GALAXY INTERNATIONAL INTERDISCIPLINARY RESEARCH JOURNAL (GIIRJ)
ISSN (E): 2347-6915
Vol. 11, Issue 11, November (2023)

C AND C++ PROGRAMMING LANGUAGES CAPABILITIES AND DIFFERENCES

Abrorjon Kholmatov,
Teacher
xolmatovabrorjon@gmail.com,

Abdurahimova Mubiyna,
student
Fergana branch of the Tashkent university of information
technologies named after Muhammad al-Khorazmi
abdurahimovamubiynaxon@gmail.com,

ABSTRACT

C and C++ are two closely related programming languages, each with its unique capabilities and differences. Both languages are renowned for their efficiency and versatility, making them popular choices for various applications, from system programming to application development. This abstract provides a concise overview of the capabilities and key distinctions between C and C++.

C Programming Language: C is a procedural programming language known for its simplicity and low-level capabilities. It follows a top-down approach, focusing on functions and procedures. C is well-suited for system programming and embedded systems due to its direct hardware access and manual memory management. It offers portability and can be adapted to different platforms. However, it lacks built-in support for object-oriented programming (OOP) and relies on a relatively small standard library.

C++ Programming Language: C++ is a multi-paradigm programming language that builds upon C's foundation. It supports both procedural and object-oriented programming. C++ introduces classes and objects, enabling encapsulation, inheritance, and polymorphism. It features a rich standard library, offering a wide range of pre-built functions and classes. C++ also provides features such as operator overloading, templates, automatic memory management, and exception handling. It maintains backward compatibility with C, allowing the integration of C code.

Keywords: C programming language, C++ programming language, Procedural programming, Object-oriented programming (OOP), Low-level programming, Simplicity, Rich standard library, Classes and objects, Operator overloading, Templates, Automatic memory management, Exception handling, Backward compatibility, System programming, Portability, Manual memory management, Code reusability, Polymorphism, Encapsulation, Efficiency

INTRODUCTION

C and C++ are two powerful and influential programming languages that have played a pivotal role in the world of software development for decades. Both languages are renowned for their efficiency, performance, and versatility. While they share a common ancestry, they also exhibit significant differences in terms of their capabilities and programming paradigms. This introduction sets the stage for a deeper exploration of the unique features and distinctions that define C and C++.

C Programming Language: C, born in the early 1970s, is often described as a procedural programming language. It is characterized by its simplicity, offering a minimalistic set of features and a straightforward, top-down approach to problem-solving. C's focus on functions and procedures makes it well-suited for tasks where low-level memory control and direct hardware interaction are critical. It is a language of choice for system programming, embedded systems, and applications that require precise control over resources.

C++ Programming Language: C++, emerging in the 1980s, builds upon the foundation laid by C. It is a multi-paradigm language, blending procedural and object-oriented programming (OOP) features. The introduction of classes and objects in C++ enables encapsulation, inheritance, and polymorphism, offering a more structured and modular approach to software design. C++ distinguishes itself with a rich standard library that provides an extensive collection of pre-built functions and classes for various tasks.

Key Differences: The differences between C and C++ are not limited to their programming paradigms. C++ introduces a range of advanced features, including operator overloading, templates for generic programming, automatic memory management, and exception handling. It also maintains backward compatibility with C, allowing developers to seamlessly integrate existing C code into C++ programs.

This discussion will delve deeper into the capabilities and differences of C and C++, providing insights into when to choose one over the other, depending on the specific requirements of a project. Whether you are interested in low-level system programming or creating complex, object-oriented applications, understanding the strengths and weaknesses of C and C++ is essential for making informed programming decisions.

INTRODUCTION TO THE LITERATURE REVIEW

The C and C++ programming languages have long been a subject of fascination and exploration for software developers and researchers. This literature review aims to provide an in-depth analysis of the capabilities and differences between C and C++ programming languages. Understanding these two languages is crucial for making informed decisions in software development, as they are widely used for various applications.

Historical Significance: Both C and C++ have rich historical significance in the field of computer science and software engineering. C, created by Dennis Ritchie at Bell Labs in the early 1970s, became the foundation for many subsequent programming languages and played a pivotal role in the development of Unix and countless other operating systems and applications. C++, developed by Bjarne Stroustrup in the 1980s, was inspired by C but introduced powerful features from the world of object-oriented programming.

