مجلة كلية التربية – جامعة سرت

المجلد(1) العدد (2) يوليو 2022

C++ and Java: A Comparative Study of the Most Popular Object-Oriented Programming Languages in Libyan Universities

Faiza Abdulsalam Mohammed 1, Nawara Makhzoum Alhassan 2

1 Computer Science Department, Sirte University, Sirte, Libya

2 Computer Science Department, Higher Institute of Science and Technology, Sirte, Libya

Firstfaiza.18@su.edu.ly

ABSTRACT

Keywords: **High-level programming, OOP, C++, Java, inheritance, comparative study.**

The software has been experienced a large level of improvements by a lot of scientific researchers in this world. It can be considered as a set of some meaningful instructions written in a certain programming language. There are several programming languages which got comprehensive popularity like C, C++, C#, and Java. These languages are used to primarily evolve various applications. Object-Oriented Programming (OOP) languages, such as C++ and Java are one of the main three categories of programming languages. Most of the programmers always have a debate about choosing the best object-oriented programming language to help them develop their software applications. This review paper involves a comparison of C++ and Java with respect to some theoretical aspects to figure out the basic differences between them. Consequently, guide students to make a critical decision to choose the most applicable object-oriented programming language to develop their applications. The study has focused on both programming languages because they are taught in most of Libyan universities.

1- Introduction

The Software can be defined as a simple set of instructions that can be read by machine. It is written in any programming language [1]. The research on software technology and compiler optimization has been applied in an effort to make the execution of software applications faster than before. High-level programming languages were first designed in the 1950s [1]. Since then, the study of programming languages has been a very productive area. The high-level programming language is easier to use, makes the program development more understandable and simpler as respects to a lower-level language [2]. The high-level languages can be divided into three categories which are object- oriented, procedural and functional. Numerous high-level programming languages have been developed until now such as Fortran, COBOL, Eiffel, Lisp, Basic, C, C++, C#, Java and VB.NET. Several languages support various paradigms. Some are better appropriate to express algorithms while others are targeting the non-technical users [3]. Recently, object-oriented programming has appeared as the powerful computer programming methodology, and object-oriented languages such as C++ and Java have extensive use in academia [4]. Students seem to be unaware of choosing the most suitable programming language to build up the application software. This has some unfortunate consequences. Therefore, it is important to make a rational decision that which language should be selected. In order to select the proper one for the specific problem domain, one has to know what features it provides to support the requirements. The area of comparing programming languages experienced a lot of research to prefer one language from another. C++ and Java are two of the most popular OO programming languages in most of Libyan universities. They are the most commonly used programming languages as reported by TIOBE table [5]. Therefore, they have been the choice of our study to be compared. Finally, the purpose of this study is to facilitate and encourage students to explore and understand the best characteristics of the chosen languages.

2- OBJECT-ORIENTED PROGRAMMING

"Object-Oriented Programming (abbreviated to OOP) is approach to formulate programs as a series of objects of objects and methods that interact to perform a specific task." [6].In the last two decades, research on object-oriented methodologies has become an increasing field of interest due to the growing popularity of object-oriented programming [7]. OOP is used to depict a programming approach based on classes and objects. The OO paradigm allows programmers to build software as a collection of objects that contains both data and methods [8]. It works best in languages that are meant to be OOP like C++ and Java.

A programming language is said to be an object-oriented if it immediately supports the concepts such as instantiation, abstraction, code reuse, inheritance, encapsulation and polymorphism [9]. Additionally, techniques are object-oriented if they support the use of object-oriented programming. For example, a design method is object-oriented if its regular and proper use leads to programs that exploit abstraction, inheritance, and polymorphism where appropriate [10]. Table I. Despites some object-oriented languages and their development year. The languages that are compared in this study also use the OO paradigm.

