



**IJIRCCCE**

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 2, February 2024

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

**Impact Factor: 8.379**



9940 572 462



6381 907 438



ijircce@gmail.com



www.ijircce.com

# Review on Object-Oriented Programming

<sup>1</sup>Kshama Shukla, <sup>2</sup>Swati Dangi, <sup>3</sup>Dr. Aruna Sengar

M. Sc(CS) Student, Department of Computer Science, SAM Girls College, Bhopal, India<sup>1&2</sup>

Assistant Professor, Department of Computer Science, SAM Girls College, Bhopal, India<sup>3</sup>

**ABSTRACT:** For the growth of software industry in future and the advance of software engineering, use of object oriented programming (OOP) has increased in the software real world. Some the important features that's know is compulsory and that's features are important to study the depth knowledge of object oriented programming in this paper, we study the concept of object-oriented programming and its features, advantages, disadvantages, and we also know the constructor and destructors.

**KEYWORDS:** Programming Languages, OOP.

## I. INTRODUCTION

In 1980 Bjarne Stroustrup started working on new language, called "C with Classes". This new language was the extension of C with new feature class. After some improvements and refinements this language has given name C++. With its all features and with the name C++ it was introduced in 1983. C++ OOPs aspect was inspired by a computer simulation language called simula67. Stroustrup adds OOP features to C without significantly changing the C component. Thus, C++ is a superset of C language, meaning that any valid C program is a C++ program too .The fundamental concept in OOP is that; a program is designed around the data being operated. The basic idea behind object-oriented languages is to combine both data and functions into a single unit called object. The power of object-oriented language is that the programmer can create modular, reusable code. The flexibility of program increases so programmer is able to change or replace modules of a program without disturbing other part of the program. Software development speed is increase. Programming using objects; that are close in the representation of real world objects. By including these some of the fundamental features of is OOPs are as follows. OOP is a strategy for writing software in which data and behaviour are package together as classes whose instances are objects. A class is a named software program representation for an abstraction, an abstraction is a named collection of attributes and behavior relevant to modelling a given entity for some particular purpose, an object is a distinct instance of a given category that is structurally identical to all different cases of that class. Software code in OOP is written to define classes, objects, and manipulate these objects.

## II. TECHNICAL DETAILS



Figure 1: Features of Object-Oriented Programming

### Class:

A Class is a user-defined data type that has data members and member functions. It is a collection of similar kind of objects. A class is a generic definition of an object. It is a blue print of an object. Class is an extension of structure used in C language. In the structure we can combine different data element as a single entity. Class is a user defined data

type in which we can declare variables as well as functions. Class is the very important part of object-oriented programming. The class is used to implement encapsulation, data abstraction, and data hiding. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. Class is an extension of structure used in C language. In the structure, we can combine different data element as a single entity. In the class, we can also combine different data elements as well as functions. The data elements of class are known as data members of the class and functions of the class are known as member functions of the class. Class is a user defined data type in which we can declare variables as well as functions or class can be described as a collection of data members along with member can be functions. Class is the very essential part of object- oriented programming. Important thing is that the syntax of structure and class in C++ is same. The syntax for defining class is as follows.

```
Class class_name
{
Private:
Variable declaration Function declaration Public:
Function declaration
};
```

The class specifies the type and scope of its members. The keyword class name is an abstract data type. The body of a class is Class indicates that the name which follows class name. The enclosed within the curly braces followed by a semicolon i.e. the end of a class specification. The body of a class contains declaration of variables and functions, collectively known as members. The variables declared inside a class are known as data members, and functions are known as member functions these members are generally grouped underneath two sections, private and public, which define the visibility of members. Object oriented programming uses modular programming using this data type called classes. Defining variables of a class data type is known as class instantiation or objects.

### Object:

Objects are basic run time entities in an object-oriented programming. Each object contains data and code to manipulate that data. Objects can have interaction barring having to understand details about the statistics or code. In structured programming a problem is approached by using dividing it into functions. Unlike this, in object-oriented programming the trouble is divided into objects. Thinking in phrases of objects as a substitute than functions makes the designing of program simpler.

