



Comparison of OO Programming Languages

¹Manpreet Singh*, ²Harmanpreet Singh Saluja, ³Priyanka Talwar, ⁴Asst. Prof Amandeep Kaur

^{1, 2, 4}CSE, Guru Tegh Bahadur Institute of Technology, Delhi, India

³CSE, Inderprastha Engineering College, U.P., India

Abstract— *With the software industry developing so rapidly, there is a vast number of increase in the count of people who want to enter software development industry by learning programming languages. But which language should be chosen?*

In our thesis, research was conducted on two of the most popular programming languages. These languages chosen to be our objects to be studied are C++ and Java. There are very few languages which become so popular among more than a few people that they find their place in education or research.

This research paper is further divided into four parts. First part explains us about the problem we face during selection of the language and the purpose of this research. Second part gives us the brief description of the chosen languages. Third part compares the chosen languages. Fourth part tells the final conclusion that which language is better and should be used while developing software.

Keywords— *Java, C++, Programming Languages, Object Oriented Programming, OO Languages*

I. INTRODUCTION

A. Introduction and Related Work

Modern life has been extended in many areas by Software. Software is nothing but a mere set of instructions that can be read by machine it is being designed for. Software is used for any action we do on any machine. Today, software industry has its usage in almost all the industries. Without software, neither can we write a line nor can't we hear any music in computer.

The power of the software has been experienced by everyone in this world. Software is just set of some meaningful instructions written in any programming language. There has been a rapid development in technology since the first generation of programming emerged. It was 1950s when the high level programming languages were first designed. Since then, the study of programming languages has been very productive area. Latest generation programming language (5th generation) are aimed to solve a problem without use of a programmer [1].

There are very rare programming languages which get worldwide popularity like C, C++ and Java and C#. These are used to mainly develop large level applications. There are also programming languages which are used by lower number of people because for their orientation to academics for research purpose such as Scala, OZ, or Haskell.

The main aim of computer scientists behind developing a new programming language is to combine eloquent power with simplicity and productivity [4]. With the enduring increase in the diversity and complexity of software, and with the increasing gravity of software it is important to make a rational decision that which language should be chosen [4]. In context to the TIOBE Programming Community Index which "gives us manifestation of the popularity of programming languages" [3].

Much research and work has been carried out in the field of comparing programming languages to determine the best programming language for development of software. In the paper titled "Programming Languages: A Comparative Study" [2] the composers compare the languages C++, Java, Perl and Lisp under various parameters. The parameters used for the comparison of programming languages are reliability, reusability, compilers availability, readability, efficiency, and portability. The paper also depicts memory consumption, programming effort and run time efficiency by running same program on every language. In the paper named "A Comparative Study between Computer Programming Languages for Developing Distributed Systems in Web Environment" [5], the authors discuss the efficiency of programming languages C++, ANSI C++, Java and C# for development of Distributed systems and web services. The parameters used for comparison are but not restricted to reliability, maintenance, simplicity, usage, platform independency and concurrency.

In the article named "A Comparative Study of Language Support for Generic Programming" [4] the authors throw limelight on comparison of generic programming done on various languages such as C++, Haskell, Eiffel, Generic C#, Stanford ML and Java. In the work, the authors propose generics extension for language Java. With the increase in number of language supporting generics, the authors explain that the absences of generics can cause serious difficulties for coders.

Other important papers worth a mention in the field of comparing various programming languages are [6, 7, and 8].

According to TIOBE table, this research chooses the two of the most popular languages as the objects for study. The languages chosen to be studied are C++ and Java.

B. Need for comparison of Programming Languages

There are many languages that find its use from general purpose to some special purpose languages which are used in some specific applications. Learning which programming language and use it for software development is the main difficulty of the people. With the increase in the numbers of developers in the software industry, it is really important for every developer to choose a language which being stable also is fast.