Table I: Some of the Object-Oriented programming Languages with their year of development

No.	The language	Development year
1.	Smalltalk	1972
2.	Ada	1980
3.	**C++**	1983
4.	Objective C	1984
5.	Eiffel	1986
6.	Python	1991
7.	**Java**	1995
8.	Ruby	1995
9.	PHP	1995
10.	C#	2000

2.1 Top Benefits of Object-Oriented Programming (OOP)

- OOP provides the ability of code reusing from one program into another through objects [11].
- Through encapsulation, it becomes simpler to achieve consistency and data integrity by validating data and restricting user from directly accessing the data/data members [12].
- By using polymorphism, the same operator or same function can be used for different purposes. This helps the programmer to manage software complexity easily [12].
- OO languages are good for defining abstract data types.
- An object-oriented program is much easier to modify and maintain as compared to a non-object oriented program [13].
- Inheritance gives the ability to reuse the existing class to derive new classes such that using of existing class is extended and the redundant code is eliminated. This saves cost and time of program [11].
- Based on objects, OOP allows a designer to model complex and large systems into simpler and more manageable designs [14].
- OOP emphasizes the principle of data hiding that helps the programmer to create secure programs that cannot be conquered by codes in other parts of the program. So data can be made private to a class such that only member functions of the class can access the data [11].

2.2 BASIC OBJECT-ORIENTED CONCEPTS

- *Classes*: A class is the basic unit of object-oriented programming. Class and object are two concepts related to one another, however, each concept holds its own meaning [11]. A class is sometimes called the object's type. Generally, a class is defined by its attributes and methods. After defining a class, any number of objects can be created which belong to that class [15].

- *Abstraction* is one of the basic concepts of OOP, it is defined as the process of hiding the details and revealing only the fundamental features of a particular object [10]. Computer experts use abstraction to understand and find solution to the problems and communicate their solutions with the computer using a specific computer language.
- *Encapsulation* is defined as hiding the details of implementation within a class. Encapsulation allows programmer to focus on what something does without considering the complexities of how it works [11].
- *Inheritance* is the basic mechanism of object-oriented programming for defining new objects by reusing already existing object definitions [12]. There are two types of inheritance: single and multiple. Single inheritance is the ability of a class to inherit the features of a single super class with more than a single inheritance level i.e. the super class could also be a subclass inheriting from a third class and so on. On the other hand, multiple inheritance, is the ability of a class to inherit from more than a single class [15].
- *Polymorphism* "Poly means many and morph means 'form'. Thus, polymorphism is related to the idea of 'having many forms' or 'having many types' "[16]. In the literature, polymorphism is often associated with functions that can accept parameters of several types. This feature is called parametric polymorphism [17]. Polymorphism is broadly used in implementing Inheritance.

3- BRIEF HISTORY OF PROGRAMMING LANGUAGES: C++ AND JAVA

3.1 C++

Bjarne Stroustrup was developed C++ starting in 1979 at Bell Labs as an extension to the C programming language [18]. Previously to 1983, Stroustrup added some features to C and formed what he called 'C with Classes' [2]. The Simula's use of classes and object-oriented features with the efficiency of C language has been combined [5]. In 1983, the term C++ was first appeared [4]. C++ is affected by some languages like C, Ada, Simula, etc [18]. C++ is defined by an International Organization for Standardization (ISO). In 1998, the first ISO C++ standard was published which known as C++98 [18]. The second standard was ratified in 2011, so that version of C++ is known as C++11. The third ISO standard, C++14 was released in 2014 and the recent standard known as C++17[20] was presented in 2017 [21].

3.2 JAVA

James Gosling and his team from Sun Microsystems started to develop Java in 1991 [3]. Java was originally called OAK, but it had to change its original name because Oak had been used by another programming language [28], Sun changed the name to Java in 1995 and the new name Java was inspired by a coffee bean [8]. Then the language has been modified to take features of the flourish World Wide Web [7]. “In 1995, Netscape Incorporated released its latest version of the Netscape browser which was capable of running programs of Java” [10]. The latest release of Java is Java SE 9 [9]. Java language obtained much of its syntax from C and C++ languages but has fewer low-level facilities and a simpler object model [7].

