e-ISBN : 978-629-98755-5-0
SIG : e-Learning@CS
https://appspenang.uitm.edu.my/sigcs/
Publication Date : 24 – Mac - 2025

COMPARATIVE STUDY FOR NOVICE ENGINEERING PROGRAMMER: C++ VS VBA PROGRAMMING LANGUAGES

Syarifah Adilah Mohamed Yusoff[1], *Saiful Nizam Warris[2], Mohd Saifulnizam Abu Bakar[3]
*syarifah.adilah@uitm.edu.my[1], *saifulwar@uitm.edu.my[2], mohdsaiful071@uitm.edu.my[3]*
[1,2,3]Jabatan Sains Komputer & Matematik (JSKM),
Universiti Teknologi MARA Cawangan Pulau Pinang, Malaysia

**Corresponding author*

ABSTRACT

The programming language significantly enhances students' problem-solving abilities; nonetheless, constructing a program is a complex endeavour due to insufficient metacognitive methods and the challenge of rectifying syntax errors. VBA dan C++ are both programming languages that covers all the fundamental features for novice programmers such as concept of declaration, input and output process and basic structure of programming such as sequence, selection and repetition. Descriptive survey has been done to 29 students in Degree of Civil Engineering to investigate the usability and compared the functionality of both languages in the learning process which consist of background, learning difficulties, practical applications, user experience and final thoughts. The outcome reveals that students have been more convenient with VBA for practical, beginner experience, ease of use and understand the concept to real-world problem solving. This result could serve as benchmarking on how to teach programming in a way novice can comprehend, thinking and learning in their perspective of studies.

Keywords: novice programmer, VBA, C++, programming language, comparative

Introduction

The role of programming languages in enhancing problem-solving skills among engineering students is unequivocally recognised. Learning to develop a functional program depends on comprehending the syntax and semantics of the specific language, which has proven to be the most formidable obstacles for students to overcome and excel.

In recent years, C++ has continued to advance, preserving its role as a robust and adaptable programming language. The C++20 standard introduced substantial improvements, such as ideas, coroutines, and modules, fundamentally transforming how developers compose and manage code (Crudu, 2024). These attributes have rendered C++ more efficient, comprehensible, and sustainable, addressing the requirements of contemporary software development. Moreover, the emphasis on performance enhancement via methodologies such as smart pointers and parallel programming has reinforced C++'s position in high-performance computing (adm-canpension, 2023a).

Conversely, VBA (Visual Basic for applications) is an essential instrument for automating processes within Microsoft Office programs. Notwithstanding speculation over its redundancy, VBA is extensively utilised owing to its profound connection with Excel, Word, and Access (*The Future of*

VBA: Debunking the "VBA Is Dead" Myth – Triple Your Pay with Excel, 2025). The capacity to automate monotonous tasks and provide customised capabilities has maintained VBA's relevance, particularly in sectors that depend significantly on Microsoft Office for their daily activities. The straightforwardness and availability of VBA makes it an appealing choice for non-programmers seeking to improve their productivity (Belfry, 2023).

The future of C++ appears auspicious, with continuous advancements focused on enhancing developer efficiency and code clarity. The next C++23 standard is anticipated to provide additional capabilities that will enhance the development process. Advancements in memory management and cross-platform development tools are enhancing C++'s versatility across diverse fields, ranging from embedded devices to huge business programs (adm-canpension, 2023a). The language is still a top option for developers looking for performance and versatility even as it develops (adm-canpension, 2023b).

Simultaneously, VBA's significance is enhanced by its extensive utilisation and user familiarity. The reliance of organisations on Microsoft Office sustains a robust need for VBA proficiency. The language's capacity to automate intricate operations and connect effortlessly with other Office products guarantees its ongoing significance (*The Future of VBA: Debunking the "VBA Is Dead" Myth – Triple Your Pay with Excel*, 2025; Belfry, 2023). Although contemporary technologies such as Python and JavaScript present alternatives, VBA's user-friendliness and direct relevance within Office applications render it a significant asset for numerous professionals (View, 2024). The significant of the comparison among both applications was presented in Table 1.

