

Performance Comparison of Most Common High Level Programming Languages

ZahidaParveen and Nazish Fatima

Department of Computer Science, University of Ha'il, Hail, Saudi Arabia

Abstract

A programming language that is very close to natural language and it's an abstraction from the hardware and enclosed firmware of the machine is known as high-level programming language. It may use natural language elements, be easier to use, making the process of developing a program simpler and more understandable relative to a lower-level language. The amount of abstraction provided defines how "high-level" a programming language is. The first high-level programming language designed for computers was Plankalkül, created by Konrad Zuse.

Many high level programming languages have been developed so far like Fortran, Lisp, COBOL, Basic, C, C++, Java, C# and VB.NET. Most studies in the comparison of programming languages emphasis on a particular property of languages. Our research work involves a comparative study of most common languages Java, C# and C++ with respect to the following criteria: syntax, Lines of Code, MachineDependency, Compilation Time, and ExecutionTime, Speed or efficiency and flexibility. We study these languages in the context of the above mentioned criteria and the level of support they provide for each one of them. The paper is intended to provide a comprehensive comparative knowledge about the above said languages. The use of a simple common program for all languages facilitates our comparison. We have coded the program in all of these languages, and we compare the syntax compile and run times, length of code, and memory management techniques. Ultimately, our goal is to encourage and facilitate programmers in understanding and exploring the best characteristics of the high level languages.

Keywords: High level programming languages, Comparative study of C#, JAVA and C++, Length of code, Compilation and execution time, High Performance Programming Language.

1. INTRODUCTION

To cope up with various development requirements a big variety of programming languages have been introduced in all the times. Broadly programming languages are divided into two categories Low level programming languages and high level programming languages. Our research focuses on performance of some of the high level languages. Just the characteristics of the said languages will be focused. The High level languages can be divided into three categories object oriented, procedural and functional for the following research we have chosen four mostly commonly used high level object oriented languages. The following table provides a list of high level languages with their year of development.

TABLE 1: DIFFERENT HIGH PROGRAMMING LANGUAGES WITH THEIR DEVELOPMENT YEAR

Language	Year of Development	S.No	Language	Year of Development
Fortran	1954	9	C	1971
Lisp	1958	10	Ada	1979
COBOL	1959	11	C++	1983
Simula	1964	12	Perl	1987
Basic	1964	13	Python	1991
Smalltalk	1969	14	Java	1995
Prolog	1970	15	C#	2000
Pascal	1970	16	VB.NET	2001

The object-oriented design concept has been popular for quite some time owing its success to the powerful features it offers for making program development easy and robust. Object oriented languages like java and C# offer an instinctive way of developing programs and provide prevailing characteristics for supporting the program development [1]. The previous studies about the comparison of programming languages concern about the their object oriented characteristics or the comparison of procedural and object oriented properties mostly. We have selected the most recent and commonly used languages i.e. C#, Java and C++. The selection of these languages is also inspired by the fact that java is taught in university of Hail under three compulsory courses (ICS-102, ICS201 and ICS-202) and also the students prefer to use the above said languages in development of their projects, so this research will also help them in understanding the characteristics of these languages and help them to choose the better one depending on the projects. The criteria chosen for comparing these languages syntax , Lines of Code, Machine Dependency, Compilation Time, Execution Time, Speed or efficiency and flexibility cover most of the properties of any language. We choose a common program to be executed in the platform of all these languages.

Based on our chosen program, we discuss how each language supports these concepts. We also compare the run-time efficiency and compilation time of the language implementation. The measurements are intended to provide the reader with an understanding of the estimated performance of the any high level language. Such a perceptive is useful because often performance considerations constrain what languages a programmer is likely to consider using on a particular project.

2. Properties of Programming Languages in Selected Criteria

1. **Java** .The very first name that was giving to java was OAK and it was originally developed at Sun Microsystems .The aim of OAK was to create a programming language that would be act like a bridge for communication for home entertainment appliances like VCR and TV. With the expansion of global networking (internet) the need of browsers become a cause for the emergence of Java from Oak with a specific web browser named Hot Java. [2]
- a. **Syntax of Java** .The set of rules that define how a program is written and interpreted is called syntax. The syntax of Java language is not a basic one it is derived from C and C++ with the difference that java is almost absolutely object oriented language. Except primitive types all the functions and variables directly belongs to classes and all the values are objects. Syntax of java language is kept simple by omitting the complex data types like operator overloading and unsigned integer; however its syntax is being constantly improved in every major JDK release. The following program shows the basic syntax of java.

```

publicclasssyntaxIntro{

/* Thisjava program point out basic syntax of language
 * This will print on console
 */
publicstaticvoid main(String[]args){
System.out.println("The syntax is very simple");// prints
}
}

```

In our research we have used a common program to check the simplicity and nearness of the syntax to natural language, java syntax is also evaluated in comparison with other languages.

