Programming Language Research Assignment 1

Algonquin College
School of Advanced Technology
Computer Programming

Onur Önel
Mazin Abou-Seido

23F_CST8333_360

Submitted September 15, 2023

A technical report submitted to Algonquin College in partial fulfillment of the requirements for a diploma in Computer Programming

Introduction

This report seeks to delve into the choice of the C++ programming language in the contemporary tech landscape due to its versatility, efficiency, and industry relevance. To set the context, three contemporary websites detailing popular programming languages will be identified and described to ascertain current trends. The reputation of these websites will be determined based on set criteria and cited per the IEEE referencing format. Subsequently, a closer look at C++'s alignment with future career objectives will be explored.

In parallel, the study will shed light on unit testing for C++, with a focus on two specific framework-based unit testing websites. A thorough assessment will be done to ascertain the relevance of the selected unit testing framework to the course. Technical details, including the operating system, programming language version, and IDE build number, will be duly documented. The study's progression will be structured using a Work Breakdown Structure for the Practical Project Part 1. To better visualize the WBS, a Gantt Chart will be crafted using tools like MS Project or Project Libra, with a screenshot included for reference. Both the Gantt Chart and MS Project files will be presented.

Table of Contents

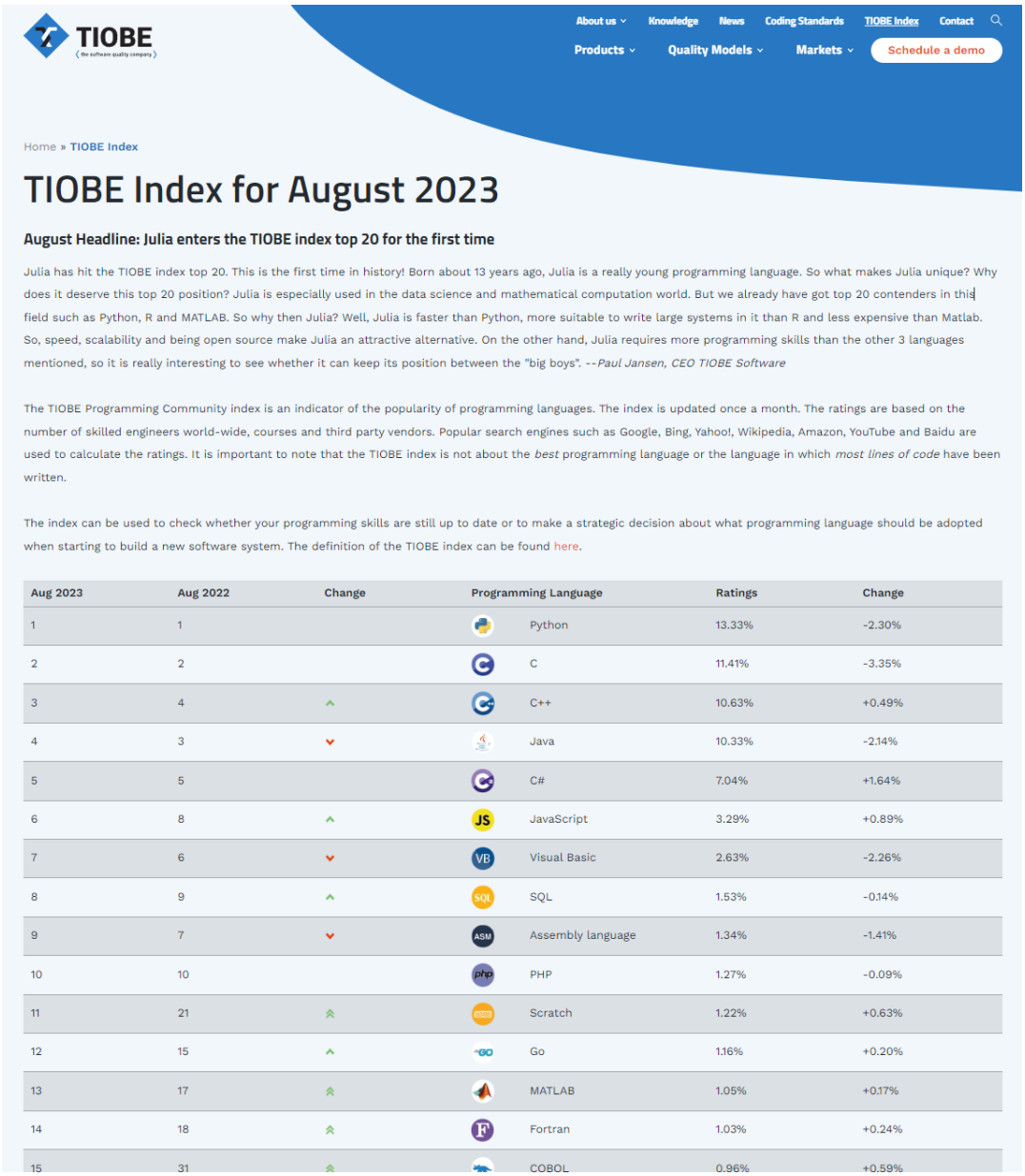
Research for Programming Language .....
TIOBE Programming Community Index .....
RedMonk Programming Language Rankings .....
PYPL (PopularitY of Programming Language) Index .....
Choice of Programming Language for Study (C++).....
Alignment with My Career Goals.....
Framework based Unit Testing – Catch2 .....
Framework based Unit Testing – Google Test .....
Chosen Unit Test Framework and Rationale for Selection .....
Development Tools for C++.....
IDE Platform Features, Supported Operating Systems, and Tool Integration .....
Chosen IDE, Licensing, Restrictions, and Compliance (CLion) .....
Advancing with CLion 2023.2 & Catch2 v3.4.0.....
WBS and Gantt Chart for Practical Project .....
References .....