4- COMPARISON OF C++ AND JAVA LANGUAGES

C++ is a general-purpose programming language [18] and has two main components: a direct map to hardware and zero-overhead abstractions [23]. C++ supports four programming styles; procedural programming, data abstraction, object-oriented programming and generic programming [22]. It offers classes which supports deterministic destruction. In addition, C++ design has been concentrated on fundamental notions such as memory, abstraction, mutability, resource management, error handling, expression of algorithms, and modularity [22]. C++ is a compiled language, and its implementations exist from some of the most modest microcomputers to the largest supercomputers. Consequently, for almost all operating systems [21].

Useful tools provided by the C++ Standard Template Library (STL) include containers as the collections of objects, algorithms that perform operations and iterators that provide array [24]. In addition, Boost [25] is a volunteer organization created to add useful library components such as network I/O based on the STL. A rich amount of third-party libraries exist for GUI and other functions [23]. The filename extensions of C++ are .cc .cpp .cxx .C .c++ .h .hh .hpp .hxx .h++ [19].C++ is successfully used in a lot of application areas such as writing system and application software, high performance server, client applications, and video games [9].

JAVA is considered a much simpler and easy to use object-oriented programming language when compared to the popular programming languages such as C++ [26]. The complexity of multiple inheritance in C++ has been replaced with a simple structure called interface in Java. However, it has strong influence of C/C++ [4]. Small Java applications are called Java applets and can be downloaded from a Web server and run on the computer by a Java-compatible Web browser, such as Microsoft Internet Explorer or Netscape Navigator [27]. Java is OO language because programming in Java is concentrated on creating objects, manipulating them, and making objects work together [26].

Java is considered an interpreted language. This feature makes it cross-platform: to run Java code, a Java Virtual Machine (JVM) is needed. Java applications are typically compiled to bytecode (class file) that can run on any machine that has a Java interpreter [26]. In Java, a precise relationship is enforced between class names and file names, e.g. source code for class Person has to be in Person.java [27]. Java is one of the first programming languages which consider security as part of its design. The interpreter, compiler, and Java-compatible browsers contain several levels of security measures that are designed to reduce the risk of security compromise, loss of data and program integrity [28].

Java has three different formats, Java2 Enterprise Edition (J2EE), Java2 Micro Edition (J2ME), and Java2 Standard Edition (J2SE). J2EE is convenient for the development of server programming, J2ME is suitable for embedded systems development for mobile phones and wireless application. And J2SE is suitable for the applications of desktop [7]. Java is mainly used to develop applications (web-based applications) and certainly in mobile applications. Teaching students programming in colleges as Java is a strong OOP’s language. Table II gives a comparison between C++ and Java according to some criteria.

The researches[1], [2], [18], [20], [21],[22], [26], [33], [34], [35], [37] have been summarized to write a comparison table between the two languages in a concise manner according to the criteria selected in the comparison, indicating differences and similarities between them.

Table II: Similarities and differences between C++ and Java

Criteria	C++	Java
Platform	Dependent	Independent
Global variables	Supports	Does not support
Access control	Private, public, protected, "friends".	Private, public, protected, package".
Pointers	Supports	Does not support
Compiler and Interpreter	C++ uses compiler	Java uses both
Operator overloading	operator overloading	No operator overloading
Garbage collection	Manual garbage collection	Automatic garbage collection
Paradigm	Multi-paradigm	Single paradigm.
Structure and union	Supports	Does not support

Criteria	C++	Java
Typing Strategy	Static typing only	Static and dynamic typing
Polymorphism	Needs declare virtual methods explicitly	Automatically supports
Inheritance	Multiple inheritance	Single inheritance
Function Overriding	supports	supports
Stand-alone data and functions	Yes	No
Concept of overloading	Supports	Supports
Interface	Interface is a class.	Interface is not a class.
Multithreading	C++11 offers threads	native multithreading support
UI prototype Design	Difficult to implement by default but supports some libraries.	UI prototype design

Much work and research have been carried out in the area of comparing programming languages to determine the best programming language for developing software. Comparing programming languages is a popular topic between software developers and programmers. These studies [26], [31], [36], [37], [38] were major influence of how this paper is written and the ideas followed to compare the said programming languages. [1] Compared C++ and Java from theoretical aspects and concluded that, C++ has pointers whereas Java does not. C++ is considered very fast while Java consumes a lot time and space. However, Java is open source and platform independent thus programmers can use it for free.

