iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** XII **Month of publication:** December 2019

DOI: http://doi.org/10.22214/ijraset.2019.12096

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue XII, Dec 2019- Available at www.ijraset.com

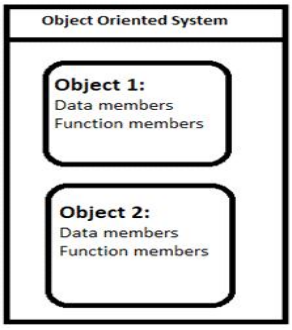
Pillars of Object Oriented System

Deepali Kasture [1], Dr. Rupesh C. Jaiswal [2]
[1,] *E&TC Department, Pune Institute of Computer Technology, Pune, Maharashtra, India.*

Abstract: In a software development process, designing the software plays an important role. Thus, the use of object oriented system has widely used for its advantages such as reusability, and providing security to the software. These features are implemented using various features of object oriented system. In this paper, we study the various pillars of object oriented system i.e. abstraction, encapsulation, inheritance and polymorphism.
Keywords: Abstraction, encapsulation, inheritance, polymorphism and resuabilty.

I. INTRODUCTION

In a Software Development Life Cycle(SDLC) cycle as mentioned in [1], the first phase is the software requirement analysis. The acquired analysis is then converted into design models. These design models follow various system to appropriately represent the engineering of the product. One of the system, which is widely used too, is Object Oriented System. This has been in much demand since 1980's. It is widely used for the software development process because oops basic concept such as an object is closely mapped with the real world entities. This enables the programmer to correlate the real world entities to the objects. An object is the instance of the class definition explained in an example in [3]. A class definition is a blue print of an object where we can define the attributes of the class and method to access or manipulate these attribute.
For example, a flower - rose. A rose is one of the flower object of the many flower object that exists. Hence, rose is the object of class flower. The object rose can have multiple data members, for instance colour, petals, etc. along with function members getColour(), getPetals(), etc.



[1]Fig. 1 Object Oriented System Block

The most important advantage of Object Oriented System mentioned in [9] is reusability of the objects in the entire software development process. [4] This helps in lowering the cost of the software and more efforts can be taken for design and analysis or the business logic of the software. Furthermore, it fastens the development process. [2] These features of object oriented system are based on the following pillars:

A. Abstraction

In simple words, abstraction is hiding of implementation which is not required by the user. Hence, the user only knows what the software does but hides its implementation from user.
For example, an ATM machine requires only the card number and pin from the user and it displays the details of the particular user. In this process it, does not show how the machine retrieves the user data from just card number and pin. Also, after performing operations such as withdrawal then producing statement of your account which displays the balance of the account. The user does not know the implementation performed behind the action. [8]
Secondly, when we take an example of a software development process itself. The customer provides the requirements of his software, the software engineer develops the software with the implementation of the requirements. He does not display its business logic to the customer.

B. Encapsulation

Encapsulation is the mechanism of protecting of data members by binding it with the function members of the class. This leads to data hiding of the object. [6]

Access to these data members is using getter and setter functions specified in class. Thus providing read and write access for the data members.

Access modifiers can be used to implement encapsulation. Data members can be made private for restricting the use of data members by other classes and making function members as public. Thus, allowing data members to be accessed by function members of the class. [5]

For example, the ATM machine asks for card number and pin, internally it verifies your details and displays your information, you do not have access to any of these internal process.

Similarly, any login page verifies your details and then gives you the access to your account, thus providing a protection to your information.

Also, when you are using a car you accelerate using accelerate () function to the desired speed, you cannot directly go to the desired speed, thus protecting the data variables. And to display the speed we use getSpeed() function.

C. Inheritance

As the name suggests inheritance help to inherit or acquire the properties of another class into the new class. The class which is supposed to be inherited is called the super class or the base class and the new class which inherits the properties of another class is called as the derived class or child class. The inherited class i.e. the derived maintains a “is-a” relationship with the base class. [7]

The key advantage of Inheritance is reusability. As we inherit the base class into derived class we can access the properties of base class, therefore reusing the implementation and reduces redundancy.

For example, a manager is an employee. He has Employee ID as all other employees only his work role is different from other employees. Thus, manager is inherited from the class employee. (Also, we human ourselves are inherit into men and women.) The class man and woman is inherited from class Human where it has hands, legs, eyes, heart, etc. Thus, man and woman class have additional attributes which distinguishes between them.

Furthermore, if we take a digital world example, in What’s App we have three ways of sending message text message, audio message, picture message. All these classes can be implemented by inheriting them from parent class message.

Generalization is same as superclass which has some common attributes which can be inherited. Specialization is similar to derived class in which we can inherit the superclass and can access its attributes.

D. Polymorphism

“Poly” means many and “morphism” means can change its form in literal meaning. It provides the facility to create more than one function with the same name but behave in different manner. The implementation is different for each method but has same name. These methods can be differentiated by the number of parameters, the type of parameters and the order in which the parameters are arranged. This gives a plus point to the programmer to write clean code. [10]

For example, we have a class vehicle, whose objects are car and aeroplane. Their function to move varies. But, we can have same function name but varying arguments passed to it which distinguishes between them. Here, car moves in forward direction and aeroplane moves in upward direction according to the implementation and instance created.

Also, a person behaves differently in different situation which gives a real time example of polymorphism. A person can be man, son, husband, employee, student, etc. and behave differently respectively.

II. CONCLUSION

Object Oriented system plays an important role in designing of a software. Object Oriented concepts gives advantages such as modularity, simplicity, reusability to build a robust software. Its pillars such as abstraction, encapsulation, inheritance and polymorphism. They enable the programmer to associate the objects with the real world entities. These pillars are the basic building block of an object oriented system and work in order respectively.