Research for Programming Language

Technology is undergoing significant changes, and new programming languages are emerging. In terms of businesses and developers, it's essential to keep track of which programming languages are trending. After my research, I've determined that reliable indices such as TIOBE, RedMonk, and PYPL provide insights into the most popular programming languages among users.

TIOBE Programming Community index

The TIOBE index is one of the most cited indices to track the popularity of programming languages over time. They update their rankings monthly based on search engine queries.



TIOBE

About us | Knowledge | News | Coding Standards | TIOBE Index | Contact

Products | Quality Models | Markets | Schedule a demo

Home » TIOBE Index

TIOBE Index for August 2023

August Headline: Julia enters the TIOBE index top 20 for the first time

Julia has hit the TIOBE index top 20. This is the first time in history! Born about 13 years ago, Julia is a really young programming language. So what makes Julia unique? Why does it deserve this top 20 position? Julia is especially used in the data science and mathematical computation world. But we already have got top 20 contenders in this field such as Python, R and MATLAB. So why then Julia? Well, Julia is faster than Python, more suitable to write large systems in it than R and less expensive than Matlab. So, speed, scalability and being open source make Julia an attractive alternative. On the other hand, Julia requires more programming skills than the other 3 languages mentioned, so it is really interesting to see whether it can keep its position between the "big boys". --*Paul Jansen, CEO TIOBE Software*

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the *best* programming language or the language in which *most lines of code* have been written.

The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found here.

Aug 2023	Aug 2022	Change	Programming Language	Ratings	Change
1	1		Python	13.33%	-2.30%
2	2		C	11.41%	-3.35%
3	4	↑	C++	10.63%	+0.49%
4	3	↓	Java	10.33%	-2.14%
5	5		C#	7.04%	+1.64%
6	8	↑	JavaScript	3.29%	+0.89%
7	6	↓	Visual Basic	2.63%	-2.26%
8	9	↑	SQL	1.53%	-0.14%
9	7	↓	Assembly language	1.34%	-1.41%
10	10		PHP	1.27%	-0.09%
11	21	⇈	Scratch	1.22%	+0.63%
12	15	↑	Go	1.16%	+0.20%
13	17	⇈	MATLAB	1.05%	+0.17%
14	18	⇈	Fortran	1.03%	+0.24%
15	31	⇈	COBOL	0.96%	+0.59%

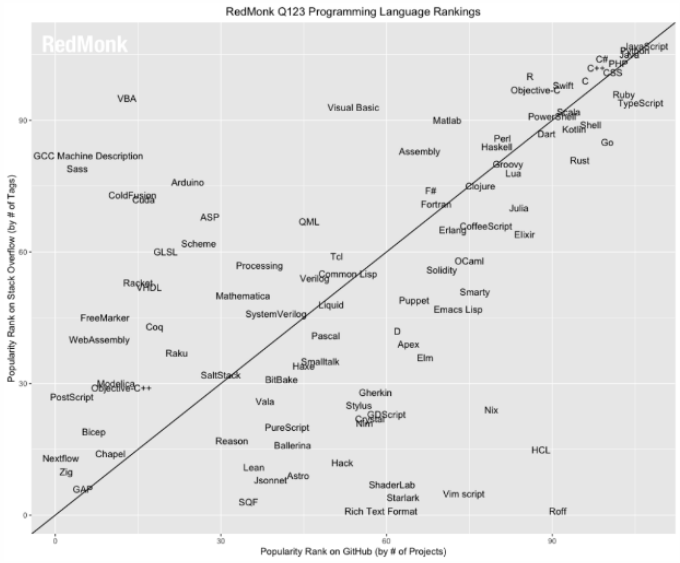
"Tiobe index," TIOBE, https://www.tiobe.com/tiobe-index/ (accessed Sep. 11, 2023).

RedMonk Programming Language Rankings

RedMonk updates its rankings twice a year and bases its rankings on data from GitHub (code repositories) and Stack Overflow (discussions).

Upon examining the rankings of programming languages, it becomes clear that languages like Python, JavaScript, and Java consistently stay near the top. Their versatility, ranging from website creation to data analysis, explains their popularity. Specialized languages are also on the rise. For instance, R is primarily for data science, while Swift is for making apps on Apple devices. Their growing popularity highlights a trend towards task-specific languages.

With that, here is the first quarter plot for 2023.



