

Unit 1: Introduction to Object Oriented Programming [3hrs]

Software Crisis:

Software crisis was a term used in the early days of computing science for, the difficulty of writing useful and efficient programs in the required time.

The software crisis was due to the rapid increase in computer power and the complexity of the problems that could be tackled. With the increase in the complexity of the software, many software problems arose because existing methods were neither sufficient nor up to mark.

Following points should be considered to address issues of crisis:

- a) How to represent real life entities of problem in system design.
- b) How to design system with open interfaces
- c) How to improve quality of software
- d) How to manage time schedules
- e) How to industrialize the software development.

Software Oriented program as a new paradigm:

The meaning of paradigm is an example or model. The programming paradigm is a way of conceptualizing what it means to perform computation and how task to be carried out on a computer should be structured and organized. The technique used in OOP gives an easy way to address problem in everyday life.

Object oriented Programming

The major motivating factor in the invention of OOP is to remove some of the flaws encountered in the procedural approach. OOP treats data as a critical element in the program development and does not allow it to flow freely around the system. OOP allows decomposition of problem into a number of entities called objects and then build data and functions around these objects. The data of an object can be accessed only by the function associated with the object. However, functions of one object can access the functions of other objects. OOP employ bottom-up approach in program design.

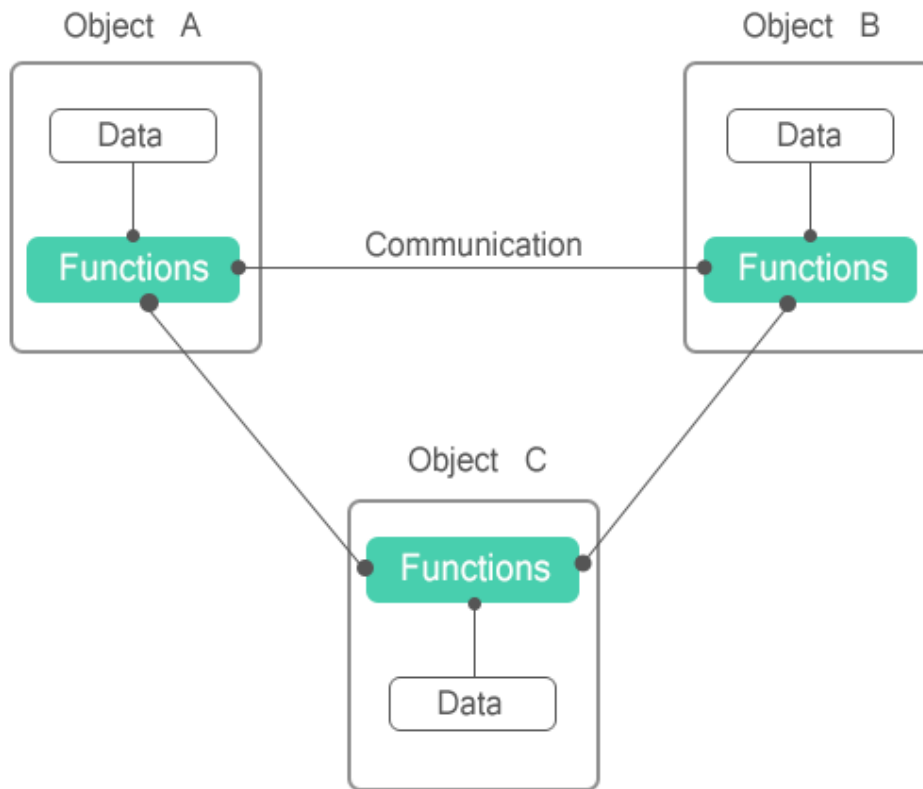


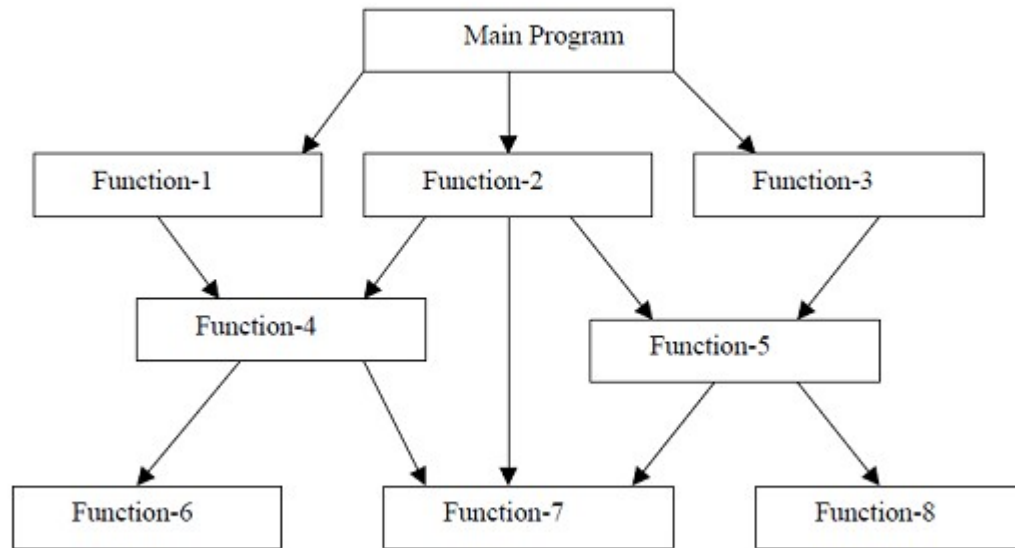
Fig: Organizational data and functions in OOP

Features of Object-Oriented Programming (OOP)