### For example:

Int y; int is a class and y is an object of that class. From the int class, we can create several objects (variables). The int class indicates what kind of data an object of its type can hold and what operations (addition, subtraction, etc.) can be performed on this data. A class is thus a description of number of similar objects. It specifies what data and what functions will be included in objects of that class. Instead of standard class like int you can think of user-defined class like employee from which objects like, e1, e2, e3 can be created through a statement, employee e1, e2, e3; Objects basic run type entities of object-oriented programming. It may represent a person, a bank account or any item that the program must be handled. A class specification only declares the structure of objects and it must be instantiated in order to make use of the services provided by it. This process of creating objects or class variables of the class is called class instantiation. Actually object is variable of class through which we can access (or handled) the data and members of the class. Whenever object is created it takes space in memory for its different members. We can access member using object similar to the structure in C.

### Inheritance:

Derive the new class from the existing classes is called inheritance. Inheritance is nothing but the concept of reusability i.e. when we drive new class it includes all the feature of existing class and also it may add sum now features. In this case existing class from which we are driving new class is called base class and the new class is called as drive class. This concept is known as reusability in C++. Inheritance provides reusability of the existing class. It is always important if we could reuse something that already exist rather than trying to create the same thing again and again. This is the important concept of object- oriented programming. For example: we want to design a car and some one has already designed wheels of the car then we can inherit that concept in our design, so there is no need to think about to design wheels again We inherit the concept of wheel and think to design other parts of the car. Inheritance is a feature or a process in which, new classes are created from the existing classes. The new class created is called “derived class” or “child class” and the existing class is known as the “base class” or “parent class”. The derived class now is

said to be inherited from the base class. The class can inherit some or all of the properties of another class. The class from which the properties are inherited is called the base class or parent class or super class and the class which inherits the properties is called the derived class or child class.

**Types of inheritance:**

1. Simple inheritance
2. Multiple inheritance
3. Multilevel inheritance
4. Hierarchical inheritance
5. Hybrid inheritance.

**Polymorphism:**

Polymorphism means the ability to take more than one form. Polymorphism is one of the crucial features of the object-oriented programming. The word polymorphism is made up of two Greek words: "poly" and "morphism". "Poly" means many and "morphism" means forms, so polymorphism means many forms. In object forms. C++ has four mechanisms that help us to implement oriented programming way, it simply means one name multiple polymorphism as Function Overloading, Operator Overloading, Template and Virtual function.

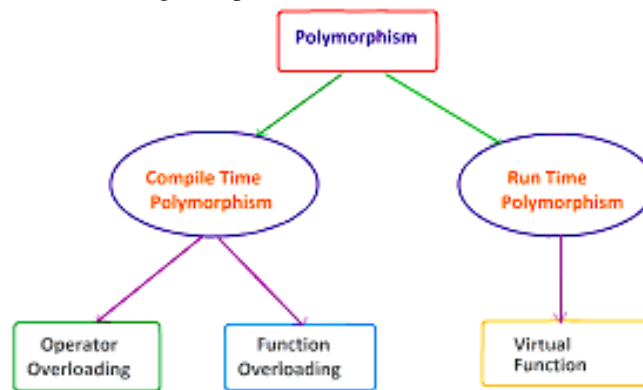


Figure 2: Polymorphism

**Abstraction:**

Data abstraction provides the foundation for object-oriented programming. In addition to providing fundamental data types, object-oriented programming languages allow us to define our own data types, called user-defined or abstract data types. In the C programming language, related data items can be organized into structures. These structures are capable operate only with data item. In C++, in addition to supplying this kind of data structure, also enable us to implement a set of operations that can be applied to the data elements. The data element and the set of operations applicable to the data element together shape the abstract data type. To guide data abstraction, a programming language should supply a assemble that can be used to encapsulate the data elements and operations that make up an abstract data type. In C++, this construct is known as a class. An instance of a class is called an object. Classes are composed of data elements called data members (member variables) and member functions (methods) that define the operations that can be carried out on the data members. In one sentence, the technique of creating new data types that are well suited to an application to be programmed is known as data abstraction.

**Encapsulation:**

A key idea in object-oriented programming (OOP) is encapsulation, which permits data hiding and gives classes a way to isolate their implementation details from their clients. C++ is an OOP language that provides several features for implementing encapsulation, such as access specifiers and member functions. In this post, we'll examine the C++ implementation of the encapsulation principle.

The act of encapsulating involves combining data and the methods that manipulate it into a single entity, referred to as a class. A class in C++ is a user-defined data type that has member methods and data members. The member functions are methods that work on the data members, which are variables that hold an object's state.