As stated by Herbert Meyer:

“No programming language is perfect. There is not even a single best language; there are only languages well suited or perhaps poorly suited for particular purposes.”[9]

He also added:

“A useful language needs arrays, pointers and a generic mechanism for building data structures.”[9]

The main problem of the carried thesis:

Among the chosen two languages that are C++ and Java, which language to be used while starting to build new software?

C. Significance of choosing a Programming Language

This paper is significant because software that has been coded in a wrong programming language would cause a lot of trouble at later time. And since the code cannot be re-used because the language has been changed, so time spent on development of old software is a wastage and with competition so high, one can't afford to lose time so it is better we choose the correct programming language in the beginning only.

This paper can help programmers and coders avoid and solve the above mentioned problem. After choosing the correct programming language, resources spent on developing software would be reduced significantly. Likewise the performance of the software can also be improved.

D. Limitations

In this paper, only two of the most popular programming languages C++ and Java are chosen as the main commodity to be introduced and are compared with each other from theoretical aspect .

II. BRIEF HISTORY OF PROGRAMMING LANGUAGES, C++ AND JAVA

In this part, we would define what a programming language is, what are programming paradigms and then we would head towards our chosen languages.

A. Brief introduction of programming languages

“A programming language is a formal constructed language designed to communicate instructions to a machine, particularly a computer. Programming languages can be used to create programs to control the behavior of a machine or to express algorithms.” [10]

There has been a rapid development in the field of programming which has lead to development of more than hundred programming languages. With the speed of the processor becoming faster and faster, powerful programming languages are being developed to meet the demand of the user. In this context, we would discuss the timeline of different programming languages.

Machine language Machine language was the first generation of programming language that appeared in early 1950's. Machine language composed of series of ones and zeroes. Binary, being difficult to interpret by human beings is very prone to errors. The main setback of machine language is that it is not platform independent, means we need to create different machine code for each machine [11].

Symbolic Languages After machine language, the humans developed symbolic language. This was the second generation of programming languages. As the name tells, instead of using binary, symbols were used in this language to give command to a machine. The main problem of platform independency was not solved completely but still this language was used at that time [11].

Problem-oriented Languages The main development of the programming language occurred during early 1960's till 1980's. This era was the era of third generation programming language. The languages developed during this time had compilers or interpreters. C, C++ and Java, all are the examples of problem-oriented and third generation language. In the third generation language, the problem of platform dependency was completely solved and the languages developed were platform independent [11].

Non-procedural languages The languages developed after the problem oriented languages were called Non-procedural languages. These were focused on problem solving. The main difference between fourth generation programming language and other languages is that this is main concerned about what needs to be done instead of how. The most common and well known languages of this generation are SQL, MYSQL [11].

Fifth-generation programming languages A fifth generation programming language (abbreviated as 5GL) is a programming language based on solving problems using constraints given to the program, rather than using an algorithm written by a programmer. By using 5GL, the computers have the ability to work out by themselves by using the information programmed in databases. The robots developed with artificial intelligence are nothing but machines using a 5GL [12].

B. Brief history of C++ Language

The history of C++ language is seen back from 1979. It was then when Bjarne Stroustrup, was busy on his Ph.D.thesis . The language he was working then was Simula, which was designed for simulations. Simula supported the concept of object oriented programming but was too slow for use, so he added classes to C and named it “C with Classes”. In 1983, he coined the name C++ for “C with Classes” adding some features to the language. The notable features he added are function overloading, virtual functions, keyword const and comments with double slashes. [13][14][15]

C. Brief history of Java Language

Green Team in June 1991, a small team of engineers at Sun, consisting of James, Mike and Patrick started the process of development of new language called “Greentalk” which was then used for small electronic appliances. Greentalk had file extension .gt. After that “Oak” was also used and later “Java” was finalized. The name Java was inspired by Java coffee [16].

Java Version History Many versions of Java have been released till date. Current stable version of Java is Java SE 8. The timeline of the release of the versions of java is as follows:

1. JDK Alpha and Beta (1995)
2. JDK 1.0 (23rd Jan, 1996)
3. JDK 1.1 (19th Feb, 1997)
4. J2SE 1.2 (8th Dec, 1998)
5. J2SE 1.3 (8th May, 2000)
6. J2SE 1.4 (6th Feb, 2002)
7. J2SE 5.0 (30th Sep, 2004)
8. Java SE 6 (11th Dec, 2006)
9. Java SE 7 (28th July, 2011)
10. Java SE 8 (18th March, 2014) [17]

D. Programming Paradigms

“A programming paradigm is a fundamental style of computer programming, serving as a way of building the structure and elements of computer programs.” [18] A programming paradigm is nothing but just a combination of thoughts for programming of computers. It is that example that would serve as a pattern or a model so that other programs could be generated using the same logic and technique.

Imperative Paradigm This paradigm follows the phrase “first do this, next do that”. In this type of programming, the order of commands given is very important. Commands offered by imperative languages are Assignment, IO, procedure calls. The languages which represent imperative programming paradigm are FORTRAN, C, Basic and Pascal [19].

Functional Paradigm This paradigm respects a much simpler and clean programming technique than a imperative paradigm. The idea behind the functional programming is the theory of functions and the mathematics involved in it. In this case, it is next to impossible to alter any part of composite value. We can also say that the value produced by this paradigm is non-mutable. Some of the functional programming languages are Hope, Rex, Common Lisp, Scheme, Closure and many more [19].

Logic Paradigm This paradigm is different of all other main programming paradigms. This is applied in those problems where an answer can be obtained from the basic facts and relations. The idea behind this programming paradigm lies in the “automatic proof within artificial intelligence” [19]. An important concept is dividing program into two components: logic and control.

Object Oriented Paradigm “Object-oriented programming (OOP) is a programming paradigm based on the concept of “objects”, which are data structures that contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods” [20]. This programming methodology has gained the maximum popularity in the past decade. The basic idea behind this paradigm is to transmit data between objects and relate it with real world phenomenon. This is the most important of all the paradigms and used widely. The languages which we are comparing also use the OOP paradigm [19].

III. COMPARISON OF C++ AND JAVA FROM THEORETICAL ASPECTS

All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified.

A. Data Types and Sizes

As per definition given in Wikipedia, data type is nothing but identification of one of many categories of data, which are integer, real number, character, or Boolean, which determines the possible value for the selected type [21]. The type of the data decides what type of operation can be performed on the given data.

In the programming languages studied under this project, the variables must be declared before executing them in any function. They can be declared at the starting of the function after opening the function parenthesis.

C++ In C++, there are five basic data types. Table1 below summarizes the basic data types

Table 1 Basic Data Types

int	Datum of integral data type is referred as int or integer.
float	Numbers having small precision of decimal point are defined under float.

double	Numbers having larger precision of decimal point are defined under double.
char	Information is stored by a single alphabet or a single digit numeral or punctuation mark is defined under char data type.
boolean	The variables having value either True or False are stored under Boolean data type. It is mainly used to express logical result.

Table below summarizes the basic data types and qualifiers in C++.

Table 2 Basic Data Types And Quantifiers

Type	Size	Values
bool	1 byte	true or false
unsigned short int	2 bytes	0 to 65,535
short int	2 bytes	-32,768 to 32,767
unsigned long int	4 bytes	0 to 4,294,967,295
long int	4 bytes	-2,147,483,648 to 2,147,483,647
int (16 bit)	2 bytes	-32,768 to 32,767
int (32 bit)	4 bytes	-2,147,483,648 to 2,147,483,647
unsigned int (16 bit)	2 bytes	0 to 65,535
unsigned int (32 bit)	4 bytes	0 to 4,294,967,295
char	1 byte	256 character values
float	4 bytes	1.2e-38 to 3.4e38
double	8 bytes	2.2e-308 to 1.8e308

There are many keywords in C++, of which main keywords are listed down in Table below.