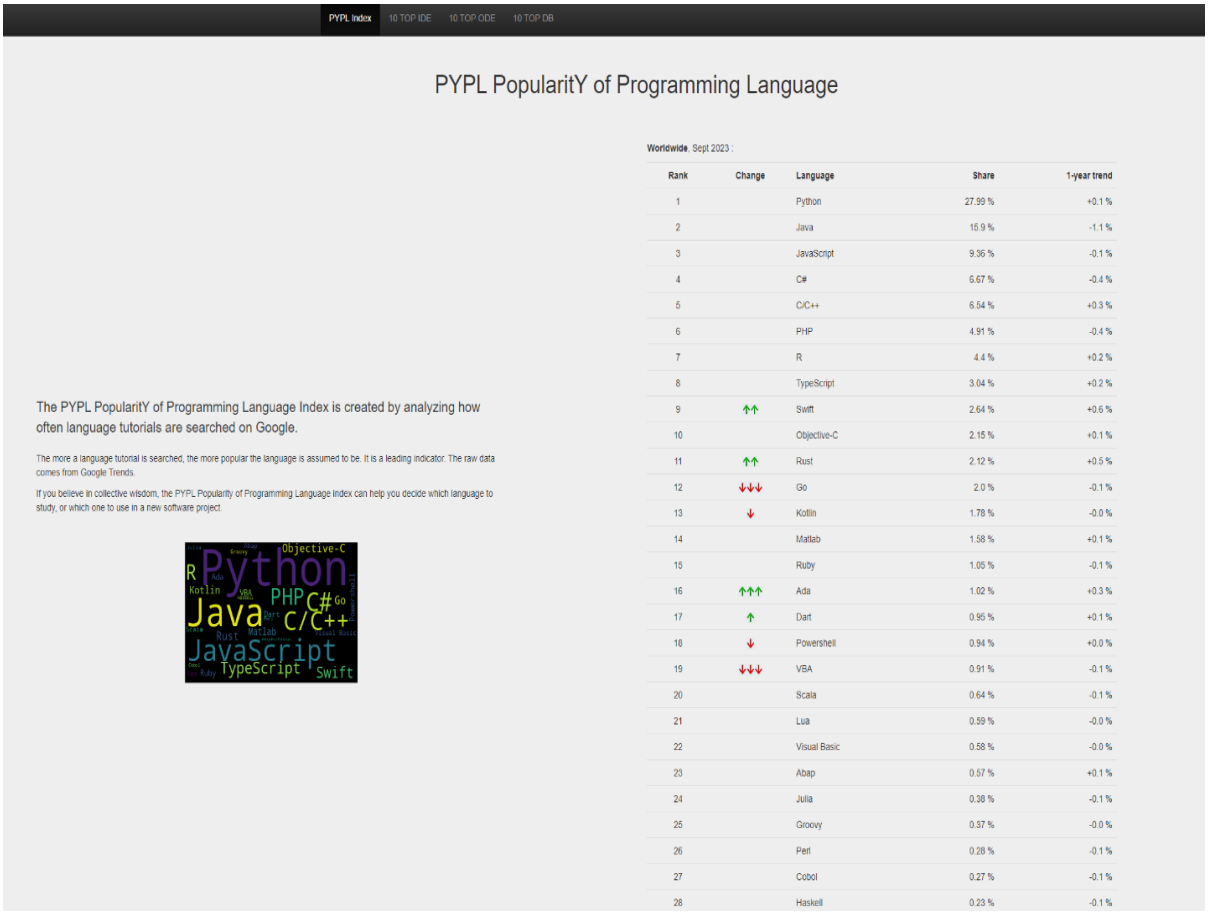
1 JavaScript
2 Python
3 Java
4 PHP
5 C#
6 CSS
7 TypeScript
7 C++
9 Ruby
10 C
11 Swift
12 Shell
12 R
14 Go
14 Scala
16 Objective-C
17 Kotlin
18 PowerShell
19 Rust
19 Dart

S. O. |@sogrady |May 16, S. O'Grady, and Name, "The Redmonk Programming Language Rankings: January 2023," tecosystems, https://redmonk.com/sogrady/2023/05/16/language-rankings-1-23/ (accessed Sep. 11, 2023).

PYPL (PopularitY of Programming Language) Index

The PYPL index is based on Google search trends for tutorials in different languages, making it a good indicator of which languages people are actively trying to learn.

In another significant development in this field, open-source languages continue to gain traction. Being open-source means that anyone can view and contribute to the language's development. Rust and Go serve as examples of these increasingly popular open-source languages. Newer languages are also making waves due to their adaptability to current needs. Kotlin, for example, is emerging as a favored choice over Java for Android app development.



PYPL Index | 10 TOP IDE | 10 TOP ODE | 10 TOP DB

PYPL PopularitY of Programming Language

The PYPL PopularitY of Programming Language Index is created by analyzing how often language tutorials are searched on Google.

The more a language tutorial is searched, the more popular the language is assumed to be. It is a leading indicator. The raw data comes from Google Trends.

If you believe in collective wisdom, the PYPL Popularity of Programming Language Index can help you decide which language to study, or which one to use in a new software project.

Worldwide, Sept 2023 :

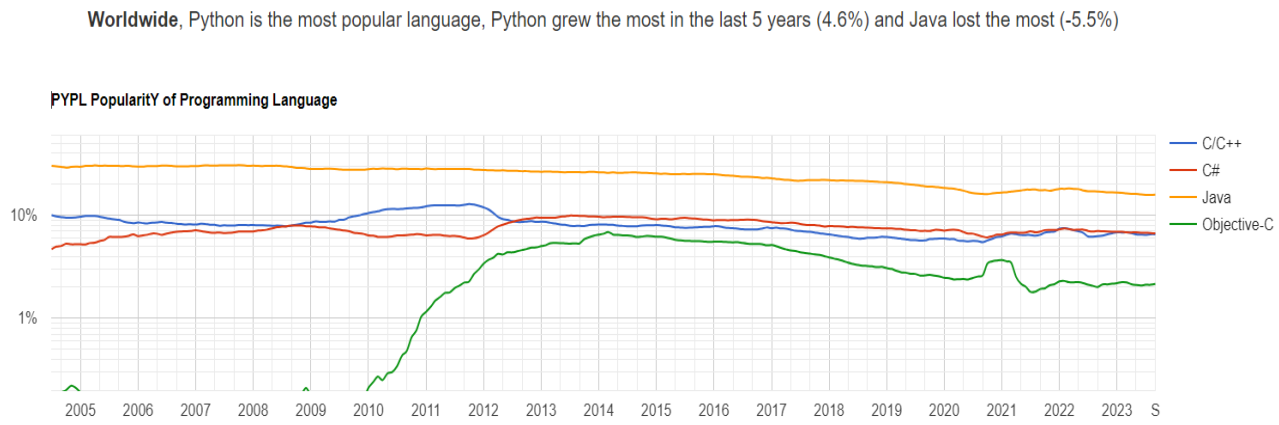
Rank	Change	Language	Share	1-year trend
1		Python	27.99 %	+0.1 %
2		Java	15.9 %	-1.1 %
3		JavaScript	9.36 %	-0.1 %
4		C#	6.67 %	-0.4 %
5		C/C++	6.54 %	+0.3 %
6		PHP	4.91 %	-0.4 %
7		R	4.4 %	+0.2 %
8		TypeScript	3.04 %	+0.2 %
9	↑↑	Swift	2.64 %	+0.6 %
10		Objective-C	2.15 %	+0.1 %
11	↑↑	Rust	2.12 %	+0.5 %
12	↓↓↓	Go	2.0 %	-0.1 %
13	↓	Kotlin	1.78 %	-0.0 %
14		Matlab	1.58 %	+0.1 %
15		Ruby	1.05 %	-0.1 %
16	↑↑↑	Ada	1.02 %	+0.3 %
17	↑	Dart	0.95 %	+0.1 %
18	↓	Powershell	0.94 %	+0.0 %
19	↓↓↓	VBA	0.91 %	-0.1 %
20		Scala	0.64 %	-0.1 %
21		Lua	0.59 %	-0.0 %
22		Visual Basic	0.58 %	-0.0 %
23		Abap	0.57 %	+0.1 %
24		Julia	0.38 %	-0.1 %
25		Groovy	0.37 %	-0.0 %
26		Perl	0.28 %	-0.1 %
27		Cobol	0.27 %	-0.1 %
28		Haskell	0.23 %	-0.1 %