### Advantages of OOP:

OOP offers several benefits to the programmers as well as the user. Some advantages are as follows:

- Security is the first main advantage of OOP. The data and functions are combined together in the form of class.
- Using inheritance we can eliminate redundant code and extend the use of existing class.
- We can build programs from the standard working modules that communicate with one another. This leads to saving of development time and increase high productivity.
- The concept of data hiding helps the programmer to build secure programs, so that data of one class will be secure from the other parts of the program.
- Software complexity can be easily managed.
- Message passing teaching for communication Between objects makes the interface description with the external system much simpler form.

### Disadvantages of OOP:

The OOP languages have the following disadvantages.

- The message base communication between many objects in a complex system is difficult to trace and debug.
- It requires more memory to process at a great speed.
- It runs at slower than the traditional programming languages.
- It works on objects and everything of the real world is not possible to divide into neat classes and subclasses.
- The reusability of code is not possible, and it requires extra overhead to develop a new code on the basis of the existing code.

### Application of OOP:

- Modelling and Simulation.
- Object Oriented Database.
- Hypertext and Hypermedia.
- Artificial Intelligence and Expert System.
- Parallel Programming and Neural Networks.
- Decision support and Office Automation Systems.
- Use in programming languages like C++, JAVA, Python etc.
- Real Time System.
- Client Server System.
- CIM/CAM/CAD Systems.

### Constructors:

- Constructor in C++ is a special method that is invoked automatically at the time of object creation. It is used to initialize the data members of new objects generally. The constructor in C++ has the same name as the class or structure. It constructs the values i.e. provides data for the object which is why it is known as constructor.
- Constructor is a member function of a class, whose name is same as the class name.
- Constructor is a special type of member function that is used to initialize the data members for an object of a class automatically, when an object of the same class is created.
- Constructor is invoked at the time of object creation. It constructs the values i.e. provides data for the object that is why it is known as constructor.
- Constructor do not return value, hence they do not have a return type.

### Destructors:

Destructor is an instance member function that is invoked automatically whenever an object is going to be destroyed. Meaning, a destructor is the last function that is going to be called before an object is destroyed.

- A destructor is also a special member function like a constructor. Destructor destroys the class objects created by the constructor.
- Destructor has the same name as their class name preceded by a tilde (~) symbol.
- It is not possible to define more than one destructor.
- The destructor is only one way to destroy the object created by the constructor. Hence destructor can-not be overloaded.

- Destructor neither requires any argument nor returns any value.
- It is automatically called when an object goes out of scope.
- Destructor release memory space occupied by the objects created by the constructor.
- In destructor, objects are destroyed in the reverse of an object creation.

### III. CONCLUSION

This paper presents, what exactly the object-oriented programming (OOPs) is, there features and we also study the uses, advantages and disadvantages of OOPs in future. The programming languages before OOP system were not easy to understand. The large code converts into short code by using this OOPs concept. With the use of feature like class, objects, encapsulations, polymorphism, inheritance, and abstraction it can be seen that development of software is increase by using these capabilities.

### REFERENCES

- [1] Kaur, I., kaur, n., ummat, a., kaur, j., & kaur, n. (2016). Research paper on object oriented software engineering. international journal of computer science and technology, 36-38.
- [2] Kak, Avinash C. Programming with Objects, A Comparative Presentation of Object-Oriented Programming with C++ and Java, John Wiley, 2003. ISBN 0-471-26852-6.
- [3] Lafore, Robert, Object-Oriented Programming in C++, Fourth Edition, Sams Publishing, 2002. ISBN 0- 672-32308-7.
- [4] Seed, Graham M., An Introduction to Object-Oriented Programming in C++ with Applications in Computer Graphics, Second Ed., Springer-Verlag, 2001. ISBN 1-85233-450-9.
- [5] <https://www.geeksforgeeks.org/references-in-cpp/>Yevick, David, A First Course in Computational Physics and Object-Oriented Programming, Cambridge University Press, 2005. ISBN 0-521-82778-7.



Impact Factor: 8.379



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 9940 572 462  6381 907 438  [ijircce@gmail.com](mailto:ijircce@gmail.com)



[www.ijircce.com](http://www.ijircce.com)

Scan to save the contact details