Table 1: Suggestion Table

Feature/Aspect	C++ programming	VBA programming
Complexity	High complexity, steep learning curve	Lower complexity, easier to learn
Performance	High performance, suitable for intensive tasks	Moderate performance, suitable for automation within Office apps
Use Cases	System software, game development, real-time simulations, high-performance	Automating tasks in Microsoft Office, creating macros, simple applications
Syntax	Complex syntax with extensive features	Simple and straightforward syntax
Memory Management	Manual memory management (pointers)	Automatic memory management
Object-Oriented	Fully supports object-oriented programming	Limited support for object-oriented programming
Integration	Can be integrated with various systems and platforms	Deep integration with Microsoft Office applications
Career Opportunities	High demand in various industries like finance, game	Limited to roles involving Microsoft Office automation

	development, and embedded systems	
Community and Resources	Large and active community with extensive resources	Smaller community, but plenty of resources for Office automation
Learning Curve	Steeper learning curve, but provides a strong foundation for other languages	Gentler learning curve, suitable for beginners and non-programmers

Methodology

This study was conducted by collecting primary data directly from 29 number of the degree students of Civil Engineering in Universiti Teknologi MARA Cawangan Pulau Pinang. Online survey was delegated for them to answer some questions regarding to respondent background in terms of programming level, learning difficulties, practical applications, user experiences and final thoughts. The questions were carefully designed to fulfil the following objectives by considering the elements of easy learning of computer programming language for novice:

I. How effectives the learning software both for C++ and VBA in engaging students throughout teaching and learning session?
II. What is students' attitude towards the use of both programming language for practical application?
III. Which programming language is more versatile to be learned for novice programmer?

The questionnaire responses by the respondents were calculated using descriptive analysis of which frequencies and percentages were tabulated in order to better comprehend the result.

Result and discussion

Primarily as a background study, all 29 respondents have considered themselves as novice programmers with basis understanding and able to write a simple programming codes independently. All of them were experience in writing C++ programs and recently accomplished VBA programming language for another course. Figure 1 illustrates the level of difficulties between C++ and VBA in four levels which are easy, moderate, difficult and very difficult. According to the result 93 percent of students vote that easy and moderate for VBA, meanwhile 52 percent vote for difficult and very difficult for C++. This result clearly indicate students were find VBA easier to understand and learn.

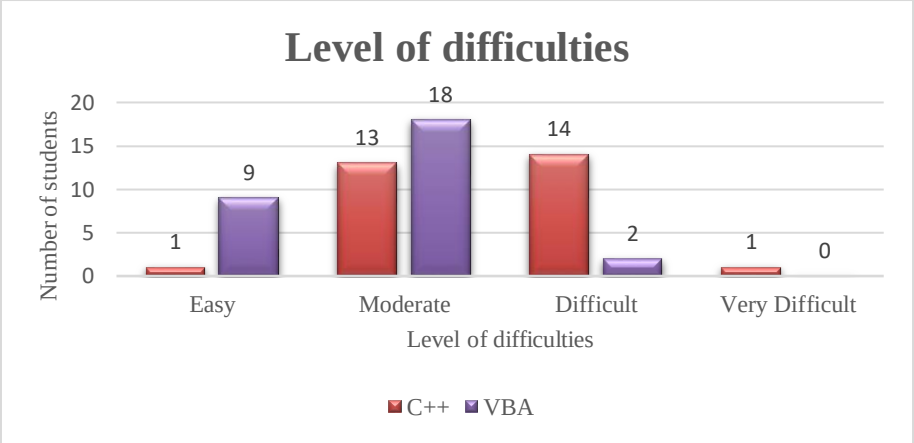

Figure 1: Comparison level of difficulties in learning

Learning time indicates appropriate time for students to catch up with each learning session after class. Figure 2 represents a week of learning progress taken by students and was split into four intervals of less than a day, 1 to 3 days, 4 to 7 days and more than 7 days of learning. The result shows that 70 percents of the students able to understand VBA within 3 days, meanwhile C++ taken longer time where 66 percent require more than 4 days and up to a week to comprehend with the C++ lesson.