Procedural vs. Object-Oriented Paradigms: The core distinction between C and C++ lies in their programming paradigms. C is fundamentally a procedural language, emphasizing top-down, structured programming with functions as its primary building blocks. In contrast, C++ embraces both procedural and object-oriented paradigms, offering developers the ability to create modular, reusable, and efficient code through classes, objects, inheritance, and polymorphism.

Capabilities of C: C is celebrated for its simplicity and efficiency. It provides a minimalist set of features and is often chosen for tasks requiring low-level memory control, hardware interaction,

and portability. Its manual memory management and straightforward design make it an excellent choice for system programming and embedded systems.

Capabilities of C++: C++ inherits C's efficiency while expanding its capabilities. With a rich standard library, C++ offers a wide range of pre-built functions and classes, simplifying common programming tasks. Features like operator overloading, templates for generic programming, automatic memory management, and exception handling elevate C++ to an ideal choice for complex, object-oriented applications.

The Influence of Backward Compatibility: One intriguing aspect of C++ is its commitment to backward compatibility with C. This means that existing C code can be seamlessly integrated into C++ programs, allowing developers to leverage legacy code while taking advantage of C++'s advanced features.

Scope of the Literature Review: This literature review will explore various aspects of C and C++, including their language features, use cases, advantages, and limitations. By synthesizing insights from existing research and real-world applications, we aim to provide a comprehensive understanding of these languages, assisting programmers, researchers, and decision-makers in selecting the most appropriate language for their projects.

CONCLUSION

The C and C++ programming languages represent two pillars of software development, each with its own distinct capabilities and differences. This review has provided a comprehensive analysis of these languages, shedding light on their historical significance, programming paradigms, and various attributes that make them invaluable in the world of software engineering.

Historical Significance: C's invention by Dennis Ritchie in the early 1970s and C++'s development by Bjarne Stroustrup in the 1980s have left an indelible mark on the history of computer science. C's role in the birth of Unix and its widespread adoption in system programming have solidified its place in computing history. C++'s incorporation of object-oriented programming (OOP) features has enabled developers to create modular, reusable, and efficient software systems.

Procedural vs. Object-Oriented Paradigms: The primary distinction between C and C++ is their programming paradigms. C is rooted in procedural programming, emphasizing structured, function-driven code. C++, on the other hand, seamlessly combines both procedural and object-oriented paradigms, empowering developers with the tools to create software that is not only efficient but also modular and maintainable through the use of classes, objects, inheritance, and polymorphism.

Capabilities of C: C's strengths lie in its simplicity and efficiency. With a minimalistic design, it is favored for low-level system programming, where direct memory management and hardware interaction are crucial. Its straightforward nature and manual memory management provide precise control over resources.

Capabilities of C++: C++ takes the efficiency of C and expands upon it. With a rich standard library and advanced features such as operator overloading, templates for generic programming, automatic memory management, and exception handling, it is ideally suited for complex, object-oriented applications. C++ simplifies common programming tasks, offering pre-built functions and classes that improve productivity.

Backward Compatibility: An intriguing aspect of C++ is its commitment to backward compatibility with C. This allows developers to seamlessly integrate existing C code into C++ programs, preserving legacy code while benefiting from C++'s advanced features.

Choosing the Right Language: The choice between C and C++ ultimately depends on the specific requirements of a project. C is the natural choice for system programming, embedded systems, and tasks demanding low-level control. C++ excels in applications where modularity, code reusability, and a rich standard library are advantageous. Both languages have their rightful place in software development and should be chosen based on the problem at hand.

In conclusion, C and C++ represent enduring languages that continue to shape the software development landscape. Their unique strengths, from low-level control to object-oriented features, make them indispensable tools for programmers and system developers alike. Understanding their capabilities and differences is essential for making informed decisions and creating efficient, reliable, and maintainable software systems.

