Analysis of Object Oriented Programming

VISHAL VAMANMEHTRE[1], AKSHAT BHALLA[2]
[1]*Assistant professor, Department of Electrical engineering, Bharati Vidyapeeth Deemed University, College of Engineering, Pune, India*
[2] *Department of Electrical engineering, Bharati Vidyapeeth Deemed University, College of Engineering, Pune, India*

Abstract- ***Object oriented programming is a programming approach based on concept of classes and objects which contains data in form of attributes or properties and procedures are thereby in the form of procedures known as methods or function. in object oriented programming the programs are made of objects and classes which interact with each other using message passing. Oop's language is diverse, the most popular are class based, means object is instance of class and class is blueprint from which object is created. Many of the most commonly used programming languages are Java, C++, Python and many more, are multi paradigm and they support the concept of object oriented programming to a greater or lesser extent.Object oriented languages include Java,C++, and Python etc. Like if we talk about a fruit like orange then the class would be oops and the object will be orange. Basically object oriented program introduced world to the concept of oops.***

Indexed Terms- Class, Object, Object Oriented Programming

I. INTRODUCTION

Object Oriented Programming (OOP) uses a different arrangement of programming language than old procedural programming approach (C and so forth.). It is a language which is mainly based on ideology that objects main objective is to convey and share the data. It changed the old monotony of old procedural programming where the main focus was on method of execution. The object oriented programming concept suggested a new idea of or a new path for giving significance to items, with four basic of object oriented programming. The first concept stated is of the object, the basic building block, the necessity of programming language. Any real world entity can be compared to an object. The next fundamental is class, a class is a blueprint from which an object is created. All the objects that are created comes under a class, it is the principle body of the system. Class forms the fundamental unit of the system. The third factor that comes into vision is Inheritance, where parent class can pass its information to the derived class. This factor contributes to the efficiency of a program because code need not to be written again and again which reduces time. By inheritance the code of derived class becomes error free. The last factor is polymorphism which is performing various operations at a single time.

II. DESIGNING PRINCIPLES

Object Oriented Programming is mainly based on six concepts.

- Class
- Object
- Inheritance
- Reusability
- Polymorphism
- Encapsulation
- Abstraction

Class-a blueprint from which objects are created. [1] Classes contain data and methods or function brought together under a single unit. In other words class is nothing just a bundle or a collection of objects that share data. When we define a class it just created a kind of template. So no memory is allocated whenever a class is created. Memory is only allocated by object.

Object- Object is nothing but an instance of a class. it is a basic fundamental unit of object oriented programming. [2] Object allows comparison of a real life entity to programming world. Objects in Oop's language like java can be interacted with by invoking methods.

Object consists of

1. State-it is represented by attributes of an object. It also reflects the properties of an object.
2. Behaviour- It is represented by methods of an object. It also reflects the response of an object with other objects.
3. Identity- It gives a unique name to an object and enables one object to interact with other objects.

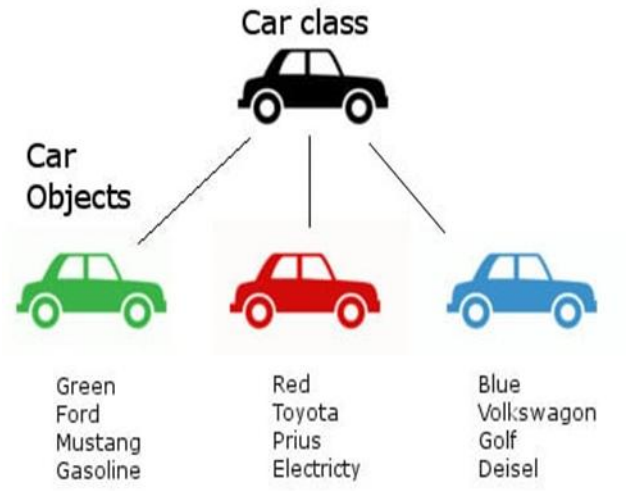

Fig 1.1

- Inheritance- Inheritance as it suggest is concept of inheriting or gaining properties of an already existing class to get new class or a number of classes. [3] In other words derived classes may have same properties as that of their parent base class.by taking an example we can consider the base parent class as tree steam and branches coming out from it can be considered as derived class. All the derived classes will have same properties as that of base class but some additional properties. The main advantage of inheritance in Object Oriented Programming is that it helps in reducing the size of code and hence reducing time of execution and increasing the efficiency. Inheritance is related to reusability.