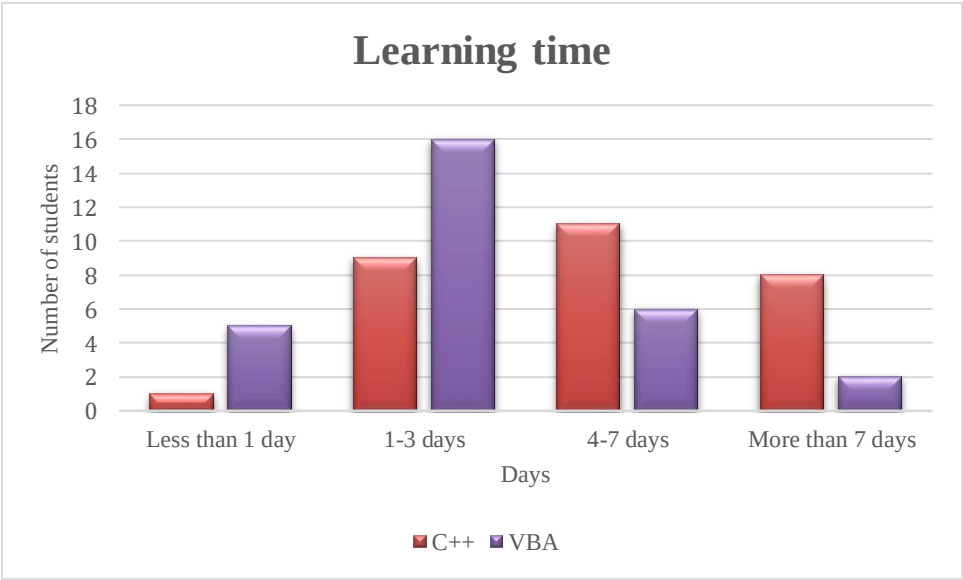

Figure 2: Comparison level of learning time for revision at each lesson learning in class

Regarding learning tools that facilitate better comprehension and further exploration of knowledge, it is noteworthy that over fifty percent agrees that classroom courses are the most beneficial learning resources, encompassing both instructional and practical learning processes for both languages.

Meanwhile, online tutorials, books, and trial-and-error options garnered less than 2 percent of their votes. Figure 3 illustrates the learning resources voted by the students.

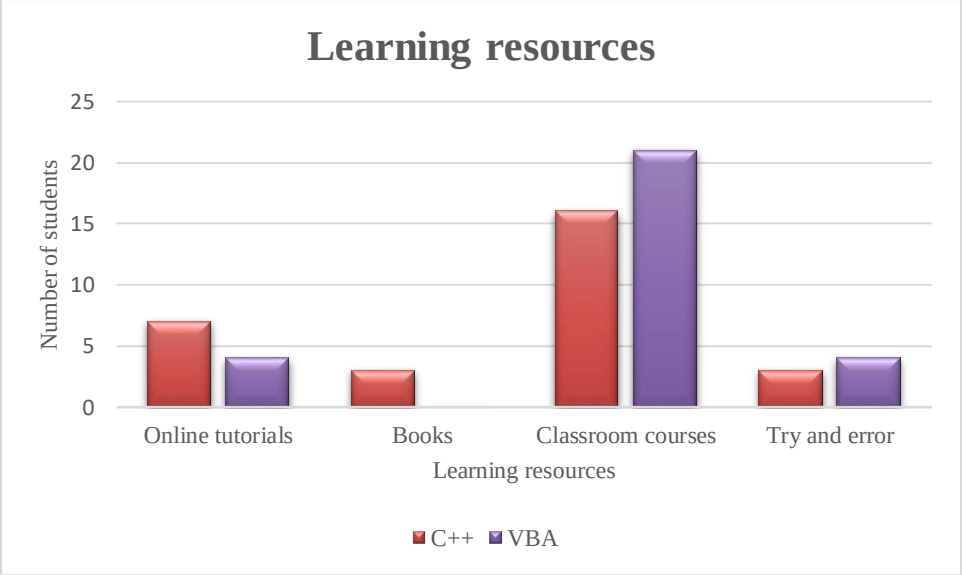

Figure 3: Comparison level of learning time for revision at each lesson learning in class

The next illustration in Figure 4 demonstrates how students perceive the constructed programming code as beneficial in addressing and resolving real-world problems. Both languages demonstrated their significance to the students, as over half of them deemed them useful or very useful. VBA received the highest score with 24 votes, while C++ garnered 21 votes. While C++ significantly contributes to careers in real-world applications, VBA demonstrates more utility for end users in daily tasks, showcasing its widespread applicability across various purposes.

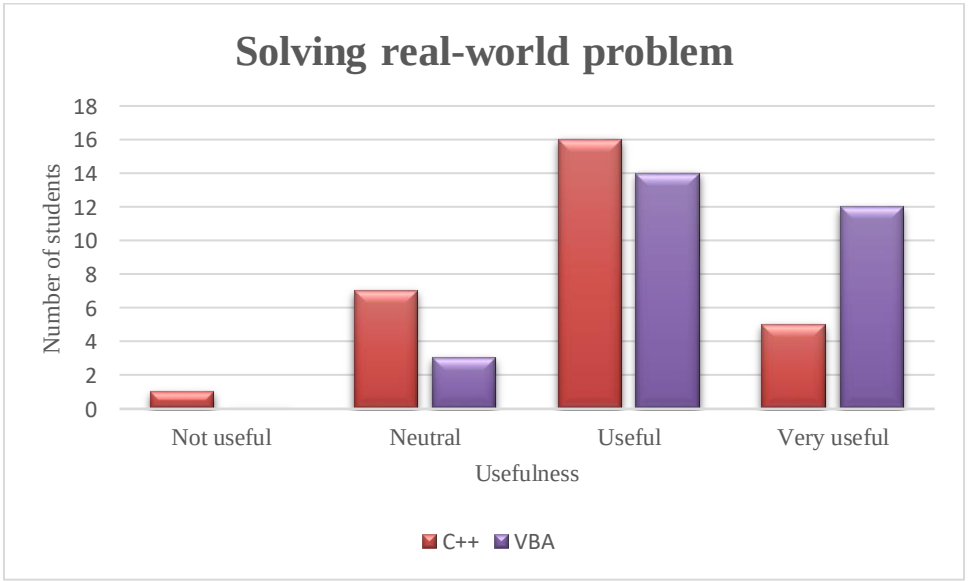

Figure 4: Comparison level of contribution to serve real-world problems

The prior vote is likely substantiated by the subsequent comparison about the diversity of functional tasks. Figure 5 illustrates the versatility of tasks pertaining to both general and automated tasks as evaluated by the students. The VBA achieves scores of 86 percent and 93 percent, respectively, for general and automated tasks in comparison to C++. This occurred because the implementation of VBA encompasses both coding and the end-user interface, allowing students to comprehend the process and application comprehensively from source code to completion. Regardless of both languages, students have acquired all the fundamental concepts of coding, beginning with variable declaration and the use of sequential, selection, and repetition programming structures.

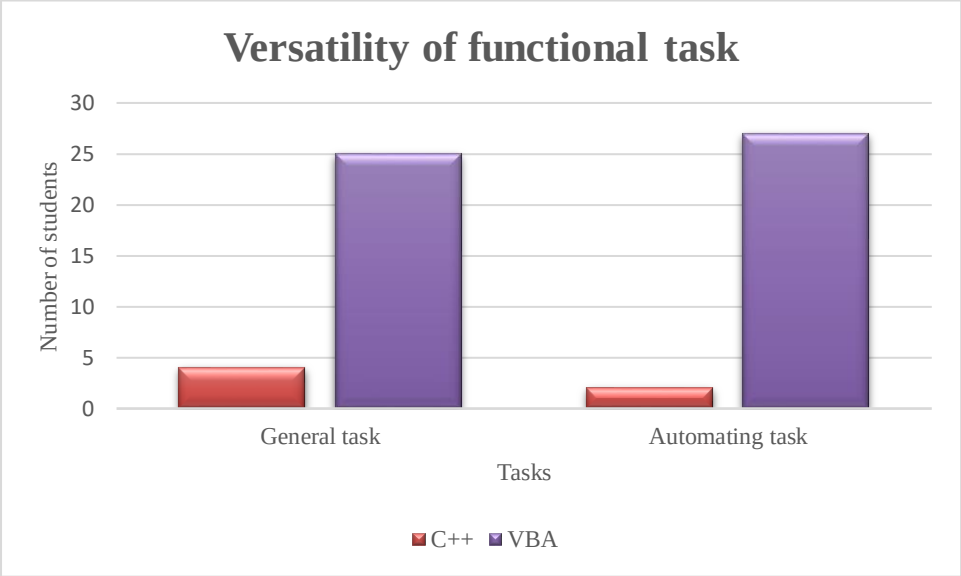

Figure 5: Comparison level of versatility of the languages for novice programmer learning process

The debugging process pertains to how students rectify their code syntax upon the detection of an error. Figure 6 delineates debugging levels categorised into four tiers: easy, moderate, difficult, and very difficult. Overall, VBA received the highest votes for moderate difficulty, with none classified as very tough, but C++ was predominantly chosen as difficult, with 2 percent rated as very difficult.

