

Object-Oriented Programming and its Concepts

Ashwin Urdhwareshe

Department of Technology Management,
University of Bridgeport,
Bridgeport, CT 06604, USA

Copyright © 2016 ISSR Journals. This is an open access article distributed under the ***Creative Commons Attribution License***, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT: Object Oriented Programming (OOP) utilizes an alternate arrangement of programming language than old procedural programming dialects (C, Pascal, and so forth.). It is a methodology which is essentially centered on the way objects collaborate to convey and share the data. It changed the utilization of procedural oriented programming, where the attention was on the methodology of execution. The object oriented methodology brought another way, giving more significance to the items. There are four basic concepts which are to be emphasized. The first concept is that of an Object. It is the basic building block of any programming language. It represents the way you can describe anything in the programming world. The next is Class; a class in OOP is the principle body of any system. The classes frame the fundamental improvement unit of any system. Third factor is Inheritance, where we can create a new class known as derived class from the parent base class. This helps in reducing the time for coding and the derived class is error free. The last factor is Polymorphism, which means the capacity to ask for that the same operations be performed by an extensive variety of diverse sorts of things.

KEYWORDS: OOP, Object, Class, Inheritance, Polymorphism.

1 INTRODUCTION

Object-oriented programming (OOP) is a programming language that has been designed around objects rather than "procedures" and actual information rather than programming logic. Basically, a program is a set of executable methods which takes data as input, does the processing, and then gives the desired result. The actual effort or the foundation of programming was focused on the way of writing the logic rather than the definition of the data. It was Object-Oriented Programming that mainly laid emphasis that the objects in the programming paradigm are the one that should be altered according to our need and usage; and not the logic required behind its manipulation.

1.1 OBJECT-ORIENTED PROGRAMMING DEFINITION

Object oriented programming is an approach which is mainly focused on the way object interacts to communicate and share the information. It altered the usage of traditional procedural oriented programming, where the focus was on the procedure of execution. The object oriented approach brought a new path, giving more importance to the objects [1]. Object oriented programs are strictly type checked and flexible outlined, modifications on one object have no impact on the other related objects, which is one of the most important features in OOP [2]. Finding easy solutions to the programming problems is the first approach of any code. It also provides a new way to divide the problem and give proper programmable solutions [3]. The impact change that OOP has brought up is very unique and has various diverse aspects of programming [4]. This has been a recent technology which has replaced the usage of the traditional structural programming language [5]. The concepts used are for systemizing and implementing the simulation tools in the programming language [6].

1.2 IMPACT OF OBJECT-ORIENTED PROGRAMMING ON SOFTWARE DEVELOPMENT

Object Oriented Programming is a transformative advancement in programming building. The establishments for some object-oriented languages were built up by many years of software development knowledge which lead to the motivation of creation of elements, for example, closed methods, modules and hidden data [7]. OOP deals with the minimum of three software engineering goals; namely reusability, extensibility and flexibility.

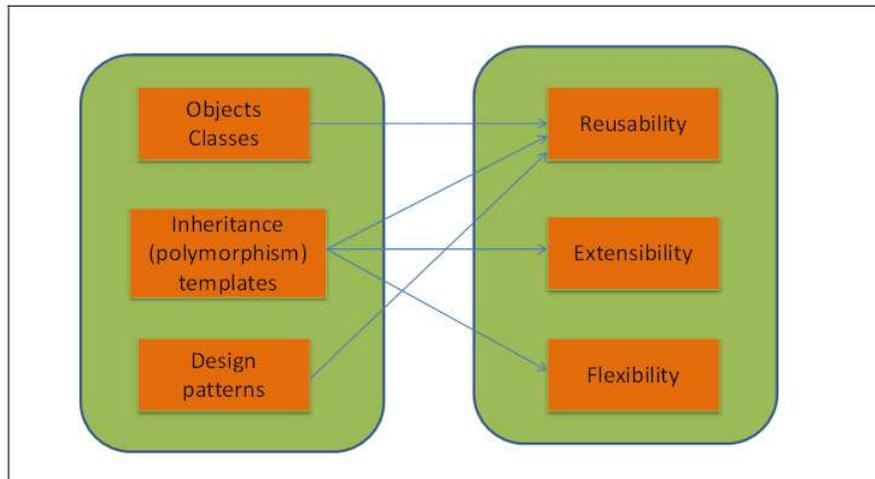


Fig. 1. Software Engineering Goals

Fig. 1 shows the goals of software engineering which are dependent on the OOP concepts.

