Module 1

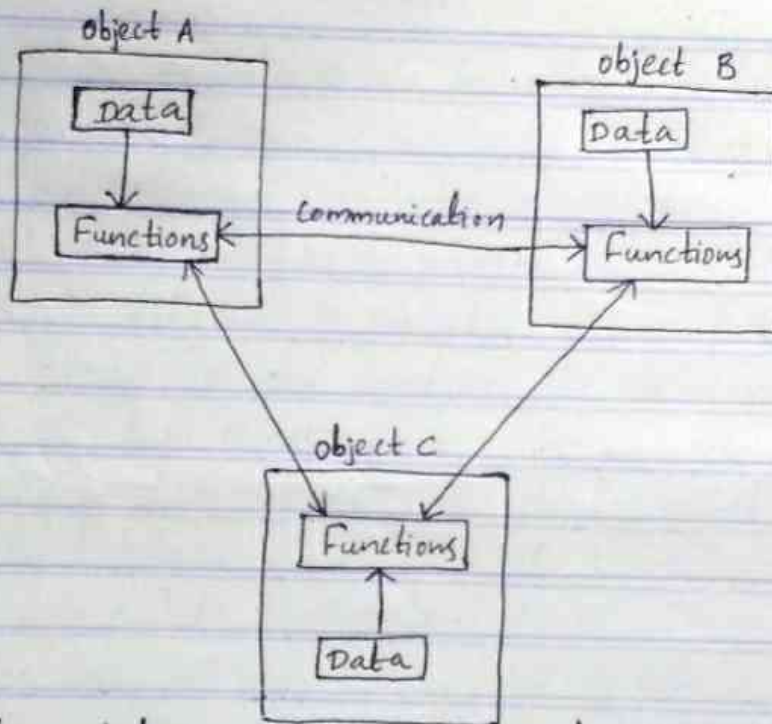### Object oriented Programming (OOP)

OOP allows decomposition of a problem into a number of entities called <u>objects</u> and then builds <u>data</u> and <u>functions</u> around these objects.

The organization of data and functions in OOP is shown below :—

The data of an object can be accessed only by the functions associated with that object. However, functions of one object can access the functions of other objects.

<u>Some of the features of OOP are</u> :—

1) Emphasis is on data rather than Procedure.
2) Programs are divided into what are known as objects.
3) Data structures are designed such that they characterize the objects.
4) Functions that operate ~~data~~ on the data of an object are tied together in the data structure.
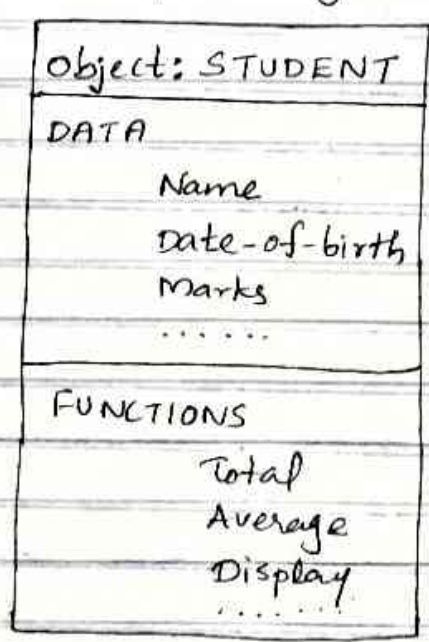5) Data is hidden and can not be accessed by external functions.

6) objects may communicate with each other through functions.

7) New data and functions can be easily added whenever necessary.

8) Follows bottoms-up approach in program design.

Characteristics of object oriented Programming

1) objects — objects are the basic run-time entities in an object-oriented system. They may represent a person, a place, a bank account, a table of data or any item that the program has to handle.

When a program is executed, the objects interact by sending messages to one another. For Eg:- if "customer" and "account" are two objects in a program, then the customer object may send a message to the account object requesting for the bank balance. Each object contains data, and code to manipulate the data.

The way of representing an object is shown below.

```
+--------------------------+
| Object: STUDENT          |
+--------------------------+
| DATA                     |
|        Name              |
|        Date-of-birth     |
|        Marks             |
|        . . . . . .       |
+--------------------------+
| FUNCTIONS                |
|        Total             |
|        Average           |
|        Display           |
|        . . . .           |
+--------------------------+
```

2) classes — The entire set of data and code of an object can be made a user-defined data type with the help of a class. objects are variables of the type

class. once a class has been defined, we can create any number of objects belonging to that class.

If fruit has been defined as a class, then the statement

fruit mango;

will create an object mango belonging to the class fruit.

3) <u>Data Abstraction</u> - Abstraction refers to the act of representing essential features without including the background details or explanations.

Since the classes use the concept of data abstraction, they are known as Abstract Data Types (ADT)

4) <u>Data Encapsulation</u> - The wrapping up of data and functions into a single unit (called class) is known as encapsulation. It is the most striking feature of a class. The data is not accessible to the outside world, and only those functions which are wrapped in the class can access it.

5) <u>Inheritance</u> - Inheritance is the process by which objects of one class acquire the properties of objects of another class. It supports the concepts of hierarchical classification. For Eg:- The bird 'robin' is a part of the class 'flying bird' which is again a part of the class 'bird'.

In OOP the concept of inheritance provides the idea of reusability. This means that we can add additional features to an existing class without modifying it.

6) <u>Polymorphism</u> - Polymorphism, means the ability to take more than one form. An operation may exhibit different behaviours in different instances. The behaviour depends upon the types of data used in the operation. For Eg:- consider the operation of addition. For two numbers, the operation will generate a sum. If the operands are strings,

then the operations would produce a ~~thi~~ third string by concatenation.

The process of making an operator to exhibit different behaviours in different instances is known as <u>operator overloading</u>. Using a single function name to perform different types of task is known as <u>function overloading</u>.

Advantages of OOP

1) Through inheritance, we can eliminate redundant code and extend the use of existing classes.

2) We can build programs from the standard working modules that communicate with one another, rather than having to start writing the code from scratch.

3) The principle of data hiding helps the programmer to build secure programs.

4) It is possible to map objects in the problem domain to those in the program.

5) It is easy to partition the work in a project based on objects.

6) Object-oriented system can be easily upgraded from small to large systems.

7) Software complexity can be easily managed.

Applications of OOP

The promising areas for application of OOP includes:

1) Real time systems - Here processing is done within a specified time

2) Simulation & modeling - means imitation of a situation or process.

3) object-oriented databases - Here information is represented in the form of objects.

4) Hypertext, hypermedia and expertext - links, multi-media, Sound & medias are included here.

5) AI (Artificial Intelligence) and expert systems- expert system emulates the decision making ability of a human expert.

6) Neural networks & parallel programming - Neural networks is a computer model for human brain & nervous systems. Simultaneous operation of 2 or more processors is parallel programming

7) Decision support and office automation System.

8) CIM / CAD system - Computer Integrated Manufacturing / Computer Aided Design.