In conclusion, the world of programming languages is ever-evolving and highly dynamic. As new languages emerge and old ones evolve, developers and industries alike shift their preferences based on various factors such as performance, security, and ease of use. While it is essential to note that no single language is the best fit for all tasks, the rankings of these languages provide invaluable insights.

"PYPL popularity of Programming Language index," index, https://pypl.github.io/PYPL.html (accessed Sep. 11, 2023).

Choice of Programming Language for Study (C++)

I chose C++, which has been an extension of the C language from the 1970s to the present. The object-oriented features, the capacity for low-level memory management, and the ability to develop high-performance applications greatly influenced my decision. C++ for the study is relevant in today's job market. As foundational technologies, like operating systems and databases, are primarily written in C or C++, proficiency in these languages remains highly valued. Lastly, studying C++ can act as a stepping stone to learning other programming languages more effortlessly. Its syntax and conventions have influenced numerous modern languages like Java and C#. In conclusion, C++ provides a robust platform for both academic exploration and real-world application. Choosing the right programming language for study can shape a career, influence academic pursuits, and dictate the types of projects one can undertake.



"PYPL popularity of Programming Language index," index, https://pypl.github.io/PYPL.html (accessed Sep. 11, 2023).

C++ Alignment with My Career Goals

When I think about the next steps in my career, I've outlined a distinct plan for myself. After finishing my diploma at Algonquin College, I'm determined to dive deeper into Computer Science or Software Engineering. This decision isn't just about personal interest; it's also about staying ahead. In my research, one thing became clear: C++ is not just a passing topic in the tech world. It's a foundational subject that many top universities around the world emphasize in their courses.

But my interest in C++ goes beyond just its academic importance. When browsing through job opportunities and talking to industry professionals, I've noticed how frequently C++ appears as a desired or essential skill. It's evident that companies value individuals who have a strong grasp of this language. This realization has motivated me to invest more of my time and energy into understanding C++ inside and out. By focusing on C++ now, I'm not only preparing myself to excel in my upcoming academic endeavors but also ensuring that I have a competitive edge when I step into the professional world.

Framework based Unit Testing – Catch2

Catch2 is rapidly gaining popularity among businesses and developers as a preferred C++ testing framework. Catch2 is known for its simplicity and header-only approach, which reduces setup hassles. Users can enjoy features such as natural expression assertions, test case organization using sections, and readable test output. Additionally, the framework allows results to be exported in multiple formats, including XML and JSON. One of Catch2's unique features is its capacity to construct tests with less unnecessary code, improving the testing procedure. However, if someone is transitioning from another testing tool to Catch2, it might take some time to adjust to its unique features. Even with this learning phase, many find Catch2's simplicity and effectiveness a welcome change from more intricate tools like Google Test.

"Catch2: Clion," CLion Help, https://www.jetbrains.com/help/clion/catch-tests-support.html#get-started (accessed Sep. 12, 2023).

Framework based Unit Testing – Google Test

Google Test is a widely-used choice among businesses for C++ testing. It's paired with Google Mock, forming a powerful combination for testing purposes. Together, they enable features such as detailed test writing, test grouping, and delivering concise feedback on test outcomes. There's even an option to obtain test results in XML format. Google Mock is especially useful for creating mock testing objects. However, to benefit from these tools, they both need to be integrated into our project. Despite their advantages, the integration process can be challenging for novices due to its intricate setup, dependency management issues, a notable learning curve, and possible performance overheads in extensive projects

"Google Test user's guide," Google Test, http://google.github.io/googletest/ (accessed Sep. 12, 2023).

Chosen Unit Test Framework and Rationale for Selection (Catch2)

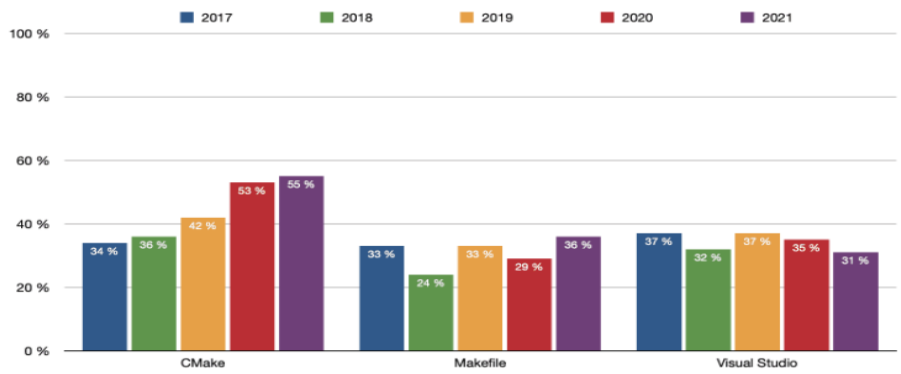
My decision was influenced by several factors. I prioritized simplicity and ease of setup for my projects and my learning path for C++. Catch2's header-only approach significantly reduces the setup hassles, allow testing without much preliminary work. The natural expression assertions, the organization of test cases using sections, and the readable test output were appealing features for me. I also appreciate the ability to export results in various formats such as XML and JSON. I value the simplicity and effectiveness it offers. I find this a welcome change from more complex tools, even though they might offer some advanced features. In conclusion, the combination of Catch2's ease of use, minimal setup, and clear testing structure made it the right choice for my C++ testing needs.