Reusability is very beneficial in two ways for any developer. Firstly, the time for producing a code is reduced; once developed it becomes easy and efficient to reuse it many times. Secondly, the reliability of the code increases as we are using the previously existing code for our development. Extensibility can be related to the concept of inheritance in OOP. The attribute or behavior that is set for the base class will be extended to its derived class automatically and hence the same attribute can be used to refer the derived class variables. OOP is very flexible in terms of software development. Its flexibility can be understood by polymorphism, where we can add more variations and do modifications using the same function name but with different variables [7].

1.3 BENEFITS OF OBJECT-ORIENTED PROGRAMMING OVER PROCEDURAL ORIENTED PROGRAMMING

The programming language that uses various high end languages like COBOL, C etc is popularly known as Procedural Oriented Programming language or POP. In this language the main emphasis is on what procedure is followed for the program execution [8]. Also, which function is used is important. It includes number of steps or procedures that the system must follow and also it organizes these procedures into several groups known as functions. The functions created in the program share the global data and these functions transfer data from one to another. It focuses on following top-down approach of design.

Object-Oriented Programming is more realistic and is very popular among the developers. It tries to develop programs more likely modeled in a way the users imagine and work along [9]. There are many benefits of Object-Oriented Programming language over the traditional Procedural Oriented Programming. Firstly, with the advent of inheritance we can avoid the redundancy in our code and can increase the extensibility factor of our class, which was not possible while implementing Procedural Oriented Programming. Secondly, by implementing the concept of data hiding, the programmer can develop secure programmable code which cannot be altered in other parts of program. With the option of having many forms of objects to work together, there is no hindrance of one object on other. The understanding of the software becomes much easy while dealing with OOP. For communication and transfer of information, the objects interact with each other, which was difficult in POP [8].

2 RESEARCH METHOD

Motivation behind the research was to understand the impact of Object- Oriented Programming and its concepts on the programming world. The Research Method used a review centric approach which was done on the articles and journals from the IEEE database and Google Scholar. In- depth research was done on the impacts which object oriented programming has done in the programming paradigm. The University of Bridgeport Library was a source to find out various journals and articles on OOP and its various concepts. The practical implementation was carried out of the concepts of OOP in the programs which showed the importance of OOP.