[3] Presented a survey and comprehensive research on ten programming languages: C++, C#, AspectJ, JavaScript, PHP, Haskell, Java, Scheme, BPEL, and Scala. The survey work involves a comparative study of these programming languages regarding the following criteria: web application development, web service composition, secure programming practices, OOP-based abstractions, reflection, aspect orientation, batch scripting, declarative programming, functional programming, and UI prototyping. These languages have been studied in the context of the mentioned criteria and the level of support they provide for each one of them. The researchers have found that C++ is not deemed a safe language. For example, C++ allows pointers to access memory. On the other, Java is considered a safe language.

In [15], the researchers used CK metrics to study the reusability and complexity of multiple inheritance as implemented in C++, Java, and Python. The analysis of results suggests that out of the three languages investigated C++ and Python offer better reusability of software when using multiple inheritance, whereas Java has major insufficiency when implementing multiple inheritance resulting in poor structure of objects. In [33] a comparative study between six programming languages: C++, C#, PHP, Java, Python, and VB was presented. These languages are compared according to characteristics of reusability, portability, reliability, readability, availability of compilers and tools, efficiency, familiarity and expressiveness. The study has disclosed that, C++ has exceeded all the languages under study in execution time and database connectivity. This explains the great power of the C++ language and its wide use in the software world. Java has revealed better suitability for web programming thanks to J2EE as well as desktop applications. Moreover, Java is a strongly typed language, which helps writing secure programs.

[36] Presented a comparative study of the languages C, C# and Java, with respect to the criteria: CPU usage and memory usage. The research provided a well-defined perspective of how each language performed based on the CPU utilization and memory consumption, of the languages C, C# and Java on algorithms bubble sort and linear search for data type (Integer). The overall best performers in case of memory usage and CPU usage are C and Java, undoubtedly with java being the first. In case of CPU usage in bubble sort and linear search java is the first, C is second and C# becomes the third. [37] Declared that it is really difficult to mention that one language is better and should be used while developing software. Each language has its advantages and disadvantages. So, developers will obtain stable and fast performance from the code written in a suitable programming language. [39] Compared and analyzed four selected programming languages i.e. C, C#, Fortran and Java based on particular runtime tests to understand the basic differences among the programming languages. The authors concluded that object-oriented programming languages like C# and Java perform better in the case of compilation and execution time as compared to procedure oriented programs such as C and FORTRAN.

[40] constructed a comparison between two main programming languages that are C++ and Java, the comparison operation includes the time required to perform some algorithm i.e. efficiency, the speed of operation, and flexibility to adjusting some code. The authors used the same code to compare between the two software to determine which one is better. It is

found that C++ needs less execution time compared with Java. However, Java requires about 10% surplus time to execute the same code comparing to C++ language.

[41] Performed a comparative research on C++, C#, and Java programs to investigate which language can help write better object-oriented programs. By analyzing 23 object-oriented metrics on the solutions to 78 real programming tasks. The statistical result reveals that the programs written in C++ and Java do not show a significant difference in class reusability, encapsulation and polymorphism. C++ classes are notably larger than Java classes in size, but their complexity does not differ remarkably, Java outperforms C++ in creating concise and cohesive classes. Java notably outperforms C++ in building deep inheritance trees. These empirical results could help practitioners choose a suitable language to develop OO systems. [42] Presented a comparative study of C, C++ and Java languages and their advantages and disadvantages.

5- Conclusions

The Continuous increase in complexity and variety of software, and with its rising importance, researchers should give more interest to decide which programming language is more suitable. Much research has been carried out in the field of comparing programming languages to determine the most suitable programming language for development of software. After comparing C++ and Java languages on different parameters, the research concluded that both the languages are different from each other and both find its usage in various areas of programming. In addition to that, there is no an ideal language to choose for all purposes, rather each language has its own distinguish characteristics depending upon the programmers requirements. If a highly secured system is needed Java can be preferred, and if the hardware implementation is the concern C++ is the preferred language.