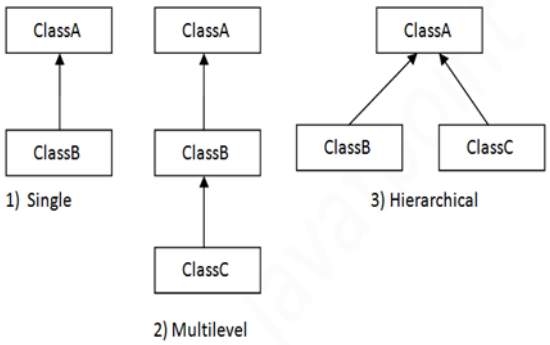

Fig 1.2

- Reusability- [4] Code reusability can also be achieved by inheritance as a derived class inherits its properties from parent class code once written can be used again and hence it will be error free. The term reusability itself points to reusing of code. Its reusability decreases the time and hence increases the efficiency of program [5] Reusability can also be using same code and adding some external features and characters to it.

- Polymorphism- Poly means many and thereby polymorphism as name suggest it means a certain item appearing in different ways. That is a function or operator to act in different ways or kind depending upon the place of its execution. Overloading means using and calling functions of same name but having different parameters. Overloading is also a kind of polymorphism.

- Program to demonstrate polymorphism.

```
public class Suma {

// Overloaded sum(). This sum takes two int parameters
public int sum(int x, int y)
{
return (x + y);
}

// Overloaded sum(). This sum takes three int parameters
public int sum(int x, int y, int z)
{
return (x + y + z);
}
```

```
// Overloaded sum(). This sum takes two double parameters
public double sum(double x, double y)
{
return (x + y);
}
// Driver code
public static void main(String args[])
{
Suma s = new Suma();
System.out.println(s.sum(10, 20));
System.out.println(s.sum(10, 20, 30));
System.out.println(s.sum(10.5, 20.5));
}
}
```

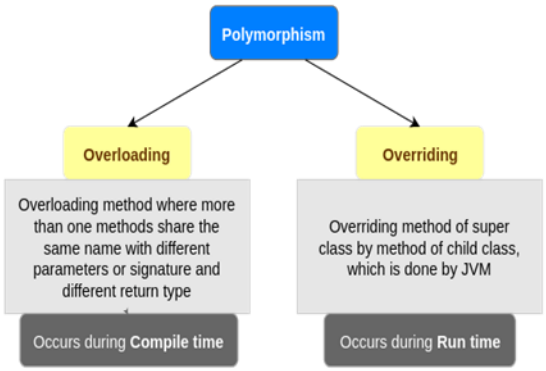

Fig 1.3

- Abstraction- Abstraction is one of the major concept of object oriented programming languages. Main goal of abstraction is to handle complexity by hiding unnecessary details from the programmer or user to reduce the complexity. This enables the user to implement more complex logic on top of the provided abstraction without even thinking or understanding about all unnecessary complex stuff.

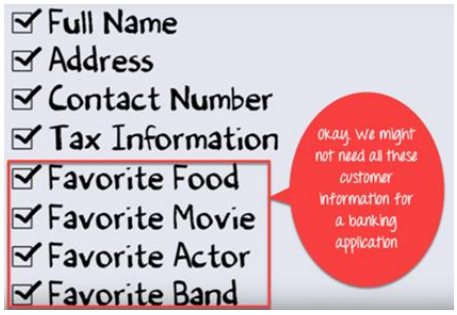

Fig 1.4

- Encapsulation- In normal terms Encapsulation is defined as wrapping up of data and information under a single unit and [6] In Object Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulates them and hiding details unnecessary.