3 OOP AND INDEPENDENT FACTORS MODEL

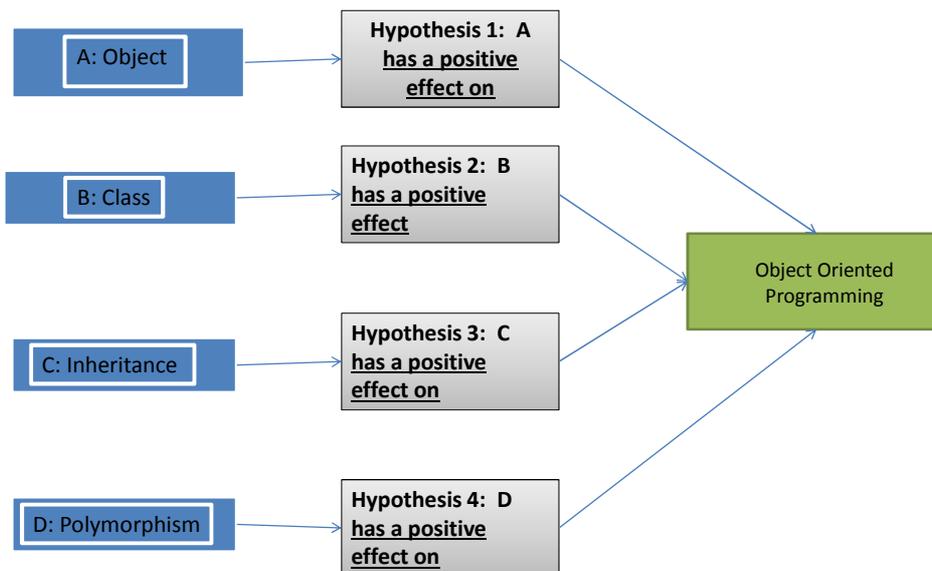


Fig. 2. OOP and Independent factor model

3.1 THE IMPORTANCE OF OOP

The most important aspect in programming is the formation of network. The emergence of Object-Oriented Programming increased drastically with the advent of the Internet for business sector. In spite of being used by almost everyone since 1960, it was not surprising that it would be so famous and adapted by majority after the invention of Internet [10]. Also, the concepts of OOP play a very vital role in programming language. OOP provides good design patterns which are basically a figurative model of the design of object; called class. It gives references for the user to create new objects and is very user friendly. It also gives us an option of sharing the code, which is called as inheritance where the old methods can be reused into new class. By giving prior error handling method, it gives user to recheck the code and compile the run-time errors with the use of exception handling [11].

3.2 FACTOR1: OBJECT

For any web application or a program to come into existence, the basic requirement is the way to code that program. The outline of any program has to be done prior developing it. For developing any program, there is necessity of an object in OOP [12]. They act as mediator between the program and the methods which are added with it. Also, privacy is the most important thing in any application or program development hence, objects contain private data as we do not want any hacker or unauthorized person to gain access to our data [13]. Only a few authorized users can access the data through certain operations. Object is anything that exists and has some value [14]. Suppose a person has his name, age and gender:

the person will then act as an object and the rest would be related attributes. The concept of private data and operations on that particular set of objects is very essential to understand [15]. For communication between different sets of objects, a particular rule is followed, which, is mandatory [16]. The contents within the objects are secure, and the structure and content in object is very necessary for users [17].

3.3 FACTOR2: CLASS

A class in OOP is the main body of any program. All the methods and its definitions are presented inside the class. Classes form the basic development unit of any program [18]. Every class contains objects inside it which are executable. It is a collection of distinct objects with which information can be shared [19]. Class features the data definition, i.e. the way data has been defined with the attributes and respective methods. Classes use attributes and functions to operate the given data [20]. The classes contain certain functions for its operation which are used for the user-defined functionality [21]. Hierarchical definition of class gives us a clear idea of which class has the priority to execute and function and the class follows a proper order of execution [22]. Class is the key element in software monitoring and implementing the concepts of OOP [23].

3.4 FACTOR3: INHERITANCE

Inheritance gives any user the provision to utilize the same code in his program just by changing the definition of its variables [24]. The process of deriving a whole new class from the base class has various advantages: the execution time would get reduced, the possibilities of errors will be less as we are using the same attributes which have been compiled by the system and, they are error free [25]. With the help of the existing class, we can create a new class using same functions [26]. The existing class will be used as a reference class for the derived class. To organize our grammar or language in a systematic way in the program, we must use inheritance. The reusability of the same grammar in the new sub-class would reduce execution and compilation time [27]. Reusing the attributes, methods make the task of programming easy and the reusability of the code from our existing class to our new derived class is very feasible for the users [28]. By making the use of inheritance, we save the execution time and the programming complexity. Inheritance has "is-a" relationship with the derived classes, which gives us a clear picture of the classes will look and their proper definition is represented while following this relationship technique [29].