C++ Development

Big names in the computer industry such as Microsoft, Intel, the Free Software Foundation, and others have their modern C++ compilers. Companies such as Microsoft, The QT Company, JetBrains, and Embarcadero provide integrated development environments for writing code in modern C++. Popular libraries are available for C++ across a wide range of computer disciplines including Artificial Intelligence, Machine Learning, Robotics, Math, Scientific Computing, Audio Processing, and Image Processing.

GCC is a free compiler mostly used for Unix systems but can work on Windows as well. It's known for supporting the latest C++ updates. *MSVC (Microsoft Visual C++)* is created by Microsoft, used mainly in its Visual Studio which is works best for Windows. *Clang* is another compiler that sticks closely to C++ rules, though it doesn't support as many systems as *GCC*. Lastly the *Intel C++ Compiler* is designed to work best on computers with Intel chips.

Make is a Build tool that often used in Unix systems primarily facilitating the transformation of source code into executable programs and libraries. *CMak*e build tool manages building programs on various systems like Windows, macOS and Linux. It's a cross-platform build tool that generates build files suitable for a wide range of compilers and environments. *Ninja* builds programs faster and paired with *CMake.* Additionally, *MSBuild* is Microsoft's tool that building and setting up programs which is mostly used for Windows. For handling and installing libraries, library management tools like *Conan, vcpkg,* and *Buckaroo* are available.



IDE Platform Features, Supported Operating Systems, and Tool Integration

In C++, there are numerous editors and integrated development environments (IDEs) available for programmers. These tools offer a range of functionalities to aid in the coding process. To start with, many developers use basic text editors when writing C++ code. Popular choices in this category include Sublime Text, Atom, Visual Studio Code, vi/vim, and Emacs. These text editors are versatile and can be used for multiple programming languages, not just C++. On the other hand, there are IDEs specifically tailored for modern C++ development. Examples of these are CLion, Qt Creator, and C++Builder. These platforms provide a more comprehensive set of tools and features designed to streamline the development process for C++ projects

C/C++ [edit]

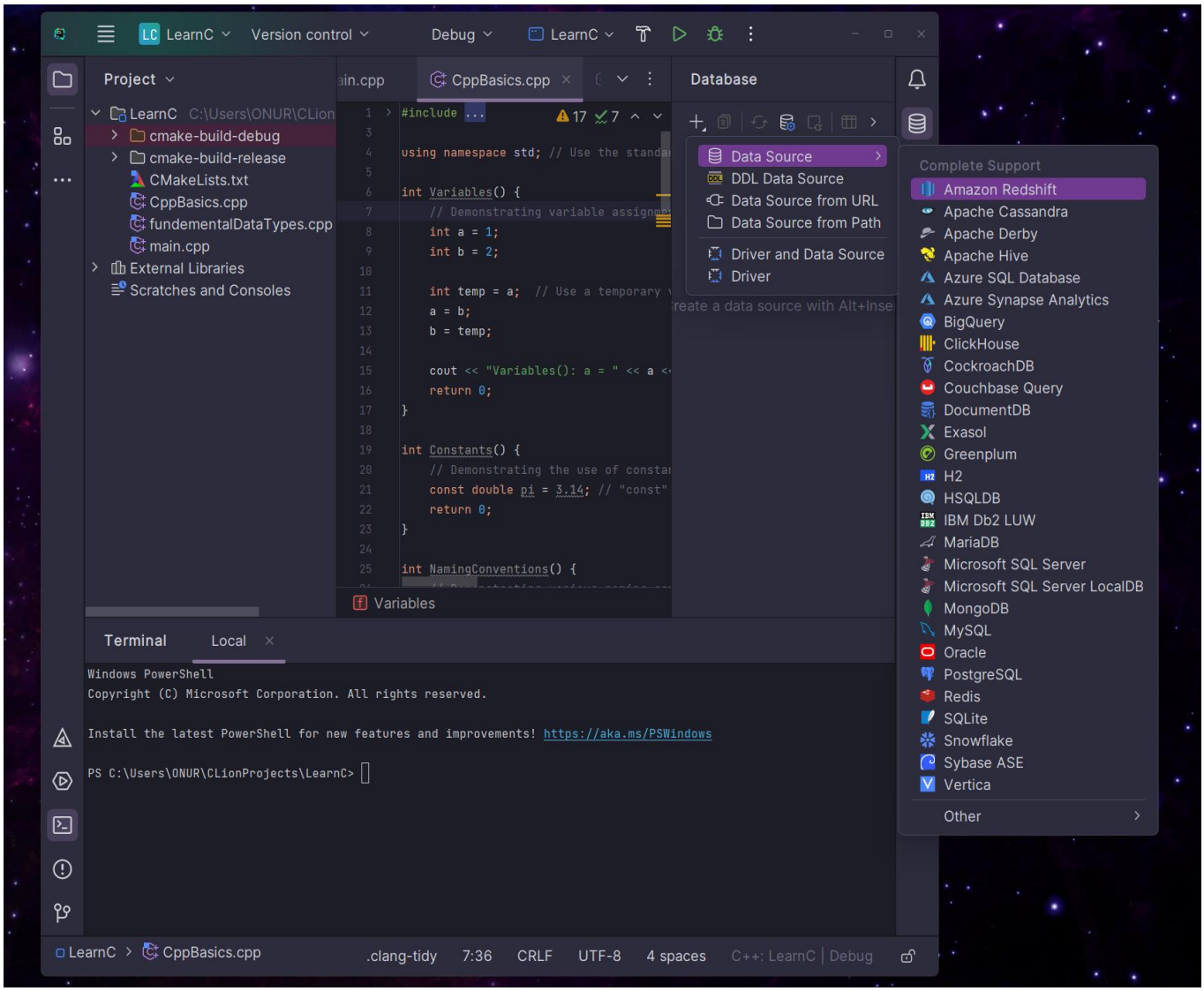
See also: C (programming language) and C++