Table 3 Basic Keywords

Do	Int	Struct	double
Default	If	Static	while
Auto	Else	Long	switch
Continue	Goto	Sizeof	volatile
Case	Extern	Return	union
Char	Float	Short	unsigned
Const	For	Signed	void
Break	Enum	Register	typedef
Asm	dynamic_cast	namespace	This
reinterpret_cast	Private	Friend	Class
const_cast	Bool	Explicit	mutable
Throw	New	True	virtual
Public	Protected	static_cast	operator
Inline	Delete	Catch	template

Java There is a difference in integer data type in C++ and Java. In Java, integer values can be represented by five data types. So, there are a total of eight primitive data types in Java. Table 4 describes the primitive data types in Java.

Table 4 Primitive Data Types In Java

Data Type	Size	Description	Default Value
boolean	1-bit	true or false	False
char	16-bit	Unicode Character	\u0000
Byte	8-bit	Signed Integer	(byte) 0
Short	16-bit	Signed Integer	(short) 0
Int	32-bit	Signed Integer	0
Long	64-bit	Signed Integer	0L
Float	32-bit	Real number	0.0f
double	64-bit	Real number	0.0d

B. Struct

“A struct in the C programming language (and many derivatives) is a complex data type declaration that defines a physically grouped list of variables to be placed under one name in a block of memory, allowing the different variables to

be accessed via a single pointer, or the struct declared name which returns the same address” [25]. There can be any number of simple and complex data types in the struct.

C++ In C++, structure can contain both function and data and we can identify a variable just by using the name of Struct while in C, we need to use the “struct” keyword to identify a variable. For creation of lightweight objects, struct are perfect because they are more efficient than classes. Class members are by private by default whereas struct members are public by default.[25]

Java Java does not support Struct value

C. String type

C++ In C++, Strings can be classified under two heads, C++-style string and C-style string.

C++-style string is a data type of class. We can create objects of C++-style string. By using the header file “string”, we can use a library functions available for string objects.

For using C-style string, we need to include another header file named “cstring”. These strings contain an array of characters and this array is terminated by a null character. There are two different libraries for C-style strings and C++-style strings [26].

Java In Java, String is treated as a reference type. Strings are treated as immutable objects in Java and a new string must be used if any change is required [27].

D. Inheritance

Inheritance is characteristic depicted by all the OOP languages. In simple words, it is a methodology for code-reuse. Invention of inheritance lies down behind at 1967 for Stimula. Inheritance gives rise to hierarchy. In inheritance, the derived class can access all of the variables and functions of the base class. It can be of many types:

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchal Inheritance
- Hybrid Inheritance

C++ C++ supports all of the above inheritance. C++ struct also support the inheritance. The format for writing the code for inheriting a class is given as follows:

```
class derived-class: access-modifier base-class
```

Java To keep it simple, Java does not allow multiple and hybrid inheritance. We can use multiple and hybrid inheritance through interfaces only [28]. We use ‘extends’ keyword for inheriting a class in Java

E. Reference and Value Types

Basic data types such as byte, short, long, int, char, float, boolean and double are known as primitive types. Reference types are arrays and classes. Examples of reference types are String, String[], Scanner , int[] [29].

C++ C++ does not support Reference and Value types. In C++, access to memory for user defines types and basic data type is same.

Java Classes in Java are stored as reference types whereas primitive types hold their value at the stack, provided they are embedded inside a reference type. Reference types also lie on the stack but they hold the address of an object which lies on the heap. They show similarity to pointers in C++. Value types can be passed directly into methods whereas reference types are passed by a reference.

F. Pointer

Pointer can be defined as a variable having the address of another variable. Another variable may hold a value or it can be null or empty.

C++ Pointers are used in C++ for optimization of the code and writing function parameters. They can also be used as Strings and arrays. C++ also supports pointer arithmetic.

Java For some security concerns, there is no provision for pointer and pointer arithmetic in Java..

G. Array

Array is a data structure which is used for storing a fixed size sequential stack of elements of same type. The array contains contiguous memory locations.

