Research Paper on Object-Oriented Programming (OOP)

Mr. RUSHIKESH S. RAUT

Student, Dept. of Information Technology, Prof. Ram Meghe Institute of Technology & Research, Badnera, Maharashtra, India

---***---

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:

Software Engineering, Software Development, Object -Oriented programming, Features of OOPs, Constructors, Destructors, C++ (programming language).

I. Introduction:

Programmers eventually discovered that it makes a program clearer and easier to understand if they were able to take a bunch of data and group it together with the functions that worked on that data. Such a grouping data and functions are called class and object. And writing programs by using classes is known as object-oriented programming.

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.

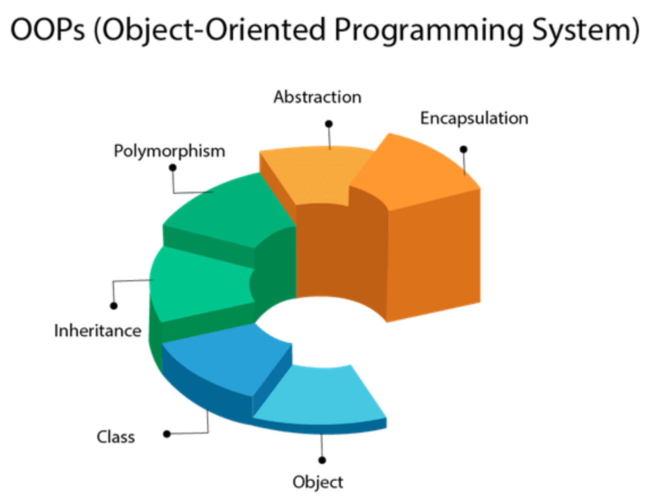
OOP is an strategy for writing software in which data and behaviour are package together as classes whose instances are objects. A class is a named software program representation for an abstraction, an abstraction is a named collection of attributes and behavior relevant to modeling a given entity for 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. Comparison with Structure Programming Language:

Structured programming languages like C define data structures (arrays, structures, unions, enums, etc.) and provide functions that inspect or change the data form any place in the program When the program grows beyond a reasonable size it becomes unmanageable since the data structures are available throughout the program and changing them in one part may have reaction on other part of the program.

Object-oriented programming reduces dependencies between different parts of a program. An object contains data structures and a set of operations for inspecting and manipulating them. All operations that require the knowledge of data structures are directly associated with the structures, rather than being spread throughout the program. Combining the data and the operations that inspect and modify the data brings in huge benefits. This arrangement ensures that you do not directly manipulate the data, instead you request functions associated with the data to do this job for you. Thus, part of the program that requests action to be performed on the data structures remains separate from the part which fulfills the request. As a result, now the parts of the program do not depend on each other through the data structures but through the functionality that the parts promise to provide. When you approach a programming problem in an object-oriented language you do not ask how the problem will be divided into functions. Instead you ask how it will be divided into objects.

III. Features of Object-Oriented Programming:



A. Class:

A class is a user defined data type which contains data members and member function to operate on those data member. 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. In the class we can also combine different data element as well as member function. Class is a user defined data type in which we can declare variables as well as functions. Class is the very essential part of object-oriented programming. The class is used to implement encapsulation, data abstraction, and data hiding.

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. The syntax for structure and class in C++ is same. The syntax for defining class is as follows.

```
Class class_name

{

Private:

Variable declaration

Function declaration

Public:

Variable declaration

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.

B. 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 x;
```

int is a class and x 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, el, 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.

C. Inheritance:

Derive the new class from the existing classes is called inheritance. Inheritance is nothing bat 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 basically done by creating new class, reusing the property of existing one. The mechanism of deriving a new class from an older one is called inheritance.

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.

