Concepts Related to Object Oriented Program OOP'S: Basics

VISHAL VAMAN MEHTRE[1], UTKARSH RAJ VERMA[2]
[1] *Assistant Prof., Department of Electrical Engineering, Bharati Vidyapeeth Deemed University College of Engineering, PUNE*
[2] *Department of Electrical Engineering, Bharati Vidyapeeth Deemed University College of Engineering, PUNE*

Abstract- This paper gives us the basic information of "object oriented programming" (OOPS).It also Provides us the concept of object and basic parameters of OOPS .Such as Data abstraction Encapsulation modularity inheritance and polymorphism. OOPS classes tend to be overly generalized, which make relations among classes becomes artificial at time. The object oriented programs are tricky in design. So to program with OOPS one needs to have proper design skills, programming skills. Since OOPS codes are more near to real world models, the programmer must have to think in terms of object.

I. INTRODUCTION

OBJECT ORIENTED PROGRAMMIN (OOPS) is a programming paradigm based on the concept of objects, which contain data in the form of field and codes in the form procedure (method). The object oriented approach views a problem in terms of object involved rather than procedure for doing it. Now the question arises 'What is object'? Well an object is an identifiable entity with same characteristics and behavior.

1) OOPS languages are diverse but the most popular ones are class based meaning objects are instances of classes, which also determine their types. significant OOPS languages includes JAVA, C++, C/,PYTHON,PHP, JAVA SCRIPT, RUBY,PERL, OBJECT PASCAL, OBJECTIVE-C,DORT, SWIFT, SCADA, LISP,MATLAB AND SMALL TALK.

- CONCEPT:

The OOPS has been developed with a view to overcome the drawbacks of conventional programming approaches. These concepts are as follows:

II. DATAABSTRACTION

It refers to the act of representing essential features without including the background details or explanation [2].

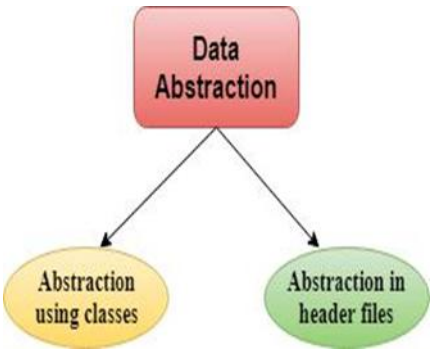

Fig1.1 Types of Abstraction

See a simple example of abstraction in header files.

- Program to calculate the power of a number:

```
#include <iostream.h>
#include<math.h>
Using namespace std;
int main ()
{
int n = 4;
int power = 3;
int result = pow(n,power);          // pow(n,power) isthe power function
std: cout << "Cube of n is: " <<result<< std::endl;
return 0;
}
```

Output:
Cube of n is: 64
In the above example, pow () function is used to calculate 4 raised to the power 3. The pow () function is present in the math.h header file inwhich all the implementation details of the pow () function is hidden.

III. ENCAPSULATION

The wrapping up of data/functions (that operates on the data) into a single unit (called class) is known as encapsulation. Or simply we can say that it is the way of combining both data and functions that operate on the data under a single unit [3].

EXAMPLE: In C++ encapsulation can be implemented using Class and access modifiers. Look at the below program:

```
#include<iostream.h>
Usingnamespacestd;
class Encapsulation
{
private:
//        data hidden from outside world Int x;
public:
//        function to set value of
//        variable x
Void set(int a)
{
x =a;
}
//        function to return value of
//        variable x
Int get()
{
Return x;
}
};
//  main function Int main()
{
Encapsulation obj;
obj.set(5);
cout<<obj.get();
return0;
}
```

Output: 5

IN the above program the variable **x** is made private. This variable can be accessed and manipulated only using the functions get() and set() which are present inside the class.

Thus we can say that here, the variable x and the functions get() and set() are binded together which is nothing but encapsulation.

IV. MODULARITY

It is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules. The act of partitioning a program into individual components is called modularity.

The justification for partitioning a program is that:-

- It reduces its complexity to some degree.
- It creates a number ofwell-defined documented boundaries within the program.

V. INHERITENCE

It is the capability of one class of things to inherit capabilities/properties from another class [4].