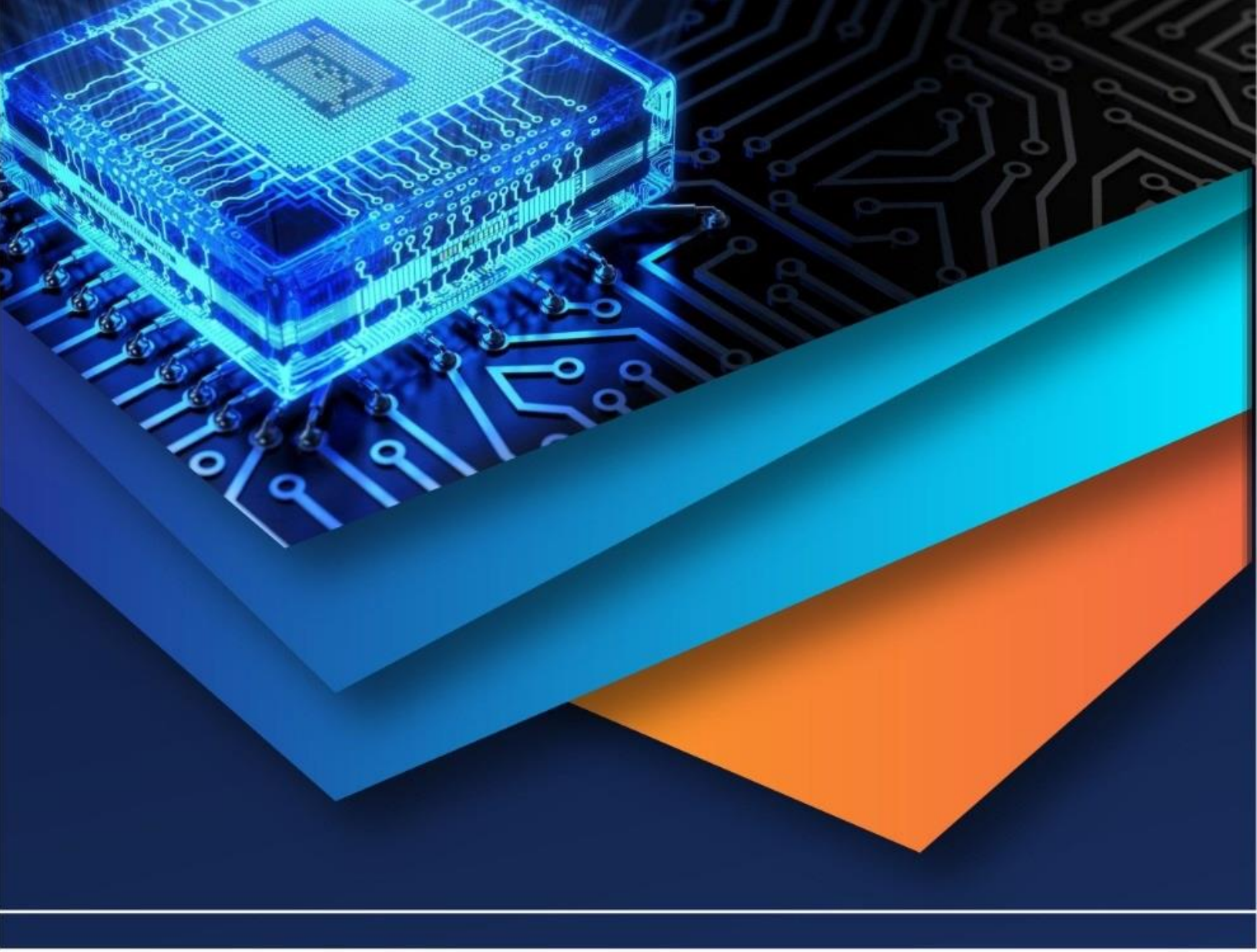
III. ACKNOWLEDGMENT

I wish to express my gratitude towards my guide, Dr. R. C. Jaiswal for being of great mentor and guiding me through the research. He gave this paper the insight and the expertise it needed for making it a presentable one. His guidance, constructive criticism and encouragement at various stages has been a great support.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue XII, Dec 2019- Available at www.ijraset.com

REFERENCES

[1] Chandramouli Subramanian, Saikat Dutt, Chandramouli Seetharaman., B. G. Geetha, “Software Engineering", Pearson, 2015.
[2] Eric Gamma.,” Design Patterns: Elements of Reusable Object-Oriented Software”, Pearson,1994.
[3] E. Balagurusamy,” Programming with Java”, Tata McGraw-Hill, 2007.
[4] Martin Fowler, “Patterns of Enterprise Application Architecture”, Addison Wesley Signature Series,2003.
[5] E. Balagurusamy,” Object Oriented Programming with C++”, Tata McGraw-Hill, 1994.
[6] Ray Klump, “Understanding Object-Oriented Programming Concept”, 2001 Power Engineering Society Summer Meeting. Conference Proceedings (Cat. No.01CH37262), IEEE, ISBN:0-7803-7173-9, 2001.
[7] Herbert Schildt, “C# 3.0: A Beginner’ Guide”, Osborne McGraw Hill, 2009.
[8] Cox, Novobilski, “Object-Oriented Programming: An Evolutionary Approach”,
[9] Gamma, Vlissides, Johnson, Helm, “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley Professional Computing Series, 1997.
[10] E. Balagurusamy, “OOPS Using C++ and Java”, Tata McGraw-Hill, 2007.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY