Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025 ASEE
Paper ID #45526

Survey of C/C++ IDEs for a First Year Programming Course

Mr. Thomas Rossi, University of New Haven

Thomas Rossi is the Assistant Chair of the University of New Haven's Electrical and Computer Engineering and Computer Science department. His research focuses on improving the post-secondary experience for students through the use of current computing tools and technologies. Thomas graduated with his MS in Computer Science from the University of New Hampshire in 2016. He has previously worked at the Rochester Institute of Technology and at Penn State Erie, the Behrend College.

Prof. Alice E. Fischer, University of New Haven

Survey of C/C++ IDEs for a First Year Programming Course

Abstract

In our first-semester programming course, we seek to help our students understand how to read code as well as how to test and debug their code, also we seek to help students understand how to write programs that are functional, informative, use basic control structures, and are well documented. We look to accomplish these goals using the C and C++ programming languages as well as an integrated development environment (IDE).

The challenge is selecting the correct IDE, given the multitude of options that are possible. Compounding the problem is the fact that some IDEs now include AI tools designed to aid professional programmers with writing and debugging/testing code. While these AI tools are beneficial in a professional setting, we believe this kind of ”help” does not help students build a strong foundation. To determine the IDE for our course we first began by creating a series of selection criteria. Our criteria for selecting an IDE were as follows:

1. The IDE must adhere to the C/C++ language standard.
2. The IDE must not have AI assistance or the AI assistance must be behind a paywall to prevent student use.
3. The IDE should be a popular IDE in industry for the C/C++ programming language.
4. The IDE should be cross-platform.

With these criteria in hand, we then looked at several popular IDEs in the C/C++ space including Visual Studio Code, Xcode, and CLion as well as online environments such as xyBooks xyLab and Replit.

In this paper, we will discuss the findings of our survey and which IDE was selected as the IDE of choice for our first programming course. We will begin by discussing how our selection criteria were determined, followed by discussing each of the aforementioned IDEs. After this we will analyze the IDEs against the selection criteria, and finally we will identify which IDE was the winner.

1 Introduction

In our first-semester programming course, we seek to help our students understand how to read code as well as how to test and debug their code, we also seek to help students understand how to write programs that are functional, informative, use control structures, and are well documented.

We seek to accomplish these goals using the C programming language with an integrated development environment (IDE).

The challenge is selecting the correct IDE, given the multitude of options that are possible. Compounding the problem is the fact that some IDEs now include AI tools designed to aid professional programmers with writing and debugging/testing code. We eschew these tools because we believe that students need to do their own programming and debugging work to learn how to program.

In the Spring of 2024, the faculty teaching the introductory course worked to determine which IDE would be best suited to the needs discussed above. A survey of common IDEs was conducted and the tools were compared against the criteria listed above. This paper will begin by discussing the IDEs that were examined, it will then proceed to discuss the criteria that were selected and used by the faculty, and then evaluate the IDEs against the criteria. We will conclude by indicating our winner and the reasons why it was chosen along with some minor shortcomings we have noted.

2 Background

2.1 Integrated Development Environments

The Integrated Development Environment (IDE) is a one stop shop for programmers. It provides all of the tools necessary to write, debug, and run code in one place. For this survey, several IDEs were examined. A brief description of each one is provided below.

2.1.1 Visual Studio Code

Visual Studio Code is an IDE developed by the Microsoft Corporation. Microsoft touts this polyglot IDE as being able to handle any language a programmer wishes to use[1]. This IDE has strong industry adoption as indicated by the JetBrains State of the Developer Ecosystem survey[2].

This IDE is also used in Computer Science courses in post-secondary education. The paper by Tan et al. discusses the deployment of this IDE in an introductory CS course with students rating the IDE highly in terms of its visual appeal, debugging capabilities, and editing capabilities. Students overall also had a high level of satisfaction with the IDE[3].

2.1.2 Xcode

XCode is developed by Apple and is marketed as a way for developers to create apps for all of Apple's operating systems (iOS, macOS, etc.). While not its primary function, this IDE also has the ability to handle C and C++ code. The IDE is freely available through the Apple App Store[4]. It is bundled with the free developer tools and can be downloaded by any Mac owner. However, it is only available to Mac users and has a steep learning curve.

2.1.3 CLion

CLion, which its creator JetBrains asserts is ”A single IDE for all your C and C++ needs”[5], has a UI closely modeled after other popular IDEs such as IntelliJ and PyCharm. CLion does require the user to download a compiler separately (Cygwin with either gcc or MinGW on PC, XCode Developer Tools on Mac, and clang on Linux). This IDE does support development in other languages such as HTML as well as frameworks such as React and Bootstrap.

2.1.4 xyBooks xyLab

xyLabs is a web-based tool that requires no configruation on the student's part. xyLab is a part of xyBooks which is owned by Wiley. xyLabs prides itself on several features including a built in autograder, easy student submission, and easy assignment generation [6].

In 2023, xyLab was updated with a host of new features to create ”Advanced xyLabs”. This included Linux integration, new tools for instructors, and auto-grading functionality. zyLab can also be integrated with an IDE such as Visual Studio code to allow students to work in a desktop IDE versus a web based environment [7].

2.1.5 Replit

Replit is similar to xyBooks in that it is a web based development environment. Replit boasts significant AI capabilities designed to help with creating applications quickly and is used by over 30 million creators [8]. The AI tools that Replit has allow for a programmer to briefly describe the kind of app they wish to generate and the AI will provide assistance in generating the code to complete the assigned task [9].

2.2 IDEs Excluded from Consideration

To be clear, there are more IDEs for C/C++ than what are listed here. In order for an IDE to be a candidate for adoption, it needed to have industry adoption and have a non-trivial share of the professional market. To determine this, the JetBrains State of Developer Ecosystem survey was used. Any IDE considered here had over ten percent adoption. In the case of xyBooks and Replit, while these are not industry adopted tools nor are they on the JetBrains State of Developer Ecosystem they were tools referenced by members of the department and were thus given consideration.

3 Determining Selection Criteria

To determine the best IDE for the course, a clear set of criteria needs to be established. The four criteria that were settled upon by the course faculty were: Adherence to language standards, AI integration, industry acceptance, and cross platform usability. These criteria are discussed in more detail below. Note that requiring the tool to integrate with our LMS is not a requirement. We do not need “integration”, only the ability of students to upload their work to our LMS which is a trivial enough procedure.

3.1 Adherence to Language Standards

It is important that the students worked with an IDE that adheres to C and C++ language standards. This means that any IDE that requires publisher-specific header files in the C / C++ code or automatically injects a header file a student forgot at compile time is unacceptable. Adherence to language standards does not mean that an IDE could only support C and C++ to meet this criteria. The IDE could be polyglot and still be acceptable. Additionally, the support for C/C++ should be "out of the box": the student should not have to go and search for additional plugins to provide functionality for programming in C/C++.

3.2 AI Integration

While AI assistants are useful tools in many situations, at the introductory level it is crucial that students learn the fundamentals of coding and debugging. For these reasons, any IDE that gives a student free access to an AI assistant that was specifically trained in programming would not be acceptable. If the AI assistant were behind a paywall this would be marginally okay, since students have limited resources with which to purchase such tools.

While some institutions are adopting a philosophy of integrating AI into the introductory courses as an aid, this runs counter to our departmental philosophy of pushing students to develop a solid foundation without AI assistance. Our goal in doing this is to encourage our students to have the ability to develop anything they need to when the AI is not capable / available to assist.

3.3 Industry Acceptance

Any tool that is selected must be something that is used in industry. This does not mean the IDE must be the most popular, but there should be demonstrable evidence the IDE is commonly used. The IDE can either be from a software development company or an open source IDE that is distributed under the GNU license.

3.4 Cross Platform Usability

As students at the university are permitted to use either Windows, MacOS, or Linux, the IDE that is chosen must be cross-platform. To be considered "cross platform" the IDE must present the same feature set to the user regardless of operating system chosen. Slight variations in the presentation layer are acceptable since the design paradigm across the major operating systems varies.

4 Results

In examining the above criteria, several IDEs are quickly eliminated. Both Replit and xyLab are not industry standard tools[2] and Replit also boasts an AI assistant as available to students to help with coding[9]. Xcode was also quickly eliminated as it is only available for use in MacOS and does not support Windows or Linux.

CLion and Visual Studio Code are both industry standard tools[2], have AI assistants that are behind paywalls[10][11], and are available across all of the major operating systems. However, in terms of the adherence to language standards Visual Studio Code does not have support for C/C++ out of the box and instead requires the user to download a plugin to enable C/C++ programming[12]. This is not an ideal experience for a student just beginning to code. Downloading the wrong plugin can be problematic for a student and cause a myriad of issues that can be easily avoided. This observation is corroborated by Tan et al as they had to provide very clear guidance to students to help them get the Python plugin for VS Code[3].

It is for these reasons that CLion was chosen as the IDE of choice for our first year courses. It is cross platform, its AI assistant is not free, and it is commonly used in industry. A student does not need to worry about having the correct plugins for coding and can more easily get their home environment up and running.

4.1 Student Reception

While a formal survey of student satisfaction with CLion was not done, after the change the course evaluations were monitored for any indications of student dissatisfaction with the IDE. In the three semesters since this change, there have been no documented complaints regarding the IDE in the course evaluations.

4.2 CLion Shortcomings

CLion does have some minor shortcomings with how it handles buffered versus unbuffered output. These shortcomings are not encountered in introductory courses and do not impede a student's ability to meet the learning outcomes set forward in our curriculum.

5 Conclusion

Given its cross-platform nature, adherence to industry standards, and also its restricted access to AI tools, CLion is the supported IDE for our introductory course. This IDE can also be applied to other courses in our curriculum and also sets students up to easily transition to other languages given CLion is part of a larger family of tools.

References

[1] Microsoft. https://code.visualstudio.com, 2024. [Accessed 12-11-2024].

[2] "C Programming - The State of Developer Ecosystem in 2021 Infographic — jetbrains.com." https://www.jetbrains.com/lp/devecosystem-2021/c/. [Accessed 09-15-2024].

[3] J. Tan, Y. Chen, and S. Jiao, "Visual studio code in introductory computer science course: An experience report," *arXiv preprint arXiv:2303.10174*, 2023.

[4] Apple. https://developer.apple.com/xcode/, 2024. [Accessed 12-11-2024].

[5] JetBrains. https://www.jetbrains.com/clion/, 2024. [Accessed 12-02-2024].

[6] xyBooks. https://www.zybooks.com/catalog/zylabs-programming/, 2024. [Accessed 01-01-2025].

[7] C. Y. D. Leung, J. Mazzone, E. Kazakou, C. Gordon, A. D. Edgcomb, and Y. Rajasekhar, "A powerful labs environment for computer science courses," in *2024 ASEE Annual Conference & Exposition*, 2024.

[8] Replit, "Ai-replit." https://replit.com/, 2024. [Accessed 01-01-2025].

[9] Replit, "Ai-replit." https://replit.com/ai, 2024. [Accessed 12-04-2024].

[10] JetBrains. https://www.jetbrains.com/ai/#plans-and-pricing, 2024. [Accessed 12-02-2024].

[11] GitHub. https://docs.github.com/en/billing/managing-billing-for-your-products/managing-billing-for-github-copilot/about-billing-for-github-copilot, 2021. [Accessed 12-02-2024].

[12] K. Yip, "How to set up c++ in visual studio." https://www.youtube.com/watch?v=DMWD7wfhgNY&t=325s, Winter 2023. [Accessed 12-02-2024].