REFERENCES

1. Abrorjon Kholmatov. (2023). WIDELY USED LIBRARIES IN THE JAVASCRIPT PROGRAMMING LANGUAGE AND THEIR CAPABILITIES. Intent Research Scientific Journal, 2(10), 18–25. Retrieved from https://intentresearch.org/index.php/irsj/article/view/220
2. Kholmatov, Abrorjon. "Pedagogical Technologies in Teaching Students About Web Programming." Journal of Pedagogical Inventions and Practices 25 (2023): 40-44.
3. Abdurasulova D. B. Q., Yakubov M. S. YUK OQIMLARINI BOSHQARISHNI TASHKIL ETISHNING O'ZIGA XOS XUSUSIYATLARI //Academic research in educational sciences. – 2022. – Т. 3. – №. 3. – С. 734-737.
4. Soliev B. N., Abdurasulova D., Yakubov M. S. USING THE DJANGO FRAMEWORK FOR E-COMMERCE PROCESSES //Journal of Integrated Education and Research. – 2022. – Т. 1. – №. 6. – С. 229-233
5. Sodikova M. MOBIL QURILMALAR ISHLAB CHIQISH FANINI O ‘QITISHDA SUN’IY INTELLEKTNING ROLI //Research and implementation. – 2023. – Т. 1. – №. 2. – С. 79-83.
6. Xolmatov Abrorjon Alisher o‘g‘li, Muminjonovich Hoshimov Bahodirjon, and Uzokov Barhayot Muhammadiyevich. "Teaching Children to Programming on the Example of the Scratch Program." Eurasian Scientific Herald 9 (2022): 131-134.
7. Xadjayev S. АСИНХРОННАЯ БИБЛИОТЕКА PYTHON ASYNCIO: ПРЕИМУЩЕСТВА И ПРИМЕРЫ ПРИМЕНЕНИЯ //Потомки Аль-Фаргани. – 2023. – Т. 1. – №. 2. – С. 45-48.

8. Ravshanbek Z., Saidakbar X. Staff Incentives Based on Kpi Principles //Periodica Journal of Modern Philosophy, Social Sciences and Humanities. – 2023. – Т. 17. – С. 101-105.
9. Muminjonovich K. A. SUN'IY INELLEKTNI RIVOJLANTIRISHDA DASTURLASH TILLARINING RO'LI //Journal of new century innovations. – 2023. – Т. 12. – №. 4. – С. 159-161.
10. Kayumov A. CREATING AN EXPERT SYSTEM-BASED PROGRAM TO EVALUATE TEXTILE MACHINE EFFECTIVENESS //Потомки Аль-Фаргани. – 2023. – Т. 1. – №. 2. – С. 49-52.
11. Sodikova M. EFFECTIVE METHODS OF TEACHING HISTORY //НАУКА И ТЕХНИКА. МИРОВЫЕ ИССЛЕДОВАНИЯ. – 2020. – С. 29-31.
12. Sadikova M. OPTIMIZATION OF THE BUSINESS PROCESS AS ONE OF THE MAIN TASKS IN MODERN MANAGEMENT //Теория и практика современной науки. – 2022. – №. 9 (87). – С. 3-7.
13. Sadikova Munira Alisherovna. (2023). DISADVANTAGES OF TEACHING PROGRAMMING IN DISTANCE EDUCATION. Intent Research Scientific Journal, 2(10), 26–33.
14. Alisherovna S. M. WAYS TO WRITE CODE ON ANDROID DEVICES //American Journal of Technology and Applied Sciences. – 2023. – Т. 17. – С. 39-42.
15. Хаджаев С. И. ЦИФРОВЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ: ПРЕИМУЩЕСТВА И ПЕРСПЕКТИВЫ //Молодые ученые. – 2023. – Т. 1. – №. 4. – С. 10-11.
16. Nabijonovich S. B. et al. UNVEILING THE FUTURE OF DATA EXTRACTION USING PYTHON AND AI FOR VIDEO-BASED INFORMATION RECOGNITION //American Journal of Technology and Applied Sciences. – 2023. – Т. 17. – С. 26-32.
17. Abdurasulova, D. B. kizi, & Irmatova , D. B. (2023). USE OF DIFFERENT ALGORITHMS AND APPLICATION OF SOFTWARE PRODUCT CREATION SEQUENCES IN ORGANIZING COMPLEX STRUCTURED PROJECTS. Educational Research in Universal Sciences, 2(11), 170–173. Retrieved from http://erus.uz/index.php/er/article/view/3947
18. Abdurasulova D. THE MAIN DIRECTIONS OF MODERN PRAGMALINGUISTICS: IDEAS AND PERSPECTIVES //InterConf. – 2021.
19. Asrayev M. 0-TARTIBLI BIR JINSLI FUNKSIONALLAR KO 'RINISHIDAGI SODDA MEZONLAR UCHUN 1 INFORMATIV BELGILAR MAJMUASINI ANIQLASH USULLARI //Потомки Аль-Фаргани. – 2023. – Т. 1. – №. 2. – С. 9-1
20. Asrayev M., Dadaxonov M. BERILGAN TASVIR SIFATINI BAHOLASH //Потомки Аль-Фаргани. – 2023. – Т. 1. – №. 2. – С. 13-16.
21. Fazilov S. K. et al. State of the art of writer identification //Compusoft. – 2019. – Т. 8. – №. 12. – С. 3514-3524.
22. Asrayev M. MEZON KO'RINISHIGA BOG'LIQ BO 'LMAGAN INFORMATIV BELGILAR FAZOSINI SHAKLLANTIRISH USULLARI //Research and implementation. – 2023.
23. Дадаханов, М. Х., and М. А. Асраев. "Алгоритмы предварительной обработки изображений рукописного текста." 1· 2019_ (1992): 69.
24. Фазылов, Ш. Х., Мирзаев, Н. М., Раджабов, С. С., Дадаханов, М. Х., & Асраев, М. А. (2020). Методы и алгоритмы идентификации личности на основе анализа изображений рукописного текста. Проблемы вычислительной и прикладной математики, (5), 5-26.

