International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307

OPEN ACCESS

Available Online at : www.ijsrcseit.com

doi : https://doi.org/10.32628/CSEIT243647

Survey on Concept of Object-Oriented Programming

Mrs. Mansi Dhirajsinh Parmar[1], Ms. Sarthavi Parmar[2]

Lecturer, Department of Computer Engineering, Parul Institute of Engineering & Technology-Diploma Studies, Vadodara, Gujarat, India

[2]Lecturer, Department of Computer Engineering, Parul Institute of Engineering & Technology-Diploma Studies, Vadodara, Gujarat, India

ARTICLEINFO

Article History:

Accepted: 15 March 2024

Published: 05 April 2024

Publication Issue

Volume 10, Issue 2

March-April-2024

Page Number

427-431

ABSTRACT

These days, object-oriented programming is regarded as an essential programming concept.The moment Simula brought it into life. The use of object-oriented programming (OOP) has expanded in the software real world due to the future growth of the software business and the advancement of software engineering.

The following review examines different oop concepts that are essential to object-orientation, in great detail. Many widely used object-oriented programming languages implement various parts of inheritance and polymorphism. We come to the conclusion that much more work needs to be done to find a middle ground so that these can accomplish OOPs features.

Keywords : Object Oriented Programming, Class, Object, Encapsulation, Inheritance, Polymorphism

I. INTRODUCTION

Programmers promptly found that organizing an amount of data with the functions that processed it into a logical group makes a program easier to read and execute. We refer to this type of grouping of functions and data as class and object. And object-oriented programming is the process of writing programs using classes.

Bjarne Stroustrup began developing "C with Classes," a new language, in 1980. This new language was a feature class extension for C. Following certain modifications and enhancements, this language was given the name C++. Introduced in 1983, it was given the suffix C++ and came with all of its features. The OOPs feature of C++ was motivated by the simula67 a computer simulation language.

Stroustrup enhances C with OOP features without appreciably altering the C component. Because C++ is a superset of the C language, any legitimate C program can also be written in C++.

Real-world concepts like inheritance, hiding, and polymorphism are intended to be implemented in

programming through object-oriented programming. OOP's primary goal is to link together the data and the functions that manipulate it such that only that function and no other portion of the code may access the data.

Programming in terms of classes, objects, inheritance, polymorphism, abstraction, encapsulation, and other related concepts are referred to as object-oriented programming.

IMPORTANCE

With C++, object-oriented programming is essential to developing reliable and accessible software systems because it places a strong emphasis on abstraction, polymorphism, inheritance, modularity, and encapsulation. Open Object Programming facilitates code reusability, scalability, and simplicity of maintenance by decomposing programs into manageable objects with clean interfaces. Encapsulation guarantees data confidentiality and minimizes unwanted side effects, whilst polymorphism and inheritance provide flexibility and order in the code. Furthermore, abstraction makes complex systems simpler by concealing implementation details, enhancing the readability and dependability of the code. All things considered, OOP with C++ complies to industry standards and best practices, making it a potent paradigm for creating scalable and effective software solutions in a variety of fields.

OOP CONCEPTS

Programming languages that employ objects are referred to as object-oriented programming, or OOPs, as the name would imply. Real-world concepts like inheritance, hiding, polymorphism, etc., are intended to be implemented in programming through object-oriented programming. OOP's primary goal is to bind together the data and the functions that work with it, preventing access to the data by any other portion of the code except that function.