3.5 FACTOR4: POLYMORPHISM

The fast execution of any process or program is the main motivation behind the programmer's logic. There are many aspects to think upon when coding is done for any program, for example, the programs logic, the scope, the semantics of the program. One does not want to put his time and efforts in giving each variable or an object a specific individual meaning and definition; hence the concept of polymorphism was introduced. Polymorphism gives the way to represent variables as a group or a common way to represent the variables [30]. The code usage gives the programmer to utilize the time and get fast results [31]. We could define only one singular code for many different types of variables to perform the action of execution quickly. By allowing common functions to work for various objects we avoid the complexity and bind objects together [32]. Binding of objects makes the task of giving definition easy, which is very necessary if we have to write long codes for programs [33]. By giving access to users on defining their objects limits to certain classes, the user gets the leverage to use the same object at different positions with different functions [34]. It joins the derived classes together which gives us a clear picture of what polymorphism and its importance in OOP [35].

4 CONCLUSION

The research now has proven the importance and the impact that Object-Oriented Programming has in today's programming environment. Also it has shown a very drastic change in the nature of programming by the advent of Object-Oriented Approach. The elements of OOP play a very important role in the development of the language and give broad options to the users while developing in the programming environment. Object, which is a crucial part of OOP, is the basis for representing of any information. Class; a blueprint or prototype is essential for the structure of the program. Every function and method resides inside the class definition. Inheritance, which gave the concept of reusability of the code, makes the programming easy and time efficient. Hence concluding, OOP plays a vital role and its usage is being increasing in every programming module.

ACKNOWLEDGMENT

I would like to thank Professor Christian Bach for his valuable guidance and support in this research.

REFERENCES

- [1] Laimek, R. and W. Pawgasame. Internal ballistics simulation based on object oriented programming. in Defence Technology (ACDT), 2015 Asian Conference on. 2015.
- [2] Neyer, A., F.F. Wu, and K. Imhof, Object-oriented programming for flexible software: example of a load flow. Power Systems, IEEE Transactions on, 1990. 5(3): p. 689-696.
- [3] Longshu, L. and X. Yi. The teaching research on a case of object-oriented programming. in Computer Science and Education (ICCSE), 2010 5th International Conference on. 2010.
- [4] Dewar, R. and F. Gasperoni. Safety and OOP. in System Safety, 2006. The 1st Institution of Engineering and Technology International Conference on. 2006.
- [5] Lei, Z. Study on comparison of AOP and OOP. in Computer Science and Service System (CSSS), 2011 International Conference on. 2011.
- [6] de Lima, C.H.M., et al. A software development framework based on C++ OOP language for link-level simulation tools. in Telecommunications Symposium, 2006 International. 2006.
- [7] kafura. Object-Oriented Programming and Software Engineering. 1996
Available from: <http://people.cs.vt.edu/kafura/cs2704/oop.swe.html>.
- [8] Procedure Oriented Programming(POP) vs Object Oriented Programming(OOP). 2011.
- [9] Pillay, A., Object Oriented Programming using Java. 2007.
- [10] Weisfeld, M. The Importance of Object-Oriented Programming in the Era of Mobile Development. 2013; Available from: <http://www.informit.com/articles/article.aspx?p=2036576>.
- [11] Course, B.J.E., Object-oriented concepts, 2010.
- [12] Reyna, A.M., et al. Object-oriented programming as an alternative to industrial control. in Electrical Engineering, Computing Science and Automatic Control (CCE), 2012 9th International Conference on. 2012.
- [13] D'Andrea, R.J. and R.G. Gowda. Object-oriented programming: concepts and languages. in Aerospace and Electronics Conference, 1990. NAECON 1990., Proceedings of the IEEE 1990 National. 1990.
- [14] Zimmerman, M.L. and P. Mallasch. Using object-oriented programming in computational electromagnetic codes. in Antennas and Propagation Society International Symposium, 1995. AP-S. Digest. 1995.
- [15] Marchesi, M., G. Orlandi, and F. Piazza. Network synthesis using object-oriented programming. in Electrotechnical Conference, 1989. Proceedings. 'Integrating Research, Industry and Education in Energy and Communication Engineering', MELECON '89., Mediterranean. 1989.
- [16] Khan, E.H., M. Al-A'ali, and M.R. Girgis, Object-oriented programming for structured procedural programmers. Computer, 1995. 28(10): p. 48-57.
- [17] Breen, D.E., P.H. Getto, and A.A. Apodaca. An object-oriented programming methodology for a conventional programming environment. in Software Engineering, 1988 Software Engineering 88., Second IEE/BCS Conference.: 1988.
- [18] Heung Seok, C. and K. Yong Rae. A cohesion measure for classes in object-oriented systems. in Software Metrics Symposium, 1998. Metrics 1998. Proceedings. Fifth International. 1998.
- [19] Omar, N. and N.A. Razik. Determining the basic elements of object oriented programming using natural language processing. in Information Technology, 2008. ITSIM 2008. International Symposium on. 2008.
- [20] Seban, R.R. An overview of object-oriented design and C++. in Aerospace Applications Conference, 1994. Proceedings., 1994 IEEE. 1994.
- [21] Maney, H.D. An object-oriented synthetic aperture radar processor implemented using C++ classes. in Aerospace and Electronics Conference, 1994. NAECON 1994., Proceedings of the IEEE 1994 National. 1994.
- [22] McIntyre, S.C. and L.F. Higgins, Object-oriented Systems Analysis and Design: Methodology and Application. Journal of Management Information Systems, 1988. 5(1): p. 25-35.
- [23] Butler, S., et al. Mining java class naming conventions. in Software Maintenance (ICSM), 2011 27th IEEE International Conference on. 2011.
- [24] Vedpal, N. Chauhan, and H. Kumar. A hierarchical test case prioritization technique for object oriented software. in Contemporary Computing and Informatics (IC3I), 2014 International Conference on. 2014.
- [25] Makkar, G., J.K. Chhabra, and R.K. Challa. Object oriented inheritance metric-reusability perspective. in Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on. 2012.