25. Abdurakhimovich U. A. THE FUTURE OF JAVASCRIPT: EMERGING TRENDS AND TECHNOLOGIES //FORMATION OF PSYCHOLOGY AND PEDAGOGY AS INTERDISCIPLINARY SCIENCES. – 2023. – Т. 2. – №. 21. – С. 12-14.
26. Abdurakhimovich U. A. THE POWER OF ENGLISH FOR PROGRAMMING. WHY IS ENGLISH IMPORTANT TO SOFTWARE DEVELOPERS? //MODELS AND METHODS FOR INCREASING THE EFFICIENCY OF INNOVATIVE RESEARCH. – 2023. – Т. 3. – №. 26. – С. 145-148.
27. O'rinboev A. STRATEGIC PROJECT MANAGEMENT FOR SCIENTIFIC WEB APPLICATIONS: LESSONS LEARNED AND FUTURE TRENDS //Current approaches and new research in modern sciences. – 2023. – Т. 2. – №. 9. – С. 9-13.
28. Ermatova Z. ZAMONAVIY DASTURIY MAHSULOTLAR YARATISH VA SIFATINI YAXSHILASHDA DASTURLASH TILLARINI O 'QITISHNING O 'RNI //Research and implementation. – 2023.
29. Ermatova Z. Q. TALABALARNING BILIMLARINI MASOFAVIY TA'LIM PLATFORMASI VA AN'ANAVIY USULDA O 'TKAZILGAN MASHG 'ULOTLAR BO 'YICHA BAHOLASH TURLARI VA ULARNING AFZALLIK HAMDA KAMCHILIKLARI //СОВРЕМЕННЫЕ ТЕНДЕНЦИИ РАЗВИТИЯ ФУНДАМЕНТАЛЬНЫХ И ПРИКЛАДНЫХ НАУК. – 2021. – С. 7-10.
30. Siddiqov M. Y. et al. MULTISERVISLI ALOQA TARMOQLARIDA AXBOROTLARGA TAHDID TURLARI //Educational Research in Universal Sciences. – 2023. – Т. 2. – №. 10. – С. 122-124.
31. Tohirov O. B. et al. MULTISERVISLI TARMOQ XAVFSIZLIGIDA NEYRON TARMOQLARINI O 'RNI //Educational Research in Universal Sciences. – 2023. – Т. 2. – №. 10. – С. 119-121.
32. Muhammadjonov A., TURLARI T. S. Y. T. ICHKI VA TASHQI YARIMO 'TKAZGICHLAR //Research and implementation.–2023.20. – Т. 23.
33. Ogli K. A. M. Modern programming languages: classification and characterizatioN //International Journal of Advance Scientific Research. – 2022. – Т. 2. – №. 11. – С. 108-111.
34. Azizjon Mo'minjon o'g X. et al. The Importance of Mathematical Game and Methods in the Formation of Mathematical Concepts in Primary Schools //Journal of Pedagogical Inventions and Practices. – 2022. – Т. 8. – С. 208-211.