C++ In C++, sub-array of a multi-dimensional array can’t be jagged or they need to be of same dimension of the bigger array.

Java In Java, we can have jagged arrays, means that the contents of the multidimensional array are arrays that may hold references to other arrays depending on how much deep array is r how many level does it contains.

H. Compiler Technology

Programming languages can be implemented by two types Compilation and Interpretation. Compilation is when the source code is read and translated into a machine code and if any errors are found , then they all are reported at the same time whereas in Interpretation, each line of source code is read and then checked for error and then translated into ma-

chine code. In other words, we can say that Compilation can be referred to translating an English article into Hindi totally and then reading the Hindi article whereas Interpretation is translating every word of English to Hindi by using a dictionary.

C++ C++ is a statically compiled language. “A static compiler converts source code directly to machine code that can be directly executed on the target platform, and different hardware platforms require different compilers”[30].

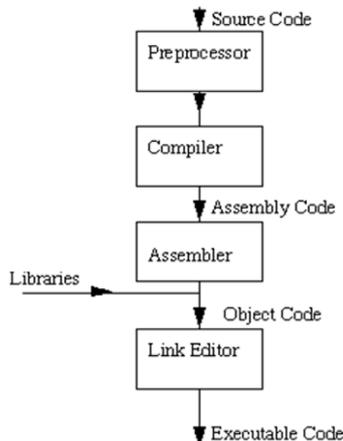


Fig. 1. C++ Compilation Model

The preprocessor is the first step for compilation of the code. The main functions of the preprocessor are Removing Comments and Interpreting special preprocessor directives which are denoted by # in the starting of the program. The compiler changes the output of the preprocessor to the assembly code. Object code is created by the assembler. Link Editor links the external variables and library functions with the object code to create a executable file [31].

Java, Java is a dynamically compiled language. The Java compiler converts Java source code into portable JVM (Java Virtual Machine) byte codes, which are ‘virtual machine instructions’ for the JVM.” [30]. There have been two generations of Java Virtual Machine. The first generation of Java Virtual Machine was comparatively slower than the second because the Java Virtual Machine interpreted the byte codes instead of compiling them to machine code and then executing them directly. This was not good approach because more time was consumed in executing the interpreter than the program. The later generation of Java Virtual Machine improved the performance a lot by using the Just in Time compilers. The Just in Time based virtual machines converted all the byte code into its machine code before the program has been executed. This approach allowed the program to lessen the time because no lengthy compilation phase is required before execution to begin [30].

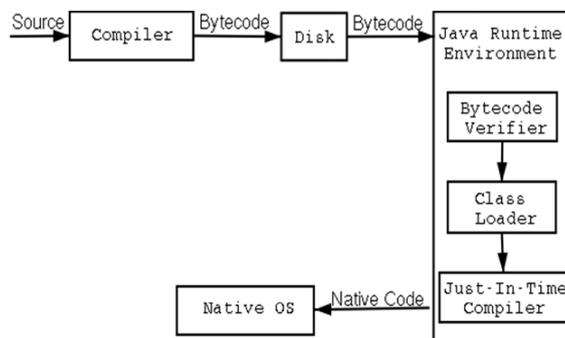


Fig. 2. Java Compilation Model

The process of compilation of a program in Java seems different from the process used in C++. Java can be termed as a semi-interpreted language. The reason being for this term is that” Java programs execute in JVM, which makes it an interpreted language” [30].

IV. CONCLUSIONS

After comparing both languages on various parameters we conclude that both the language are different from each other and both find its usage in different section of programming. C++ has pointers whereas Java does not. C++ being very fast can be used in software industry while Java consumes a lot time and space but Java is platform independent and open source thus people can use it for free. So , it is really hard to say which is better of the both.

REFERENCES

- [1] Wikipedia, Fifth-generation programming language, Available at http://en.wikipedia.org/wiki/Fifth-generation_programming_language
- [2] P. Kulkarni, H. D. Kailash, V. Shankar, S. Nagarajan, and D. L. Goutham. Programming Languages: A Comparative Study. Information Security Research Lab, NITK, Surathkal.