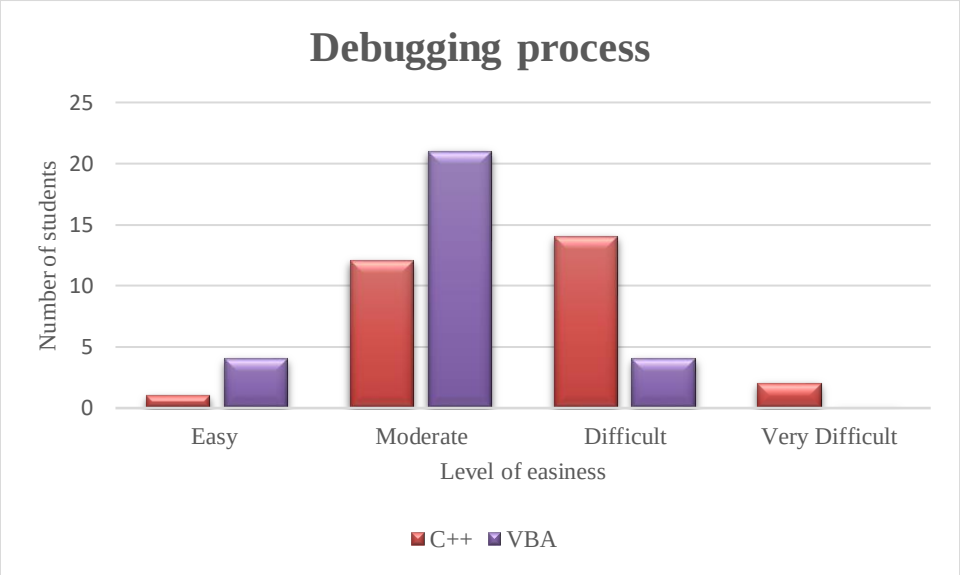

Figure 6: Comparison level of debugging experiences

The development of the environment is based upon the application utilised for writing, compiling, and executing the output. Figure 7 illustrates that VBA scores for neutral, intuitive, and very intuitive categories average approximately 10 votes, whereas C++ received 66 percent for neutral and 34 percent for intuitive. The implementation of DEV C++ emphasises students' understanding of syntax and semantics, facilitating basic output without GUI interaction. In contrast, VBA is centred on interface design and is completely connected with MS Office products, which are prevalent among users in Malaysia, both personally and in enterprises.

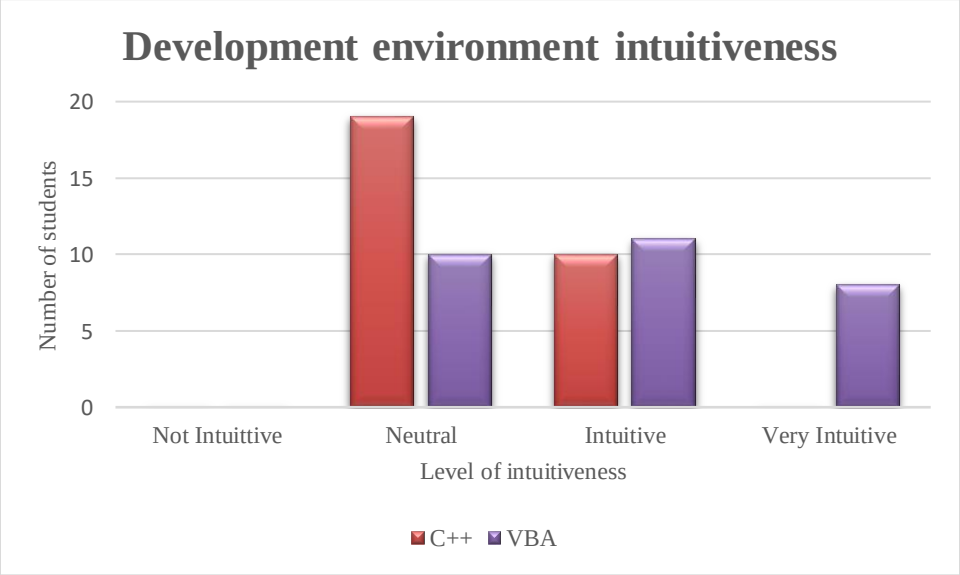

Figure 7: Comparison level of development environment

The satisfaction derived from learning is manifested in students' enjoyment of the educational experience. Figure 8 illustrates the votes for four levels of satisfaction, ranging from the least enjoyable to the most enjoyable: not enjoyable, neutral, enjoyable, and very enjoyable. Approximately 90 percent of students rated VBA as either enjoyable or very enjoyable, whereas 61 percent of students rated C++ as neither enjoyable or neutral. This sentiment arises mostly from the user-friendly nature of VBA program outputs, which seamlessly merge with the working interface, providing it more suitable and valuable for novices, particularly when students comprehend their expectations from real-world problems.