IDE	License	Windows	Linux	macOS	Other platforms	Written in	Debugger	GUI builder	Integrated toolchain	Profiler	Code coverage	Autocomplete	Static code analysis	GUI-based design	Class browser	Latest stable release	C compiler	C++ compiler	Refactoring
Anjuta	GPL	No	Yes	No	FreeBSD	C	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	2016-03	Yes	Yes	No
AppCode (IntelliJ IDEA)	Proprietary	No	No	Yes		Java	Yes	Yes	No	Yes (Xcode profiler)	No	Yes	Yes	Yes	Yes	2012-12	Yes (Xcode toolchain)	Yes (Xcode toolchain)	Yes
C++Builder	Proprietary, Freeware (Starter edition only)	Yes	No (Cross compiler planned)	Yes (Cross compiler)	cross-compiles for Android and iOS	C++ and Object Pascal	Yes	Yes	Yes	Yes (AQTime Standard in package manager)	Yes	Yes	Yes	Yes	Yes	2017-03 Tokyo 10.2	Yes	Yes	Yes
Code::Blocks	GPL	Yes	Yes	Yes	FreeBSD, OpenBSD, Solaris	C++	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes[6]	Yes	2022-12[7]	Yes (MinGW + custom)	Yes (MinGW + custom)	Yes
CodeLite	GPL	Yes	Yes	Yes	FreeBSD	C++	Yes	Yes	Yes	Yes (As of CodeLite 6.1, integration with Valgrind)	No	Yes	Yes	Yes[8]	Yes	2023-01-14	Yes (GCC, Clang, VC + custom)	Yes (GCC, Clang, VC + custom)	Yes
Dev-C++	GPL	Yes	No[9]	No	FreeBSD	Object Pascal	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	2021-01-30	Yes	Yes	No
Eclipse CDT	EPL	Yes	Yes	Yes	FreeBSD, JVM, Solaris	C++, Java	Yes	Yes[2]	Yes[10]	Yes[11]	Yes[12]	Yes	Yes	Yes	Yes	2020-06[13][14][15]	External	External	Yes
Geany	GPL	Yes	Yes	Yes	FreeBSD, AIX, OpenBSD, Solaris, other Unix	C	Yes (via a plug-in)	No	No	No	No	Yes	No	No	Yes	2019-04[16]	External	External	No
GNAT Programming Studio	GPL	Yes	Yes	Yes	DragonFly BSD, FreeBSD, NetBSD, OpenBSD, Solaris	Ada	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	2016-08	Yes	Yes	Yes
JetBrains CLion	Proprietary	Yes	Yes	Yes		Java	Yes	No	Yes	No	No	Yes	Yes	No	Yes	2019-07[17]	Yes (customizable)	Yes (customizable)	Yes
KDevelop	GPL	Yes	Yes	Yes	FreeBSD, Solaris	C/C++	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	2022-12-08	External	External	Yes
LabWindows/CVI	Proprietary	Yes	No	No	cross-compile to Linux, Phar Lap ETS	?	Yes	Yes	Yes	Yes	No	Yes	No	Yes	—	2016-12	Yes	No	No
Microsoft Visual Studio	Proprietary, Freeware (Community edition only)	Yes	Yes (Cross compiler)[18]	No	Mac OS 7 (v2.x-v4.x only)	C++ and C#	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	2019-04	Yes	Yes	Yes (also plugin)[19]
Microsoft Visual Studio Code	MIT	Yes	Yes	Yes		TypeScript JavaScript CSS	Yes	No	Yes	No	No	Yes	No	Yes	Yes	2023-09-07	External	External	Requires language server support[20][21]
MonoDevelop	LGPL	Yes	Yes	Yes	FreeBSD, OpenBSD, Solaris	C#	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	2016-11	Yes (GCC + custom)	Yes (GCC + custom)	Yes
NetBeans C/C++ pack	Apache License	Yes	Yes	Yes	OpenBSD, Solaris	Java	Yes[22]	Yes[22]	Yes[23]	No[22]	No	Yes	No	Yes	Yes	19[24] (1 September 2023) [±]	External	External	Yes
OpenWatcom	OSI Approved	Yes (32-bit only)	partial	No	FreeBSD, DOS, OS/2	C/C++	Yes (GUI remote)	Yes	Yes	Yes	No	No	No	Yes	Yes	2010-06	Yes	Yes	No
Oracle Solaris Studio (formerly Sun Studio)	Proprietary, Freeware	No	Yes	No	Solaris	?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	2008-11	Yes	Yes	Yes
Pelles C IDE	Proprietary, Freeware	Yes	No	No		?	Yes	No	Yes	?	?	?	?	?	?	Aug 8, 2021	Yes	Yes	?
Qt Creator	GPL / LGPL / Proprietary	Yes	Yes	Yes	FreeBSD, Maemo, OpenBSD, Symbian	C++	Yes	Yes	Yes	Yes	No	Yes	Yes (clang)	Yes	Yes	2019-06	External	External	Yes[25]
Rational Software Architect (Eclipse IBM)	Proprietary	Yes	Yes	No	FreeBSD, JVM, Solaris	Java	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	2015-09	External	External	Yes

"Comparison of Integrated Development Environments," Wikipedia, https://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments (accessed Sep. 12, 2023).