Consider a real life example of encapsulation, in a company there are different sections like the accounts section, finance section, sales section and many others. The finance section handles all the financial transactions and keep records of all the data related to finance similarly the sales section handles all the sales related activities and keep records of all the sales. Now there may arise a situation when for some reason an official from finance section needs all the data about sales in a particular month. In this case, he is not allowed to directly access the data of sales section. He will first have to contact some other officer in the sales section and then request him to give the particular data. This is what encapsulation is. Here the data of sales section and the employees that can manipulate them are wrapped under a single name “sales section”.

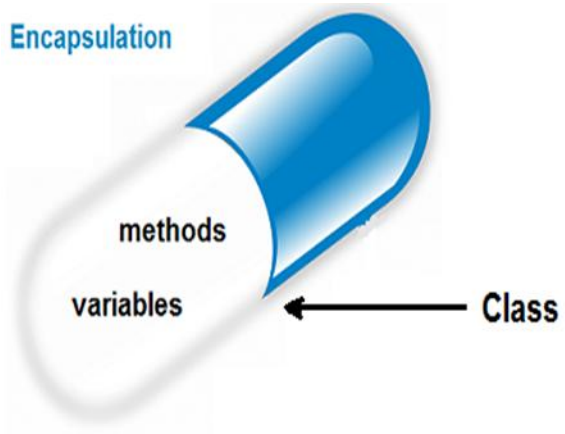

Fig 1.5

III. ADVANTAGES OF OBJECT ORIENTED PROGRAMMINGTHE MAIN ADVANTAGES OF OOPS ARE

a. The main advantage in oops is inheritance in which properties of base class is passes to the derived class which is also code reusability and minimizes code length.

b. The concept of encapsulation is a data hiding property to hide unnecessary details from the user so user can focus on program rather than details that are not important.
c. Polymorphism concept which is useful in overriding class properties and methods.
d. Abstract class concept is just a placeholder for a class and object cannot be created from it and its use comes in task and function call.
e. Multiple inheritance and composition concept where a single child class can inherit properties attributes data and information from more than one base class and reduces code length and execution time.

IV. CONCLUSION

The further research on object oriented programming shows the impact and importance of object oriented programming has in today's technology or programming world. Object oriented programming approach made a major change in programming world by breaking the monotony of procedural programming. Object which is the most important and the crucial fundamental unit of object oriented program, is basic of representing any information. The main objective of object is representing real life entity in programming world. Class is blueprint from which object is created, a prototype is essential for the structure of program. Every function and method present inside the class definition. Inheritance which gave the concept of code reusability made program easy, efficient and shorter. Hence thereby concluding that object oriented programming plays a vital role and its usage is being increased in every programming kind.

ACKNOWLEDGEMENT

We would like to express our special thanks of gratefulness to Dr. D.S. Bankar, Head, and Department of Electrical Engineering for their guidance and support for completing my research paper. I would also like to thank the faculty members of the department of electrical engineering who helped us with extended support.

REFERENCES

[1] "Object Oriented Programming and its concept" International Journal of Innovation and Scientific Research paper.2016.

[2] Neyer, F. FWu "Object Oriented Programming for flexible software". IEEE1990

[3] A.P.Flores , A. Nacul , L. Silva , J. Netto , L. Bacellar "Object Oriented Programming environment on real time applications" IEEE 1999

[4] 'The Design, Implementation and Use of Object-Oriented Systems' IEE 1994.

[5] LongshuLi, Yi Xu "The teaching research on a case of object-oriented programming" IEEE 2010.

[6] J.C. McKim "Using a multi-term project to teach object oriented programming and design." IEEE 1998.