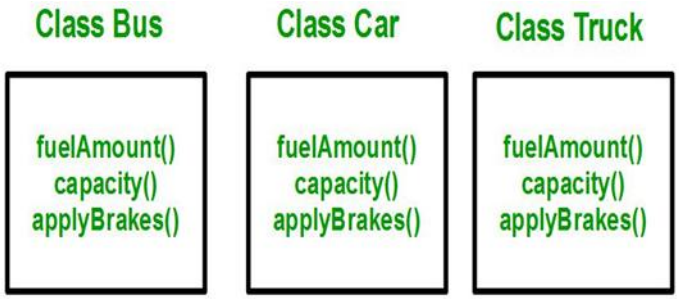

Fig1.2 Types of class

We can clearly see that above process results in duplication of same code 3 times. This increases the chances of error and data redundancy. To avoid this type of situation, inheritance is used. If we create a class Vehicle and write thesethree functions in it and inherit the rest of the classes from the vehicle class, then we can simply avoid the duplication of data and increase re-usability. Look at the below diagram in which the three classes are inherited from vehicle class:

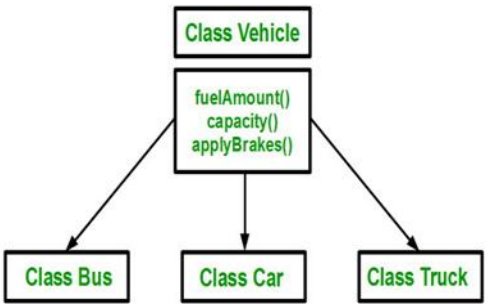

Fig1.3 Inheritance in Class

```
#include <bits/stdc++.h> usingnamespacestd;
//Base class
Class Parent
{
public:
int id_p;
};
//  Sub class inheriting from Base Class(Parent) Class
Child : public Parent
{
public: intid_c;
};
//main function
Int main ()
{
Child obj1;
//        An object of class child has all data members
//        and member functions of class parent
obj1.id_c = 7;
obj1.id_p = 91;
cout << "Child id is "<< obj1.id_c << endl; cout <<
"Parent id is "<< obj1.id_p << endl;
return0;
}
```

Output

```
Child id is 7
Parent id is 91
```

In the above program the 'Child' class is publicly inherited from the 'Parent' class so the public data members of the class 'Parent' will also be inherited by the class 'Child'.

VI. POLYMORPHISM

Polymorphism is a key feature of the object-oriented paradigm. However, [6] polymorphism induces hidden forms of class dependencies, which may impact software quality. In this paper we investigated about the impact of polymorphism in an object-oriented design.

It is the ability for a message or data to be processed in more than one form. It is basically of two types:

- Compile time Polymorphism
- Runtime Polymorphism

Let us have an example of Function Overloading which is sub type of Compile time Polymorphism:

```
#include <bits/stdc++.h>
usingnamespacestd;
class Geeks
{
public:
//      function with 1 int parameter Void func(int x)
{
cout << "value of x is "<< x << endl;
}
//      function with same name but 1 double
parameter Void func(double x)
{
cout << "value of x is "<< x << endl;
}
//      function with same name and 2 int parameters
Void func(int x, int y)
{
cout << "value of x and y is "<< x << ", "<< y <<
endl;
}
};
Int main() {
Geeks obj1;
//Which function is called will depend on parameter
//The first 'func' is called
obj1.func(7);
//      The second 'func' is called obj1.func(9.132);
//      The third 'func' is called obj1.func(85,64);
return0;
}
```

Output:
Value of x is 7
Value of x is 9.132

Value of x and y is 85, 64

In the above example, a single function named *func*tion acts differently in three different situations which is the property of polymorphism.

VII. ADVANTAGES AND DISADVANTAGES

- OOPS codes are nearer to real world models than other programming methodology codes.
- Encapsulation allows class definition to be reuse in other applications. The availability of a consistent interface to objects lessens codes duplication and there by improves code reusing ability.
- [5] Use of OOPS concept narrows down the search for problems in the programs .OOPS facilitates us for easy redesigning and extension of a program we can use same code and modify it asper our use.

Every coin has two sides. The same can be said for OOPS on one hand it has certain advantage over other programming methodology, but on the other hand it has also some disadvantages. It has been criticized for a number of reasons including not meeting its stated goals of reuseability and modularity and for over emphasizing one expects of software design and modeling (data and objects) at the expense other important aspects.

CONCLUSION

In our opinion OOPS classes tends to be in generalized form which make relations among classes becomes artificial at times. The object oriented programs are tricky in design. So to program with OOPS one needs to have proper design skills, programming skills. Since OOPS codes are more near to real world models, the programmer must have to think in terms of object.

ACKNOWLEDGEMENT

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

REFERENCES

[1] Sumita Arora,"Concept of C++" (Dhanpat Rai &co.)

[2] E Balagurusamy, "object Oriented Programming With C++",(Mc Grawhill Education)

[3] Prof. Dharminder Kumar, "Introduction of OOPS"

[4] Shivam, "A Study on Inheritance Using OOP with C++", International Journal of Advance research in computer science and management Studies, Issue 2, July, 2013.

[5] Ashwin Urdhwareshe, "Object-Oriented Programming and its Concepts", Issue 1 Aug, 2016.

[6] Saïda Benlarbi, "Polymorphism Measures for Early Risk Prediction", IEEE