Chosen IDE, Licensing, Restrictions, and Compliance (CLion)

JetBrains' CLion stands out as a top-tier Integrated Development Environment (IDE) for programmers working in the C and C++ programming languages. Along with its effective code aid tools, CLion also stands out for its thorough debugging capabilities and extensive code analysis features. CLion offers a variety of choices designed to meet various demands. The "Individual License" may be appropriate for some people, including solo professionals, open-source developers, and students. Larger businesses and corporations, on the other hand, might want the "Commercial License." Every CLion user must continue to be careful when using the program. These limitations, which are carefully outlined in the licensing agreement, are binding regulations rather than just suggestions. Adherence involves staying away from potential legal problems in addition to using software ethically. As a result, ensuring compliance becomes crucial for any developer or company using the platform.



"Learn CLion," JetBrains, https://www.jetbrains.com/clion/learn/ (accessed Sep. 12, 2023).

Advancing with CLion 2023.2 & Catch2 v3.4.0

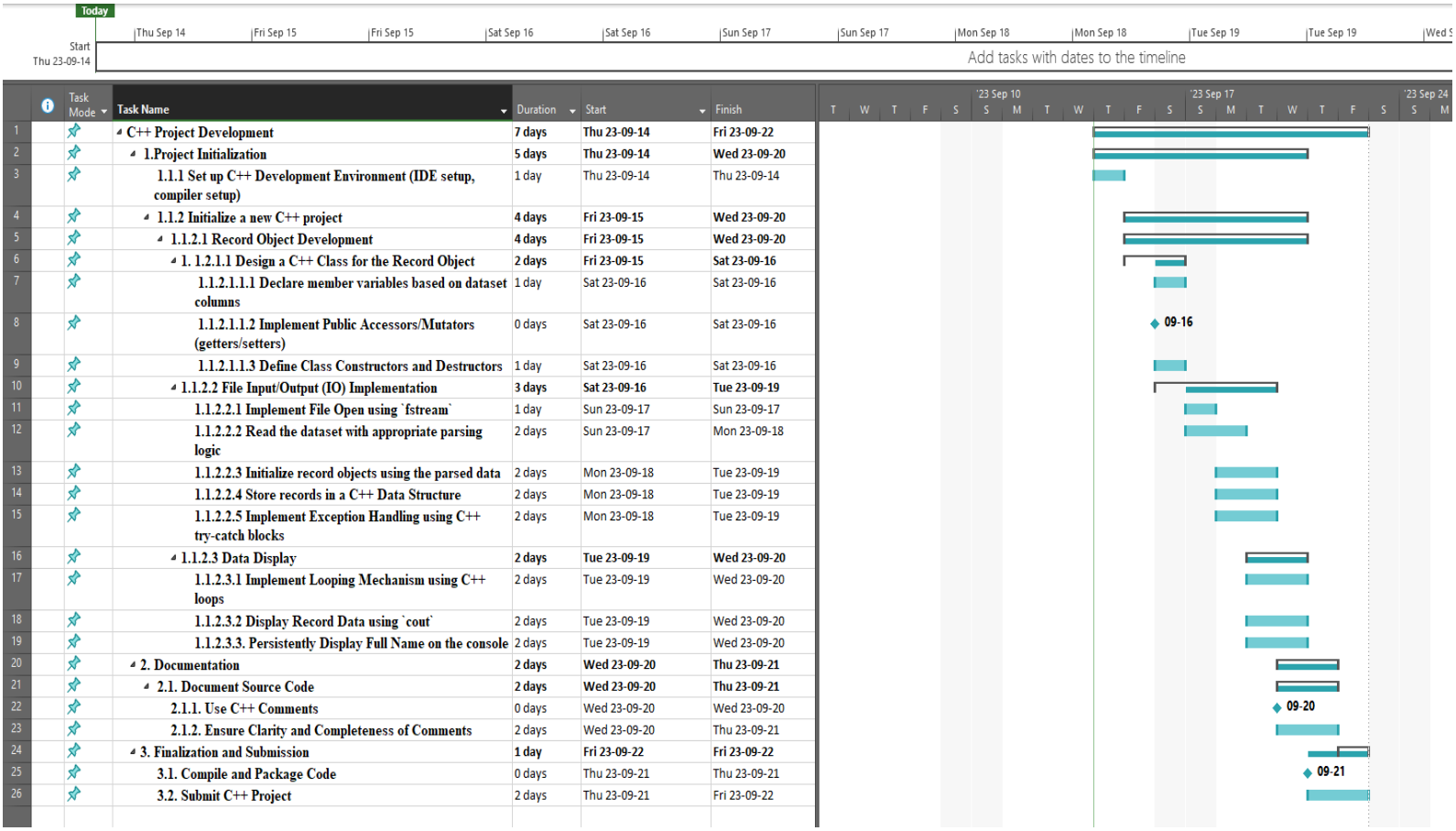
I've decided to embark on a journey with *C++20*. My main tool for this will be the *CLion 2023.2* software, known for its easiness and efficiency. To test my programs, I'll integrate *Catch 2 v3.4.0,* Unit testing framework known to work well with IDE CLion. My choice for Catch 2 is due to its compatibility and reputation in the coding community. In the long run, as I dive deeper into C++, I plan to familiarize myself with CMake. These are essential tools for any programmer looking to manage more significant C++ projects smoothly. My overarching goal is to become proficient in all areas of C++ programming.

"C++20," cppreference.com, https://en.cppreference.com/w/cpp/20 (accessed Sep. 12, 2023).

"Learn CLion," JetBrains, https://www.jetbrains.com/clion/learn/ (accessed Sep. 12, 2023).

Catchorg, "Releases · Catchorg/Catch2," GitHub, https://github.com/catchorg/catch2/releases (accessed Sep. 12, 2023).

