

# About C++

---

Developments in software technology continue to be dynamic. New tools and techniques are announced in quick succession. This has forced the software engineers and industry to continuously look for new approaches to software design and development, and they are becoming more and more critical in view of the increasing complexity of software systems as well as the highly competitive nature of the industry.

These rapid advances appear to have created a situation of crisis within the industry. The following issues need to be addressed to face this crisis:

- How to represent real-life entities of problems in system design?
- How to design systems with open interfaces?
- How to ensure reusability and extensibility of modules?
- How to develop modules that are tolerant to any changes in future?
- How to improve software productivity and decrease software cost?
- How to improve the quality of service?
- How to manage time schedules?
- How to industrialize the software development process?

Many software products are either not finished, or not used, or else are delivered with major errors.

To reduce such errors software technology evolution gradually increases. Since, The invention of the computer, many programming approaches has been tried. These include techniques such as modular programming and structured programming.

## **Procedure-Oriented Programming**

Conventional programming, using high-level languages such as COBOL, FORTRAN and C, is commonly known as procedure-oriented programming (POP).

### **Problems with POP**

- In the procedure-oriented approach, the problem is viewed as a sequence of things to be done such as reading, calculating and printing.
- In a large program it is very difficult to identify what data is used by which function.
- Another serious drawback with the procedural approach is that it does not model real world problems very well. This is because functions are action oriented and do not really corresponds to the elements of the problem

## Object–Oriented Programming

The major motivating factor in the invention of object-oriented approach 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. It ties data more closely to the functions that operate on it and protects it from accidental modification from outside functions.

Object – oriented programming is the most recent concept among programming paradigms. We define “object-oriented programming as an approach that provides a way of modularize problems by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand.

### Characteristics of OOP languages

1. **Object:** Objects are entities, which can exist individually. It has its own properties and methods, where properties define the outlook of the object and methods define their procedures.
2. **Class:** It is a template used to define different objects of same type.
3. **Encapsulation:** The data and the methods, which operate on the data, are combined and placed in a group, this phenomenon is known as encapsulation, and the group is known as the object.
4. **Abstraction:** It means hiding of the data of one object of a class from another object of the same class.
5. **Inheritance:** Inheritance is the property by which an existing class can be used to create new classes, by deriving all the properties and methods of the old class to the new class and also adding new properties /methods in the new class. The old class is known as a base class or super class. The new class is known as derived class or sub class.
6. **Polymorphism:** Polymorphism means “One interface and multiple methods” i.e. one interface can be used to provide different functionalities.

There are two kinds of polymorphism:

- a) **Compile time polymorphism:** It is also known as early binding, as the interfaces are bind with their methods at compile time. It is accomplished using function overloading and operator overloading.
- b) **Run time polymorphism:** It is also known as late binding. In run time polymorphism the interface and its method bind at the time of execution. It is accomplished using virtual function.

**7. Multiple Inheritance:** When two or more classes are used to define a single class, then it is known as multiple inheritance.

### Benefits of OOP

Object oriented contributes to solution of many problems associated with the development quality of software products. The new technology promises greater programmer productivity, better quality of software and lesser maintenance cost. The principal advantages are:

- Through inheritance, we can eliminate redundant code and extend the use of existing classes.
- We can build programs from the standard working modules that communicate with one another rather than having to start writing the code from scratch. This leads to saving of development time and higher productivity.
- The principle of data hiding helps the programmer to build secure programs that cannot be invaded by code in other parts of the program
- It is possible to have map objects in the problem domain to those in the program.
- It is easy to partition the work in a project based on objects.
- The data centered design approach enables us to capture more details of a model in implemental form,
- Object-oriented systems can be easily upgraded from small to large systems.
- Message passing techniques for communication between objects make the interface descriptions with external systems much simpler.

Software complexity can be easily managed.