A. CLASS: A class is a user-defined data type containing member functions that can alter the data members as well as data members themselves. It is an assortment of things that are similar in kind. An object's generic definition is called a class. It is an object's blue print.
B. OBJECT: Objects are actually run-time units that exist in the real world; they are instances of classes; they contain data and code to manipulate that data; they can interact with one another without needing to comprehend specifics about the code or statistics; we can access members using objects that resemble the structure in C.
C. ENCAPSULATION: The method that connects the functions and data is called encapsulation. This safeguard prevents outside influence and misuse of both data and functions. Encapsulated code has the benefit that the user is aware of how to access it and is not concerned with implementation details.
D. ABSTRACTION: One of the most crucial and significant aspects of C++ object-oriented programming is data abstraction. Abstraction is the process of revealing only the pertinent details and keeping the rest hidden. Data abstraction is the process of revealing to the outside world only the most important characteristics about a set of data while keeping the implementation and background information hidden.
E. INHERITANCE: It is referred to as inheritance when a new class is derived from an existing class.

The idea of reusability, or the ability to construct a new class that incorporates all of the capabilities of an existing class while also potentially adding some new features, is what inheritance is all about. In this instance, the base class is the existing class from which we are driving the new class, and the drive class is the newly created class. In C++, this idea is referred to as reusability. The current class is reusable because of inheritance.

F. POLYMORPHISM: The concept of polymorphism denotes to the capacity to take several forms. One of the key components of object-oriented programming is polymorphism. The Greek terms "poly" and "morphism" combine to form the word polymorphism.

Since "morphism" means forms and "poly" means many, polymorphism is a description of multiple forms.

G. MESSEAGE PASSING: Message passing refers to the exchange of information between objects through messages. Just as individuals exchange information, so does the sending and receiving of information by the object. The following steps must be taken in order for message passing to occur:

- Classes that describe objects and their behaviors must be created.
- Next Using the class definitions to create the objects
- Calling and establishing a connection for object communication

II. LITERATURE REVIEW

The literature study of the text mentioned above emphasizes the importance of Object-Oriented Programming (OOP) and its underlying ideas in contemporary software engineering. From its beginnings in the 1960s with programming languages like Simula and Smalltalk to its widespread adoption in languages like C++, Java, and Python, it charts the historical evolution of OOP. The fundamental ideas of object-oriented programming (OOP), such as classes, objects, encapsulation, inheritance, polymorphism, and abstraction, are covered in the review with a focus on how they help write code that is modular, reusable, and maintainable.

The assessment also looks at how OOP languages and frameworks have changed over time, pointing out the appearance of contemporary languages like Kotlin, Swift, and TypeScript that have more features and are more productive. It examines the many uses of OOP in a variety of fields, including software design and analysis and real-world applications.The assessment also acknowledges OOP's advantages in fostering code scalability and managing complexity, while simultaneously addressing its drawbacks and complaints, including complexity, tight coupling, and performance overhead. It covers how OOP is incorporated into agile approaches and DevOps procedures in contemporary software development, as well as how it affects cutting-edge fields like block chain, machine learning, and the Internet of Things.

A discussion of instructional strategies and materials for teaching OOP principles is also included, with a focus on the value of theoretical ideas and hands-on training in developing competent workers. Lastly, the review emphasizes OOP's continued significance in the constantly changing field of software development while highlighting trends and future directions for the language, such as continuous innovation and adaption to new technologies and paradigms.

CHALLENGES

Complexity: Especially for novices, OOP ideas like inheritance and polymorphism might be confusing.
Memory Management: Handling memory by hand can be difficult and result in problems like memory leaks and hanging pointers.
Performance Overhead: Encapsulation and abstraction can occasionally result in performance overhead.

Numerous Inheritance: Numerous inheritances can result in complicated class hierarchies and problems like the diamond dilemma.

Information loss can happen when derived class objects are assigned to base class objects, a phenomenon known as object slicing.

Resource management: To avoid leaks, cautious attention must be paid to the management of resources such as file handles and network connections.

OOP codebases can be difficult to debug and manage, particularly when there are intricate classhierarchies. Learning Curve: Making the switch to OOP calls for a new way of thinking and grasping concepts.

Architectural Complexity: Careful preparation is necessary to prevent tight coupling and code duplication while designing efficient class structures.

Mobility and Consistency: It might be difficult to guarantee portability and compatibility across many compilers and systems.