WBS and Gantt Chart for Practical Project



	Task Mode	Task Name	Duration	Start	Finish
1		**C++ Project Development**	**7 days**	**Thu 23-09-14**	**Fri 23-09-22**
2		**1.Project Initialization**	**5 days**	**Thu 23-09-14**	**Wed 23-09-20**
3		**1.1.1 Set up C++ Development Environment (IDE setup, compiler setup)**	1 day	Thu 23-09-14	Thu 23-09-14
4		**1.1.2 Initialize a new C++ project**	**4 days**	**Fri 23-09-15**	**Wed 23-09-20**
5		**1.1.2.1 Record Object Development**	**4 days**	**Fri 23-09-15**	**Wed 23-09-20**
6		**1. 1.2.1.1 Design a C++ Class for the Record Object**	**2 days**	**Fri 23-09-15**	**Sat 23-09-16**
7		**1.1.2.1.1.1 Declare member variables based on dataset columns**	1 day	Sat 23-09-16	Sat 23-09-16
8		**1.1.2.1.1.2 Implement Public Accessors/Mutators (getters/setters)**	0 days	Sat 23-09-16	Sat 23-09-16
9		**1.1.2.1.1.3 Define Class Constructors and Destructors**	1 day	Sat 23-09-16	Sat 23-09-16
10		**1.1.2.2 File Input/Output (IO) Implementation**	**3 days**	**Sat 23-09-16**	**Tue 23-09-19**
11		**1.1.2.2.1 Implement File Open using `fstream`**	1 day	Sun 23-09-17	Sun 23-09-17
12		**1.1.2.2.2 Read the dataset with appropriate parsing logic**	2 days	Sun 23-09-17	Mon 23-09-18
13		**1.1.2.2.3 Initialize record objects using the parsed data**	2 days	Mon 23-09-18	Tue 23-09-19
14		**1.1.2.2.4 Store records in a C++ Data Structure**	2 days	Mon 23-09-18	Tue 23-09-19
15		**1.1.2.2.5 Implement Exception Handling using C++ try-catch blocks**	2 days	Mon 23-09-18	Tue 23-09-19
16		**1.1.2.3 Data Display**	**2 days**	**Tue 23-09-19**	**Wed 23-09-20**
17		**1.1.2.3.1 Implement Looping Mechanism using C++ loops**	2 days	Tue 23-09-19	Wed 23-09-20
18		**1.1.2.3.2 Display Record Data using `cout`**	2 days	Tue 23-09-19	Wed 23-09-20
19		**1.1.2.3.3. Persistently Display Full Name on the console**	2 days	Tue 23-09-19	Wed 23-09-20
20		**2. Documentation**	**2 days**	**Wed 23-09-20**	**Thu 23-09-21**
21		**2.1. Document Source Code**	**2 days**	**Wed 23-09-20**	**Thu 23-09-21**
22		**2.1.1. Use C++ Comments**	0 days	Wed 23-09-20	Wed 23-09-20
23		**2.1.2. Ensure Clarity and Completeness of Comments**	2 days	Wed 23-09-20	Thu 23-09-21
24		**3. Finalization and Submission**	**1 day**	**Fri 23-09-22**	**Fri 23-09-22**
25		**3.1. Compile and Package Code**	0 days	Thu 23-09-21	Thu 23-09-21
26		**3.2. Submit C++ Project**	2 days	Thu 23-09-21	Fri 23-09-22

The Gantt chart for the C++ Project spans 7 days, starting on Thursday, 23rd September and ending on Friday, 30th September. The project begins with a setup phase on the first day. During this phase, tools for C++ programming are set up, including the software to write and test the code. From days 2 to 5, a new C++ project is initiated with a focus on the 'Record Object'. The design of this object takes 2 days, during which decisions about what data it holds and how to access or change this data are made. Also, how this object is created and destroyed is defined. Then, for 3 days, file handling tasks are performed. Files are opened and read, data from these files are used to create 'Record Objects', and these objects are stored in a way C++ understands. Additionally, error checking is added using special C++ techniques. In the subsequent 2 days, data is displayed on the screen using C++ techniques. The full name is always shown. For the next 2 days, comments are added to the code to help others understand its purpose. On the last day, the code is compiled to ensure its functionality. Once verified, the project is prepared for submission.

C++ program to read and display a file's content, https://codescracker.com/cpp/program/cpp-program-read-and-display-file.htm (accessed Sep. 12, 2023).

References

1. "Tiobe index," TIOBE, https://www.tiobe.com/tiobe-index/ (accessed Sep. 11, 2023).
2. S. O. |@sogrady |May 16, S. O'Grady, and Name, "The Redmonk Programming Language Rankings: January 2023," tecosystems, https://redmonk.com/sogrady/2023/05/16/language-rankings-1-23/ (accessed Sep. 11, 2023).
3. "PYPL popularity of Programming Language index," index, https://pypl.github.io/PYPL.html (accessed Sep. 11, 2023).
4. "PYPL popularity of Programming Language index," index, https://pypl.github.io/PYPL.html (accessed Sep. 11, 2023).
5. "Catch2: Clion," CLion Help, https://www.jetbrains.com/help/clion/catch-tests-support.html#get-started (accessed Sep. 12, 2023).
6. "Google Test user's guide," Google Test, http://google.github.io/googletest/ (accessed Sep. 12, 2023).
7. "Comparison of Integrated Development Environments," Wikipedia, https://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments (accessed Sep. 12, 2023).
8. "Learn CLion," JetBrains, https://www.jetbrains.com/clion/learn/ (accessed Sep. 12, 2023).
9. "C++20," cppreference.com, https://en.cppreference.com/w/cpp/20 (accessed Sep. 12, 2023).
10. Catchorg, "Releases · Catchorg/Catch2," GitHub, https://github.com/catchorg/catch2/releases (accessed Sep. 12, 2023).
11. C++ program to read and display a file's content, https://codescracker.com/cpp/program/cpp-program-read-and-display-file.htm (accessed Sep. 12, 2023).