- Emphasis is on data rather than procedure
- Programs are divided into objects
- Data structures are designed such that they characterized the objects.
- Data are hidden and cannot be accessed by external features.
- Objects may communicate with each other through functions
- New data and functions can be easily added whenever necessary.
- Follows bottom-up approach in program design.

Procedural oriented programming

Conventional programming using high level programming is commonly known as POP. In the procedure-oriented approach, the problem is viewed as a sequence of things to be done such as reading, calculating and printing. POP basically consists of writing a list of instructions for the computer to follow and organizing these instructions into groups known as functions. These functions share global data and can transfer data from one form to another. POP employs top-down approach in program design.



Structure of procedural oriented programs

Features of procedural oriented programming (POP)

- Emphasis is on doing things algorithm
- Large programs are divided into smaller programs known as function
- Most of the functions share global data
- Data move freely (openly) around the system from function to function
- Function transfer data from one form to another
- Employs top-down approach in programming.

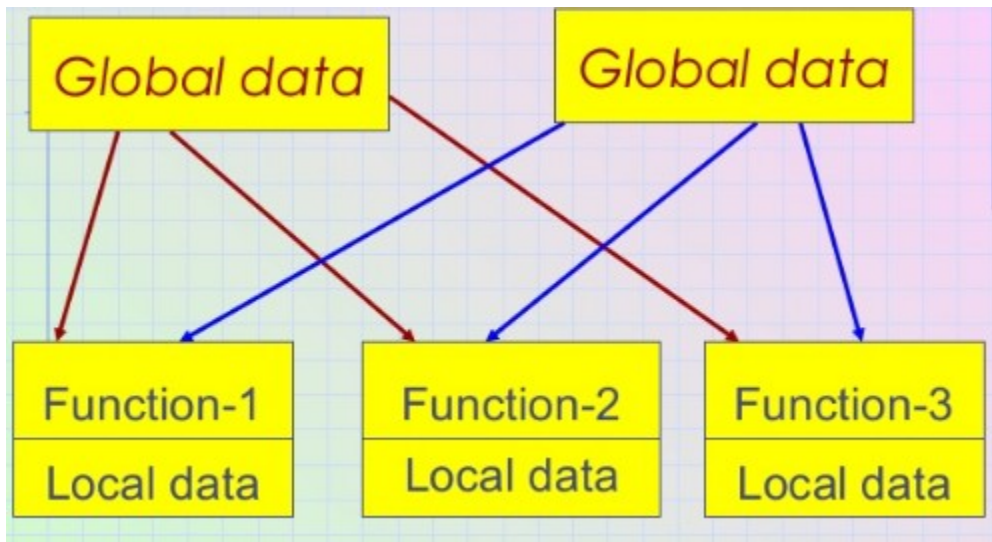


Fig: Relationship of data and functions in POP

Basic Concepts of Object-Oriented programming:

1.) Object:

It represents the entities of the real world. For e.g. Bank, Person, Company etc. They have attributes and function. For e.g. Attributes like color, address, name etc. The function like working, speaking, public relation. In OOP, the problem is always analyzed in terms of object communication through function i.e. the functions are used when some message needs to be passed and worked out accordingly

2.) Classes:

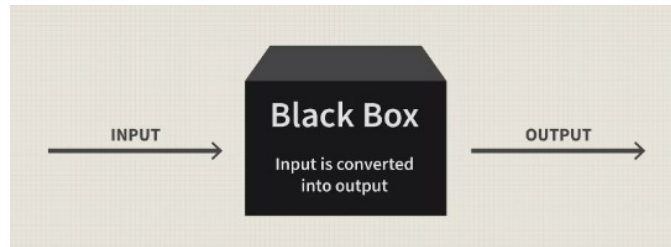
The group of objects that share common properties.