Many programmers enjoy writing codes using C++ with more flexibility and with few time required to execute their code comparing to Java. Programmers who are interested in developing Android apps choose Java. As well, C++ is the choice of developers who want to develop games, process audio or video files. The main objective of this research paper is to raise the student's awareness when it comes to choosing the programming language as the tool to develop their applications. Each of the languages considered has its own specific characteristics that distinguish them from the others. In future, we would like to increase the comparative study by comparing the mentioned languages with another language like C#. Moreover, comparing the languages from a practical perspective according to their features and using object-oriented metrics.

Acknowledgements

We thank God Almighty, who enabled us to complete this scientific research, and who inspired us with health, wellness and determination. I would like to dedicate this work to my friend and research partner *Nawara Alhassan*, I ask God to forgive her sins, and have mercy on her.

References

[1] M. Singh, H.S. Saluja, P. Talwar, and A. Kaur (2015) Comparison of OO programming languages, IJARCSSE, Volume 5, Issue 8, pp. 666-672.

[2] Z. Parveen and N. Fatima (2016) Performance comparison of most common high level programming languages, IJCAR, ISSN 2305-9184, Volume 5, Number 5, pp.246-258.

[3] R. Naim, M. F. Nizam, S. Hanamasagar, J. Noureddine, and M. Miladinova, Comparative studies of 10 programming languages within 10 diverse Criteria, a team 10 COMP6411-S10 Term Report.

[4] B. K. Bruce (2002) Foundations of object-oriented languages: Types and semantics", Computer Science and Intelligent System, MIT Press.

[5] TIOBE programming community index for January 2022, available at http:// https://www.tiobe.com/tiobe-index/

[6] K.C. Chen (2004) Comparison of object-oriented and procedure-based computer languages: Case study of C++ programming, Issues in Information Systems, Volume V, No 1.

[7] L. F. Capretz (2003) A brief history of the object- oriented approach", Software Engineering Notes vol 28 no 2.

[8] F. U. Onu, M. O. Ezeji (2016) Comparative Study of Structured and OOP Paradigms, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 8.

[9] W.R. LaLonde, J. McGugan, and D. Thomas (1989) The real advantages of pure object-oriented systems or why object-oriented extensions to C are doomed to fail", ." In Computer Software and Applications Conference, 1989. COMPSAC 89., Proceedings of the 13th Annual International, pp. 344-350. IEEE.

[10] B. Stroustrup (1995) Why C++ is not just an object-oriented programming language, addendum to OOPSLA95 Proceedings, ACM OOPS Messenger.

[11] P. Wegner (1990) Concepts and paradigms of object-oriented programming, ACM SIGPLAN OOPS Messenger 1.1: 7-87.

[12] E. Kindler,I. Krivy (2011) Object-oriented simulation of systems with sophisticated control, International Journal of General Systems: 313–343.

[13] L.K. Ferrett, and J. Offutt (2002) An empirical comparison of modularity of procedural and object-oriented software, Thirteenth International Conference on Engineering of Complex Computer Software, Annapolis, MD.

[14] A. Gupta, S. Chempath, M.J Sanborn, , l.a. Clark, and R.Q Snurr (2003) Object-oriented programming paradigms for molecular modeling". Molecular Simulation, 29(1), 29-46.

[15] F. Albalooshi, and A. Mahmood (2017) A Comparative study on the effect of multiple Inheritance mechanism in Java, C++, and Python on Complexity and Reusability of Code, IJACSA, Vol. 8, No. 6.

[16] A. Shalloway, and J. R. Trott (2005) The Object-oriented paradigm", Volume 2, Issue 3.

[17] K. Nørmark (2010) Object-oriented Programming in C# for C and Java programmers, Aalborg University, Denmark.

[18] S. Bjarne (2014) Programming: principles and practice using C++. Pearson Education.

