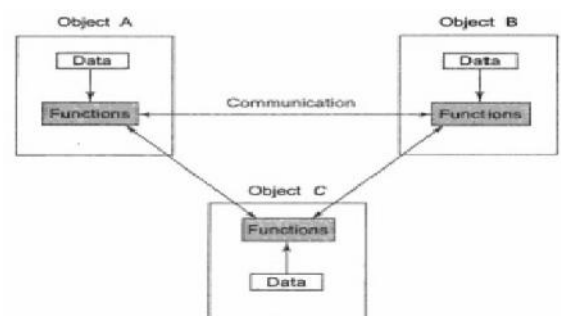
➤ Structured programming:

- An organized approach to programming involving the use of three basic control structures – Sequence, Conditional and loop.
- The top-down concepts to decompose main functions into lower level components for modular coding purpose.
- The major drawback is that it is very difficult to model the real world scenario using this model.



➤ Object oriented programming:

- Object oriented programming (OOP) is a concept that combines both the data and the functions that operate on that data into a single unit called the object.
- An object is a collection of set of data known as



member data and the functions that operate on these data known as member function.

- OOP follows bottom-up design technique.
- Class is the major concept that plays important role in this approach. Class is a template that represents a group of objects which share common properties and relationships.

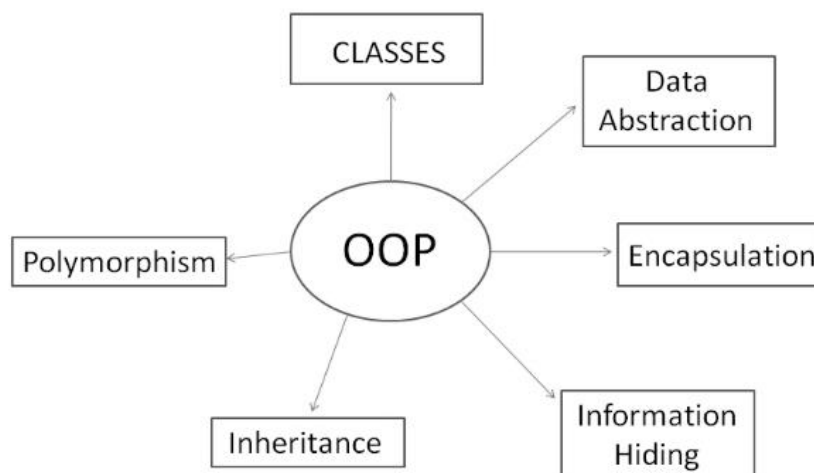
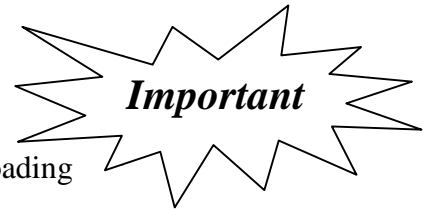
➤ **Difference between Procedural Programming & Object Oriented programming:**

Procedural Programming	Object Oriented Programming
Large programs are divided into smaller programs known as functions	Programs are divided into objects
Data is not hidden and can be accessed by external functions	Data is hidden and cannot be accessed by external functions
Follow top down approach in the program design	Follows bottom-up approach in the program design
Data may communicate with each other through functions	Objects may communicate with each other through functions.
Emphasize is on procedure rather than data	Emphasize is on data rather than procedure

➤ **Basic Concepts of OOP's:**

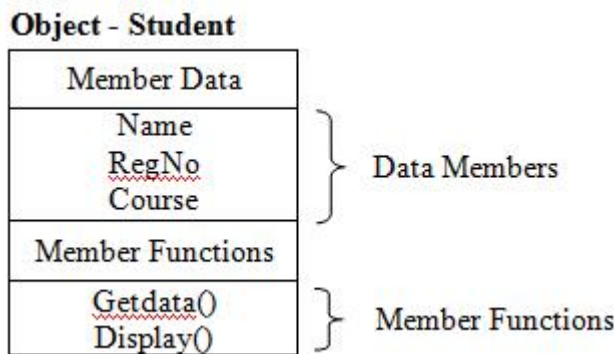
The following are the major characteristics of OOP's:

- Objects
- Class
- Data abstraction
- Data encapsulation
- Inheritance
- Overloading
- Polymorphism
- Dynamic Binding
- Message Passing



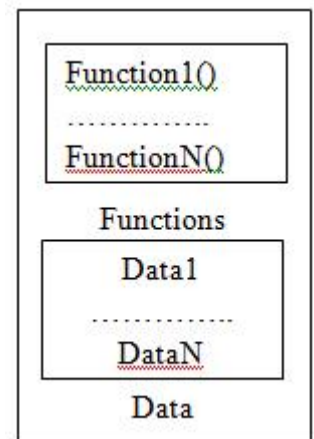
➤ **Objects**

- Objects are basic building blocks for designing programs.
- *An object is a collection of data members and associated member functions.*
- An object may represent a person, place or a table of data.
- Each object is identified by a unique name. Each object must be a member of a particular class.
- Example: Apple, orange, mango are the objects of class fruit.
- Objects take up space in memory and have address associated with them.
- At the time of execution of a program, the objects interact by sending the messages to one another.
- The objects can interact with one another without having to know details of data or functions within an object.