3.) Abstraction:

The process of representing essential features without internal details. The classes use the concept of abstract attributes such as size, weight, cost and the function to operate these attributes. The attributes are called data members and the function or methods as member function. Since the classes use the data abstraction, they are known as abstract data-type.

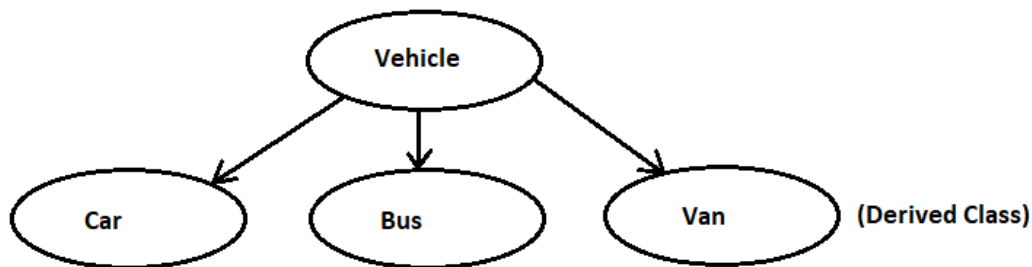
4.) Encapsulation:

The hiding of data and function in a single unit is known as encapsulation. It is not accessible to object, treats like a discrete black box testing.



5.) Inheritance:

The mechanism of deriving from its base class. OR it is a process by which the objects of one class acquire the properties of objects of another class. It supports the hierarchy of classification.



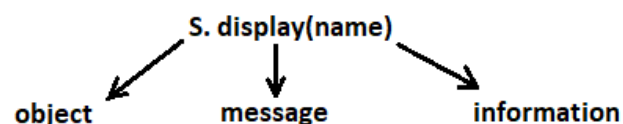
6.) Polymorphism:

It stands for ability to make more than one. In C++, the operators and a methods area different depending on what they are operating. E.g. When an existing as operator '+' or '=' is given, the capability to operate on a new data type, it is said to be operator overloading. Similarly, if a function call 'sum' can have two integers but if we define a function with same name, but taking its string as argument, we can add them together. Then it is called the function overloading.

e.g. `sum (int a, int b)`
 `sum(char N1[20], char N2[20])`

7.) Message Passing:

It is the way of communication in a similar way that we human do. It involves the name of message, information to be send.



8.) Dynamic Binding:

Binding refer to linking of a procedure call to the code to be executed in response to the call. It is also called late binding as the code associated to the given procedure call is not known until the time of a call run time. It is associated with polymorphism and inheritance.

Class and Structure

C++ supports all the features of structures as defined in C. But C++ has expanded its capabilities further to suit its oop philosophy. It attempts to bring the user-defined types as close as possible to the built-in data types, and also provides a facility to hide the data which is one of the major precepts of OOP. Inheritance, mechanism by which one type can inherit characteristics from other types, is also supported by C++.

In C++, a structure can have both variables and functions as members. It can also declare some of its members as 'private' so that they cannot be accessed directly by the external functions.

In c++, the structure names are stand-alone and can be used like any type names. In other words, the keyword 'struct' can be omitted in the declaration of structure variable. For example, we can declare the student variable A as :

Student A; // C++ declaration

Remember, this is an error in C.

C++ incorporates all these extensions in another user-defined type known as class. There is very little syntactical difference between structures and classes in C++ and therefore, they can be used interchangeably with minor modifications.

Since class is a specially introduced data type in C++, most of the C++ programmers tend to use the structures for hiding only data and classes to hold both the data and functions.

The only difference between a structure and a class in C++ is that, by default, the members of a class are private, while by default, the members of a structures are public.

Structured Programming Language

Structured Programming Approach can be defined as a programming approach in which the program is made as a single structure. It means that the code will execute the instruction by instruction one after the other. It doesn't support the possibility of jumping from one instruction to some other with the help of any statement like GOTO, etc. Therefore, the instructions in this approach will be executed in a serial and structured manner. The languages that support Structured programming approach are:

- C
- C++
- Java etc.

The structured program mainly consists of three types of elements:

- Selection Statements
- Sequence Statements
- Iteration Statements

The structured program consists of well-structured and separated modules. But the entry and exit in a Structured program is a single-time event. It means that the program uses single-entry and single-exit elements. Therefore, a structured program is well maintained, neat and clean program. This is the reason why the Structured Programming Approach is well accepted in the programming world.