- [3] TIOBE Programming Community Index for April 2015, available at <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- [4] R. Garcia, J. Jarvi, A. Lumsdaine, J. Siek, and J. Willcock. A Comparative Study of Language Support for Generic Pro-gramming. Open Systems Lab, Indiana University Bloomington, IA. November 2003
- [5] K. Aldrawiesh, A. Al-Ajlan, Y. Al-Saawy and A. Bajahzar. A Comparative Study between Computer Program- ming Lan- guages for Developing Distributed Systems in Web Environment. Software Technology Research La- boratory (STRL) De Montfort University, Leicester, UK. November 2009
- [6] S. Vinoski. The Language Divide. Internet Computing, IEEE. Volume 10, Issue 2. March 2006
- [7] E. Specht, R. Redin, L. Carro, L. Lamb, E. Cota, and F. Wagner. Analysis of the Use of Declarative Languages for Enhanced Embedded System Software Development. Informatics Institute, Federal University of Rio Grande do Sul, Porto Alegre, Brazil. September 2007.
- [8] C. R. Roast, B. Khazaei, and J. I. Siddiqi. Formal Comparisons of Program Modification. School of Computing and Man- agement Sciences Sheffield Hallam University, Sheffield, UK. August 2002.
- [9] <http://www.iitk.ac.in/LDP/LDP/abs/html/why-shell.html>
- [10] Programming language, Available at https://en.wikipedia.org/wiki/Programming_language
- [11] The Evolution of Computer Programming Languages, Available at, http://www.associatedcontent.com/article/369862/the_evolution_of_computer_programming.html
- [12] Fifth generation programming language, Available at https://en.wikipedia.org/wiki/Fifth-generation_programming_language.
- [13] A Brief History of C++, available at <http://hubpages.com/hub/A-Brief-History-of-the-C-Language>
- [14] Online C++ tutorial, A Brief History of C++, available at http://www.intap.net/~drw/cpp/cpp01_03.htm
- [15] <http://www.cplusplus.com/info/history/>
- [16] <http://www.javatpoint.com/history-of-java>
- [17] <http://www.javatpoint.com/history-of-java#version>
- [18] https://en.wikipedia.org/wiki/Programming_paradigm
- [19] http://people.cs.aau.dk/~normark/prog3-03/html/notes/paradigms_themes-paradigm-overview-section.html
- [20] https://en.wikipedia.org/wiki/Object-oriented_programming
- [21] Data type, Available at https://en.wikipedia.org/wiki/Data_type
- [22] Jesse Liberty and Bradley Jones, “Working with Variables and Constants” in Sams Teach Yourself C++ in 21 Days, Fifth Edition, Sams Publishing, 800 East 96th Street, Indianapolis, Indiana, 46240 USA, 2005, pp 43-65
- [23] C++ Keywords, <http://en.cppreference.com/w/cpp/keyword>
- [24] Java Language Keywords, Available at http://java.sun.com/docs/books/tutorial/java/nutsandbolts/_keywords.html
- [25] Struct, Available at [https://en.wikipedia.org/wiki/Struct_\(C_programming_language\)](https://en.wikipedia.org/wiki/Struct_(C_programming_language))
- [26] Strings in C and C++, Available at http://cs.smu.ca/~porter/csc/ref/c_cpp_strings.html
- [27] String (computer science), Available at [https://en.wikipedia.org/wiki/String_\(computer_science\)](https://en.wikipedia.org/wiki/String_(computer_science))
- [28] <http://www.javatpoint.com/inheritance-in-java>
- [29] Primitive vs. Reference Data Types, Available at <http://pages.cs.wisc.edu/~hasti/cs302/examples/primitiveVsRef.html>
- [30] <http://www.ibm.com/developerworks/library/j-jtp12214/>
- [31] <http://www.cs.cf.ac.uk/Dave/C/node3.html#app:compiler>