Abstraction in Matlab and OOPS

VISHAL V. MEHTRE [1], SUBHROJIT SENGUPTA [2]

[1] *Assistant Professor, Department of Electrical Engineering, Bharati Vidyapeeth Deemed (To Be) University College of Engineering Pune, India*

[2] *Department of Electrical Engineering, Bharati Vidyapeeth Deemed (To Be) University College of Engineering Pune, India*

Abstract -- One of the foremost necessary facet is abstraction in Matlab and Object bound programming (OOPS). It solely displays essential data of information within the outside world. Its main objective is to cover unneeded contents and show the particular output. Abstraction is one in every of the key ideas of object-oriented programming (OOP) languages. Its main goal is to handle quality by concealment unneeded details from the user. That permits the user to implement a lot of complicated logic on prime of the provided abstraction while not understanding or maybe puzzling over all the hidden complicated.

I. INTRODUCTION

Data abstraction is one amongst the foremost essential and necessary feature of object homeward programming in C++. Abstraction suggeststhat displaying solely essential data and concealment the main points. Knowledge abstraction refers to providing solely essential data regarding the info to the surface world, concealment the background details or implementation.[1]

Consider a true life example of a person driving a automotive. the person solely is aware of that pressing the accelerators can increase the speed of automotive or applying brakes can stop the automotive however he doesn't fathom however on pressing accelerator the speed is really increasing, he doesn't fathom the inner mechanism of the automotive or the implementation of accelerator, brakes etc within the automotive. this is often

What abstraction is ?
Abstraction issues the ways that} by which higher, less real ideas ar derived from the usage and classification Of literal ("real" or "concrete") knowledge or alternative data. Abstractions could also be shaped by reducing the data content of a thought or a development as viewed, often to keep solely the data that is required for a selected purpose.[2 In philosophical word, abstraction is that the thought method whereby concepts are distanced from objects Abstraction uses a technique of simplification, wherever at one time concrete details are left ambiguous, vague, or undefined; so effective communication regarding things within the abstract needs associate degree intuitive or common expertise between the individual and therefore the communication recipient.

Abstractions generally have ambiguous referents; as an example, "happiness" (when used as associate degree abstraction) will sit down with as several things as there ar individuals and events or states of being that build them happy. Likewise, "architecture" refers not solely to the look of safe, useful buildings, however additionally to parts of creation and innovation that aim at elegant solutions to construction issues, to the utilization of area, associate degreed to the commit to evoke an emotional response within the builders, owners, viewers and users of the building.[3]

Abstraction using Classes:
We can implement Abstraction in C++ using classes. Class helps us to group data members and member functions using available access specifies. A Class can decide which data member will be visible to outside world and which is not.

Abstraction in Header files:
One more type of abstraction in C++ can be header files. For example, consider the pow() method present in math header file. Whenever we need to calculate power of a number, we simply call the function pow() present in the math. header file and pass the numbers as arguments without knowing the

underlying algorithm according to which the function is actually calculating power of numbers.[4]

Abstraction using access specifies:
Access specifier's square measure the most pillar of implementing abstraction in C++. we are able to use access specifiers to enforce restrictions on category members. For example:

• Members declared as public in a very category, are often accessed from anyplace within the program.

• Members declared as personal in a very category, are often accessed solely from inside the category. they're not allowed to be accessed from any a part of code outside the category.

We can simply implement abstraction exploitation the on top of 2 options provided by access specifiers. Say, the members that define the interior implementation are often marked as personal in a very category. and therefore the necessary info required to tend to the skin world are often marked as public. And these public members will access the personal members as they're within the category.[5]

Example:

```
#include <iostream>

classimplementAbstraction

{

   private:

      inta, b;

   public:

   // method to set values of

   // private members

   voidset(intx, inty)

   {

      a = x;

      b = y;

   }

      Void display()

   {

      cout<<"a = "<<a <<endl;

      cout<<"b = "<< b <<endl;

   }

};

 intmain()

{

   implementAbstractionobj;

   obj.set(10, 20);

   obj.display();

   return0;

}
```

Output:
a = 10
b = 20

Now you can see in the above program we are not allowed to access the variables a and b directly, however one can call the function set() to set the values in a and b and the function display() to display the values of a and b.

Advantages of Data Abstraction:
• Helps the user to avoid writing the low level code
• Avoids code duplication and increases reusability.
• Can change internal implementation of class independently without affecting the user.
• Helps to increase security of an application or program as only important details are provided to the user

[1] Abstract Classes

Abstract classes are useful for describing functionality that is common to a group of classes, but requires unique implementations within each class.

Abstract Class Terminology:

An abstract class serves as a basis (that is, a super class) for a group of related subclasses. An abstract class can define abstract properties and methods that subclasses implement. Each subclass can implement the concrete properties and methods in a way that supports their specific requirements.[5]

Implementing a Concrete Subclass:

A subclass must implement all inherited abstract properties and methods to become a concrete class. Otherwise, the subclass is itself an abstract class.
MATLAB does not force subclasses to implement concrete methods with the same signature or attributes.

Abstract classes:

- Can define properties and methods that are not abstract
- Pass on their concrete members through inheritance
- Do not need to define any abstract members Declare a class as abstract in the class def statement:
  Class def (Abstract) Abs Class

...
end
For classes that declare the Abstract class attribute:

- Concrete subclasses must redefine any properties or methods that are declared as abstract.
- The abstract class does not need to define any abstract methods or properties.

When you define any abstract methods or properties, MATLAB automatically sets the class Abstract attribute to true.[6]

Abstract Methods:

Define an abstract method:
methods (Abstract)
abs Method(obj)
end

For methods that declare the Abstract method attribute:

- Do not use a function. end block to define an abstract method, use only the method signature.
- Abstract methods have no implementation in the abstract class. Concrete subclasses are not required to support the same number of input and output arguments and do not need to use the same argument names. However, subclasses generally use the same signature when implementing their version of the method.[7]

II. CONCLUSION

In this paper I want to conclude that a human body is a useless or a showpiece without a heart, similarly abstraction is important for programming of Matlab and OOPS. Abstraction works or programs internally and doesn't show any calculations or theory behind it. It simply shows the output once it is done. It hides the internal mechanism and working and generally shows the output regardless of working. Hence abstraction plays a vital role in Matalb and OOPs.

ACKNOWLEDGEMENT

REFERENCES

[1] E Balagurusamy, Object Oriented Programming with C++ (2016) Mc Graw hill publication

[2] W. Weisstein, Eric. "Extended Mean-Value Theorem". Retrieved 2018-10-08.

[3] James W. Lewis, Cortical Networks Related to Human Use of Tools"(2014)

[4] Sumit Arora Object Oriented Programming with C++ Dhanpat rai and C.O. (2016)

[5] Prof. Danny, Object Oriented Programming with C++ (2013) "Green Hills Publication

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

[7]   Sumita Arora textbook (Dhanpat Rai &co.) (2017)