➤ **Classes:**

- The objects can be made user defined data types with the help of a class.
- *A class is a collection of objects that have identical properties, common behavior and shared relationship.*
- Once class is defined, any number of objects of that class is created.
- Classes are user defined data types. A class can hold both data and functions.
- For example: Planets, sun, moon are member of class solar system.

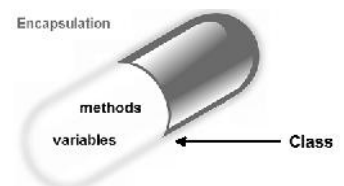


➤ **Data Abstraction:**

- *Data Abstraction refers to the process of representing essential features without including background details or explanations.*

➤ **Data Encapsulation:**

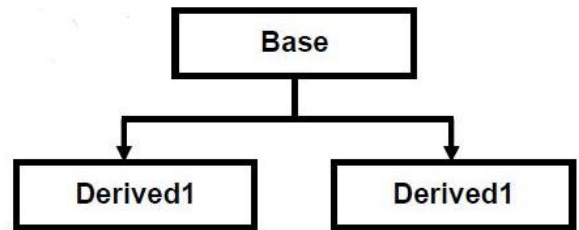
- *The wrapping of data and functions into a single unit (class) is called data encapsulation.*
- Data encapsulation enables data hiding and information hiding.



- **Data hiding** is a method used in object oriented programming to hide information within computer code.

➤ Inheritance:

- **Inheritance is the process by which one object can acquire and use the properties of another object.**
- The existing class is known as **base class** or super class.
- The new class is known as **derived class** or sub class.
- The derived class shares some of the properties of the base class. Therefore a code from a base class can be reused by a derived class.

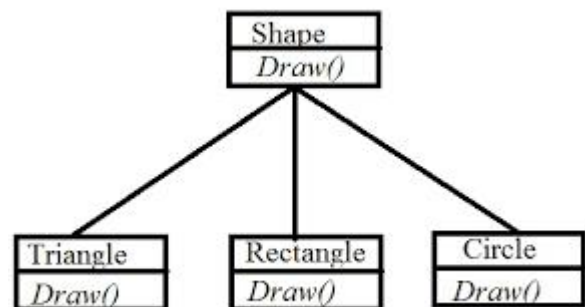


➤ Overloading:

- **Overloading allows objects to have different meaning depending upon context.**
- There are two types of overloading viz.
 - Operator Overloading
 - Function Overloading
- When an existing operator operates on new data type is called **operator overloading**.
- **Function overloading** means two or more function have same, but differ in the number of arguments or data type of arguments.

➤ Polymorphism:

- **The ability of an operator and function to take multiple forms is known as Polymorphism.**
- The different types of polymorphism are operator overloading and function overloading.



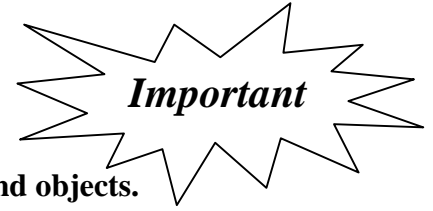
➤ Dynamic binding:

- Binding is the process of connecting one program to another.
- Dynamic binding is the process of linking the procedure call to a specific sequence of code or function at run time or during the execution of the program.

➤ Message Passing:

- In OOP's, processing is done by sending message to objects.
- A message for an object is request for execution of procedure.

- *Message passing involves specifying the name of the object, the name of the function (message) and the information to be sent.*



➤ Advantage of OOP's

- The programs are modularized based on the **principles of classes and objects**.
- Linking code & object allows related objects to share common code. This reduces **code duplication and code reusability**.
- Creation and implementation of OOP code is **easy and reduces software development time**.
- The concept of data abstraction separates **object specification and object implementation**.
- Data encapsulated along with functions. Therefore external non-member function cannot access or modify data, thus **proving data security**.
- Easier to develop complex software, because **complexity can be minimized through inheritance**.
- **OOP can communicate through message passing** which makes interface description with outside system very simple.

➤ Disadvantage of OOP's

- **Larger program size:** OOP's typically involves more lines of code than procedural programs.
- **Slower Programs:** OOP's typically slower than procedure based programs, as they typically require more instructions to be executed.
- Not suitable for all types of programs.
- To convert a real world problem into an object oriented model is difficult.
- OOP's software development, debugging and testing tools are not standardized.
- Polymorphism and dynamic binding also requires processing time, due to overload of function calls during run time.

➤ Application of OOP's

- Computer graphics applications.
- .CAD/CAM software
- Object-oriented database.
- User-Interface design such as windows
- Real-time systems.
- Simulation and Modeling
- Artificial intelligence and expert systems.
- Client-Server Systems.

CHAPTER 6 – Basic Concept of OOP's BLUE PRINT				
VSA (1 marks)	SA (2 marks)	LA (3 Marks)	Essay (5 Marks)	Total
-	01 Question	-	01 Question	02 Question
-	Question No 13		Question No 30	07 Marks

Important Questions

2 Marks Question:

1. Explain: Classes, Objects, Data Abstraction, Data Encapsulation, Inheritance, and Polymorphism.
2. What is Base class and derived class?

5 Marks Question:

1. Distinguish between procedural and object oriented programming.
2. Explain the characteristics of OOP's.
3. Briefly explain the basic concepts of OOP's.
4. Explain the advantages of OOP's.
5. Mention disadvantages of OOP's.
6. Write the applications of OOP's.