

Chapter: 1 Introduction to OOP

1.1 Introduction

Traditional programming language such as COBOL, FORTRAN etc. are known procedure oriented language. Programs written in these languages had some disadvantages like the length of code was difficult to manage. Another main problem was related to data security.

In general the conventional programming language had the following disadvantages:

- Data was not secure as data passes globally from one function to another function globally.
- The length of the code became so large that it was difficult to manage.

1.2 Programming Paradigms

Generally following programming paradigms are frequently used:

i. Monolithic Programming:

- In this approach, data variables were global in nature.
- The statements were executed in sequence. However, the control can be transferred using JUMP statements as per the requirement.
- Main problem was that data was not secure due to its global scope.
- It was suitable for smaller application as concept of subroutines was not possible.
- Example: BASIC, ASSEMBLY

ii. Procedural Programming:

- In this method, the program was divided into number of segments.
- Here also, the data was not secure as once again the data was global.
- For control of program, again JUMP statements were used.
- It was useful for medium sized applications.
- Ex: FORTRAN, COBOL

iii. Structured Programming:

- It was useful for large sized applications.

- Here, the program is divided into number of sub modules. Each module is responsible for a separate task.
- Data is a bit more secure as the concept of local scope is introduced.
- This approach introduced the concept of data types which made the program more convenient.
- Ex: PASCAL, C Programming Language.

iv. Object Oriented Programming (OOP):

- Object Oriented Programming (OOP) language is a feature that allows a mode of modularizing the program by creating separate memory area for data as well as functions that will be used by different objects as per the requirement.
- Ex: C++, JAVA, smalltalk, charm ++ etc.

1.3 Evolution of C++

The most frequent OOP language is C++ which is considered as an extension of C programming language.

C++ was developed by Bjarne Stroustrup at AT&T Bell laboratory, New Jearsy (USA) in the early eighties. The various powerful features of SIMULA 67, C and ALGOL 68 were combined together in order to obtain a more powerful language which could support OOP. The result was in the form of C++.

Initially it was named as "C with Classes". In 1983, Rick Masciti gave the name C++ by borrowing the increment operator symbol (++) from C programming language. As C++ is an extension if C, all features of C language is also valid in C++.

1.4 Key Concepts of OOP language

OOP language is enriched with following salient features:

- i. **Object:** Any real world entity which has its own existence is known as object. An object has some certain features and behaviour.
- ii. **Class:** A class is a collection of objects which has identical properties and common behaviour.
- iii. **Data Encapsulation:** The packaging of data and functions into a single unit is known as data encapsulation. This is performed using classes. By using encapsulation, we can hide the implementation details from the user.

Users will only use the application without bothering about its implementation details.

- iv. **Data Abstraction:** Data abstraction is the process of representing the essential features without including the background details. Data abstraction is achieved by encapsulation.
- v. **Inheritance:** OOP provides the facility of creating a new thing from existing thing. This process is known as inheritance. During inheritance, the newly created thing adopts the features of existing things. Inheritance provides reusability of code.
- vi. **Polymorphism:** It is an important feature of OOP. It is created using two greek words POLY and MORPH. POLY means many and MORPH means forms. So polymorphism means same thing in multiple forms. Polymorphism allows the same function to act differently in different situation as per the requirement.
- vii. **Genericity:** Generally the program is data specific i.e. the input for the program is dependent on the data type. The main purpose of efficient programming is to create a program which is independent from the input data type. This is known as generic programming and the feature is known as genericity. It allows the user to create a function or class which may work on any type of input. It can be achieved using Templates feature available in C++.

1.5 Advantages of OOP

OOP provides many advantages to the users. Some of the advantages are as below:

- It can be easily upgraded.
- Data hiding can be done using encapsulation which helps the developers to create safe programs.
- Code reusability reduces the time and effort in writing large codes. It can be achieved using inheritance.

1.6 Applications of OOP

OOP has changed the thought process of a programmer which allows them to develop software in an efficient environment. Due to this fact, OOP is widely used in different areas, few of them are Object oriented DBMS, Office automation software, CAD/CAM software, Network programming etc.