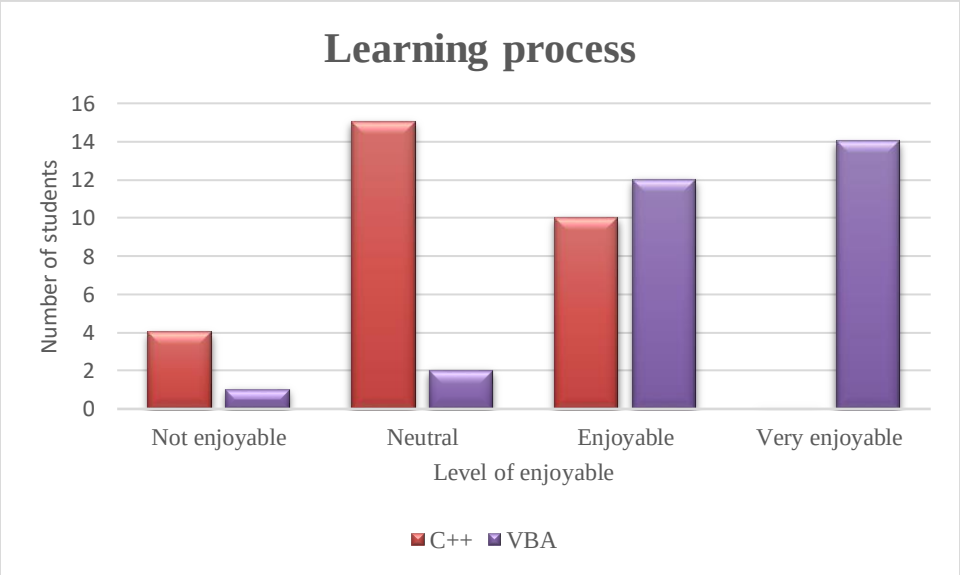

Figure 8: Comparison levels of enjoyment of the learning process

Conclusion

This study has fulfilled the objectives outlined in the methodology. C++ and VBA significantly influence the programming learning process for new programmers, particularly those with a background in engineering. They comprehended both approaches for writing source code and developed a foundational programming skill using them. Their attitudes towards learning aligned with expectations, since VBA promotes beneficial outcomes such as reduced stress and enjoyment, along with being straightforward and comprehensible for both general and automated tasks, intuitive, and easily applicable in real-world scenarios. Despite C++ offering numerous advantages for various facets of a software development career, students like VBA as an introductory learning tool since it allows them to comprehend its application to real-world problems around them.

References:

adm-canpension. (2023a, December 6). *Canada Pension Plan*. Canada Pension Plan. https://canpension.ca/articles/the-future-of-c-what-to-expect-in-2026

adm-canpension. (2023b, December 6). *Canada Pension Plan*. Canada Pension Plan. https://canpension.ca/articles/cpp-in-2024-the-future-of-c-programming-language

Belfry, M. (2023, August 7). *Is Excel VBA Still Worth Learning in 2024? - Onsite Training*. Onsite Training. https://www.onsite-training.com/is-excel-vba-still-worth-learning/

Crudu, A. (2024, October 29). *Keeping Up with the Latest Trends What Every C++ Developer Needs to Know*. Moldstud.com. https://moldstud.com/articles/p-keeping-up-with-the-latest-trends-what-every-c-developer-needs-to-know

OttoBotCode. (2022, February 9). *C++ GUI Programming For Beginners | Episode 0 - Introduction*. YouTube. https://www.youtube.com/watch?v=BjQhp0eHmJw

The Future of VBA: Debunking the "VBA is Dead" Myth – Triple Your Pay With Excel. (2025). Tripleyourpaywithexcel.com. https://www.tripleyourpaywithexcel.com/2024/10/15/the-future-of-vba-debunking-the-vba-is-dead-myth

View. (2024, October 27). *VBA Will Not Be Obsolete in the Next few Years*. Chanmingman's Blog. https://chanmingman.wordpress.com/2024/10/27/vba-will-not-be-obsolete-in-the-next-few-years/