- [26] Pole, T. A case study in the use of multiple inheritance in designing a library of reusable software components. in Southeastcon '91., IEEE Proceedings of. 1991.
- [27] Mernik, M., et al. The template and multiple inheritance approach into attribute grammars. in Computer Languages, 1998. Proceedings. 1998 International Conference on. 1998.
- [28] Rathore, N.P.S. and R. Gupta. A novel coupling metrics measure difference between inheritance and interface to find better OOP paradigm using C#. in Information and Communication Technologies (WICT), 2011 World Congress on. 2011.
- [29] Krishnapriya, V. and K. Ramar. Exploring the Difference Between Object Oriented Class Inheritance and Interfaces Using Coupling Measures. in Advances in Computer Engineering (ACE), 2010 International Conference on. 2010.
- [30] Milojkovic, N., et al. Polymorphism in the Spotlight: Studying Its Prevalence in Java and Smalltalk. in Program Comprehension (ICPC), 2015 IEEE 23rd International Conference on. 2015.
- [31] Tan, Y., C. Yau, and A. Fong. Architectural Support on Object-Oriented Programming in a JAVA Processor. in Application-specific Systems, Architectures and Processors, 2006. ASAP '06. International Conference on. 2006.
- [32] Bor-Yuan, T., S. Stobart, and N. Parrington. Using extended general statecharts to generate test cases for overloaded/polymorphic object-oriented programs: a case study. in Technology of Object-Oriented Languages, 1997. TOOLS 24. Proceedings. 1997.
- [33] Rountev, A. and A. Milanova. Fragment class analysis for testing of polymorphism in Java software. in Software Engineering, 2003. Proceedings. 25th International Conference on. 2003.
- [34] Mannino, M.V., I.J. Choi, and D.S. Batory. An overview of the Object-Oriented Functional Data Language. in Data Engineering, 1989. Proceedings. Fifth International Conference on. 1989.
- [35] Hang, Z., H. Zhiqiu, and Z. Yi. Polymorphism Sequence Diagrams Test Data Automatic Generation Based on OCL. in Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for. 2008.