III.CONCLUSION

The ideas, importance, and applications of object-oriented programming (OOP) are thoroughly examined in this review article. The modular, reusable, and maintainable OOP design methodology is ideal for creating intricate systems. It makes use of polymorphism to enable objects to display alternative behaviours, inheritance to encourage code reuse, abstraction to reduce complexity, and classes to encapsulate data and functions. Modern programming languages that offer OOP features and are widely used in software applications, such as C++, make OOP especially significant. OOP concepts will continue to be crucial for developing creative and long-lasting software solutions as technology develops.

IV.ACKNOWLEDGEMENT

We are appreciative of PIET-DS. We thank the faculty especially my HOD, Prof. Hetal Bhaidasna Ma'am, for their assistance. I am grateful for their assistance and guidance.

V. REFERENCES

[1] Mr. RUSHIKESH S. RAUT "Research Paper on Object-Oriented Programming (OOP)" International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 07 Issue: 10 | Oct 2020 www.irjet.net p-ISSN: 2395-0072

[2] Rajesh Babu Nagineni1 1Assistant Professor, Dept. Of Computer Science and Engineering, KLEF, Andhra Pradesh, India "A Research on Object Oriented Programming and Its Concepts" Volume 10, No.2, March - April 2021 International Journal of Advanced Trends in Computer Science and Engineering Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse401022021.pdf https://doi.org/10.30534/ijatcse/2021/401022021

[3] Bjarne Stroustrup AT&T Bell Laboratoriea Murray Hill, New Jersey 07974 What is "Object-Oriented Programming"? (1991 revised version)

[4] M. Surya S. Padmavathi Computer Science and engineering, Sri Krishna College of Technology, Coimbatore, Tamilnadu, India 2Assistant Professor, Computer Science and engineering, Sri Krishna College of Technology, Coimbatore, Tamilnadu, India "A Survey of Object-Oriented Programming Languages" International Journal of Scientific Research in Computer Science, Engineering and Information Technology © 2019 IJSRCSEIT | Volume 5 | Issue 2 | ISSN : 2456-3307 DOI : 10.32628/CSEIT195248

[5] Prof. Chetashri S.Bhusari, Prof. Shonal AVaz*, Prof. Supriya Angne "CONCEPTS OF OBJECT ORIENTED PROGRAMMING"

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH Volume-8 | Issue-6 | June-2019 | PRINT ISSN No. 2277 – 8179

[6] Miss. Ghodke Radha B*1, Miss. Gaikwad Gaytri D*2 *1,2Lecturer, Dept. Of Computer Engineering Technology MIT Polytechnic Yeola Maharashtra, India. DOI : https://www.doi.org/10.56726/IRJMETS31010 "BASIC CONCEPT OF OBJECT-ORIENTED PROGRAMMING (OOP)" International Research Journal of Modernization in Engineering Technology and Science (Peer-Reviewed, Open Access, Fully Refereed International Journal)Volume:05/Issue:09/September-2023 Impact Factor- 7.868

[7] Tim Rentsch Computer Science Department University of Southern California Los Angeles, California 90089-0782 "Object Oriented Programming"

[8] VISHAL VAMANMEHTRE1, AKSHAT BHALLA21Assistant 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 "Analysis of Object Oriented Programming" © DEC 2019 | IRE Journals | Volume 3 Issue 6 | ISSN: 2456-8880

[9] Research Paper on "Object Oriented Software Engineering" 1Iqbaldeep Kaur, 2Navneet Kaur, 3Amandeep Ummat, 4Jaspreet Kaur, 5Navjot Kaur. of CSE, Chandigarh Engineering College, Landran, Punjab, India IJCST Vol. 7, Issue 4, Oct - Dec 2016

[10] Soly Mathew Biju University of Wollongong, soly@uow.edu.au "Difficulties in understanding object oriented programming concepts" University of Wollongong January 2013 Research Online University of Wollongong in Dubai - Papers University of Wollongong in Dubai