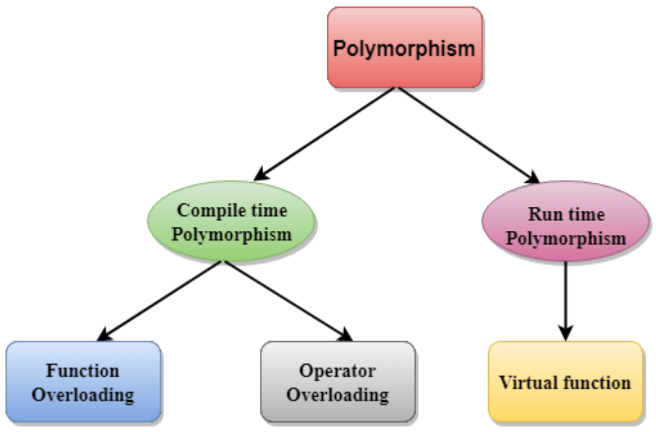
a. Types of inheritance:

The derived class may inherit some or all properties from the base class. A class can also inherit properties from more than one class or more than one level. In general inheritance in classified in five categories.

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

D. Polymorphism:

Polymorphism means the ability to take more than one from. 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.



E. 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. ln 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. ln one sentence, the technique of creating new data types that are well suited to an application to be programmed is known as data abstraction.

F. Encapsulation:

Encapsulation is mechanism that binds the data and functions together. This mechanism keeps both data and functions safe from outside interference and misuse. The advantage of encapsulated code is that user knows how to access it, there is no need of implementation details. In C++ points of view encapsulation is class. The purpose of the class is to encapsulate the complexity and hide the complexity of implementation inside the class. This insulation of the data from the direct access by the programmer is called data hiding. Actually some information of the object are made hidden from the outside world so that only the member functions of the same class can access the hidden data and no one from outside is allowed to access it. The information hiding is implemented using the three visibility modes as private, public and protected.

IV. 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.
- Massage passing teaching for communication Between objects makes the interface description with the external system much simpler form.

V. 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.

VI. Application of OOP:

Object oriented programming has application in many areas most popular application of object-oriented programming now has been in the area of interface design such as windows. There are thousands of windowing system developed using objects-oriented techniques.

The other promising areas for application of OOP are as follows.

- Real Time System.
- Modeling 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.

VII. Constructors:

Constructor is a special member function of a class which initializes itself when an object of class is created. It is special function because its name is same as the class name. The task of the constructor is to initializes the object of the class. It is called constructor because it constructor the value of data members. Some time it is known as dynamic initialization of object. Constructor is a special member function which enables, an object to initialize itself when it is created. This is known as automatic initialization of objects. Whenever we create an object of the class the constructor is automatically invoked.

Suppose we have a class with the name integer and having two private data member m and n. If we want to initialize m and n, then can define constructor as follows.

```
class integer

{

int m, n;

public:

integer (void); //constructor defined.

};

integer :: integer (void) //constructor body

{

M = 0;

N = 0;

}
```

Whenever we create an object of integer class, the constructor will initialize m and n automatically to zero. Hence the value m and n initialized to zero. There is no need to any other statement to invoke (call) the constructor.

VIII. Destructors:

Destructor is use to destroy the object that have been created by a Constructor. Like a constructor, the destructor is a member function whose name as the same as class name but is preceded by a tiled (). A Destructor never takes any argument and it does not return any value. It is invoked implicitly by the compiler while exit from the program.

Actually destructor clear or clean memory storage i.e. it Works as a garbage collector. It is good programming practice to declare destructor to release memory space for future use.

IX. Conclusion:

We have to study in above research paper, what exactly the object-oriented programming(OOPs) is, there features and we also study the use 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, l., 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] Svenk, Goran, Object-Oriented Programming: Using C++ for Engineering and Technology Delmar, 2003. ISBN 0-7668-3894-3.

[6] Yevick, David, A First Course in Computational Physics and Object-Oriented Programming, Cambridge University Press, 2005. ISBN 0-521-82778-7.

[7] Amit Verma, Navdeep Kaur Gill,"Image Processing and Watermark", International Journal of Computer Science Technology (JCST), Vol. 7, Issue 1, pp. 143-147, Jan - March 2016.

[8] Amit Verma, Navdeep Kaur Gill, "Analysis of Watermarking Techniques", International Journal of Computer Science and Technology (IJCST), Vol. 7, Issue 1, pp. 153-156, Jan- March 2016.