[19] S. Bjarne (1991-2006) Evolving a language in and for the real world: C++.

[20] S. Bjarne (2015) Stroustrup: Thoughts on C++17- An Interview, Interviewed by Sergio De Simone.

[21] S. Bjarne (2013) The C++ Programming Language. 4th ed , ISBN 978-0-321-56384-2. QA76.73.C153 S77.

[22] S. Bjarne (2015) Lecture: The essence of C++. University of Edinburgh.

[23] ISO/IEC Programming Languages– C++11 Draft" (n3797) §3.7 Storage duration [basic.stc].

[24] G.Lo. Russo (2015) An Interview with A. Stepanov. stlport.org.

[25] Boost . The Boost library collection. http://www.boost.org/

[26] V. Dwarampudi, S.S Dhillon, J. Shah, N.J Sebastian, and N. Kanigicharla (2010) Comparative study of the pros and cons of programming languages Java, Scala, C++, Haskell, VB. NET, AspectJ, Perl, Ruby, PHP & Scheme-a" Team 11 COMP6411-S10 Term Report. arXiv preprint arXiv:1008.3431.

[27] K. Arnold, J. Gosling, and D. Holmes (2005) The Java programming language, Addison Wesley Professional.

[28] V. Kleber, A. Schulter, C. Westphall, and C. Westphall (2010) Intrusion detection techniques in grid and cloud computing environment, IT Professional, IEEE Computer Society 12, no. 4: 38-43.

[29] H. Chen (2010) Comparative study of C, C++, C# and Java programming languages, university of applied sciences, Degree Program of Information Technology.

[30] A. Snyder, “Encapsulation and inheritance in object-oriented programming languages”.

[31] P. S. Gondchawar (2015) Comparison between Object Oriented Programming Languages: Java and C++, International Journal of Advance Research in Computer Science and Management Studies, Volume 3, Issue 3.

[32] G.Lo. Russo (2015) An Interview with A. Stepanov, stlport.org.

[33] Z. Alomari, O. El Halimi, K. Sivaprasad, C. Pandit (2015) Comparative Studies of Six Programming Languages.

[34] O. B. Oyenike (2012) Comparative analysis of some programming languages”, Transnational Journal of Science and Technology, vol. 2, No. 5.

[35] G. Jindal, P. Khurana, T. Goel (2013) Comparative study of C, objective C, C++ programming language, International Journal Of Engineering And Computer Science ISSN:2319-7242, Volume 2 Issue 1, Page No. 202-206.

[36] R. Sharma, A.J. Singh, Pankaj Gupta (2016) Analytical Comparison of the Programming Languages, International Advanced Research Journal in Science, Engineering and Technology, Vol. 3, Issue 11.

[37] Ruchi Sharma, A.J Singh (2016) Comparative Study of the Programming Languages, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 8.

[38] Mrs. Partibha, Mrs. Ashu Khokhar (2015) Comparative study of C, C++, C# and Java programming languages, International Journal of Enhanced Research in Management & Computer Applications, ISSN: 2319-7471, Vol. 4 Issue 6, pp: (7-12), Impact Factor: 1.296.

[39] K. Biswa, B. Jamatia, D. Choudhury, and P. Borah (2016) Comparative analysis of C, Fortran, C# and Java programming languages, Keshav Biswa et al, IJCSIT, Vol. 7 (2) , 1004-1007.

[40] A.M. Alnaser, O. AlHeyasat, A.A. Abu-Ein, H.M.Said, A.A. M. Sharadqeh (2012) Time comparing between Java and C++ software, Journal of Software Engineering and Applications, 5, pp 630-633.

[41] D. Wu, L. Chen, Y. Zhou, B. Xu (2015) A metrics-based comparative study on object-oriented programming languages, State Key Laboratory for Novel Software Technology at Nanjing University, Nanjing, China.

[42] P. Prajakta, A. Shweta, P. Priyanka, Prof. Ram Meghe (2018) C, C++ and JAVA: The Magic of Computer Programming Language, International Journal of Innovative Science and Research Technology, pp 151-153.