- b. Java Machine Dependency:** Java code mostly consider as independent of machine or platform. The program is first compiled by JVM (Java Virtual Machine) in to an intermediately language that is java byte code. This byte code is then converted to machine language again by JVM. Thus apparently java does not depend on the platform rather its code depend on the underlying JVM. Many of the previous researches claimed that JVM is dependent on the underlying operating system; different operating systems need different JVM models.[3]
- c. Speed and efficiency:** Regarding speed java is a little slower than other languages. The programs executed in java are much slower as compare to other languages. The reason of the slow speed is that java does not directly compile the program to machine code until run time, this because java can be distinguish as a portable language.
- d. High Performance:** *Performance* at all times a concern regarding the programming languages. The Java platform uses a special scheme to have its highest performance that is interpreter run at full speed without taking in account the run-time environment. To achieve high performance java make it sure that memory is available on demand that is achieved by running *automatic garbage collector* at a low-priority background thread.
- e. Java a secure Language:** A Java language comes under the leading programming languages when there is a security concern. The most reliable language that is considered is java. Java provides the programmers highly secure environment at two levels compile time and at run time. The high level security in java is achieved by the Java Virtual Machine (JVM), that act like a security manager to remove the un-trusted code from the underlying operating system, and programmer and developers can use a complete security suite APIs to achieve high security.

2. C# (C sharp)

The original name that was given C# was Cool (C-like Object Oriented Language) by Anders Hejlsberg in 1999 but it was not finalized because of some trademark reasons. Later on in July 2000 Professional Developers Conference C# name was officially announced. Most of the main concepts C# borrows from java yet it is modern, simple and general-purpose object oriented language. The .NET framework is one of the essential parts of C#, the programs written in C# required .NET framework installed on that machine for execution. Like java, C# also doesn't have global elements, even the types are also wrapped in classes (objects).[5]

C# Syntax: Syntax of C# language is quite simple and easy to understand and have a lot of similarities with java program structure; most of the properties of syntax are inherited from C and C++. Being an object oriented language class is the main building block of it ,a C# program consists of the Namespace declaration, class, class methods, class attributes, statements and identifiers and comments.

The following program shows basic syntax of a C# program

```
// A simple program showing the basic syntax
using System;
namespace Syntax
{
class Testsyntax
{
static void Main()
{
Console.WriteLine("C# syntax is very simple!");
Console.ReadKey();
}
}
```

- a. **C# Machine or platform dependency:** C# is not fully independent from the underlying platform. The programs written in C# need .NET Framework to be executed. As .NET framework is a Microsoft product and it only support windows platform. Other operating systems such as Linux and Unix does not support it. However recently a 3rd party framework MONO has been introduced to run ASP.NET on Linux. A complete program compilation and execution process is shown in the Fig. 1

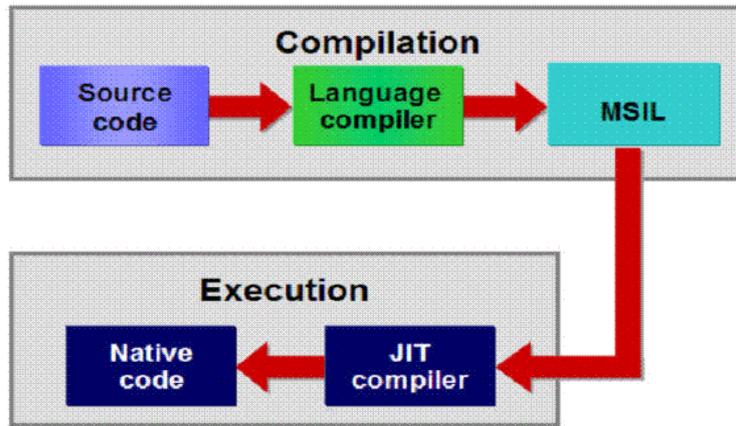


Fig. 1- Compilation and execution process of C# Program [5]

- b. **Speed and Efficiency:** All the .NET languages have slow speed when compared with other like Java and C++. The reason behind this is C# consuming high memory and the consumption cannot be controlled with ordinary methods. Another reason that cause the slow speed and deficiency in C# language is JIT low level optimization as the program optimization is done at runtime so cannot be take too much time. This factor can also affect the speed and efficiency to some extent.
- c. **C# Language Performance:** In spite of the fact that C# does not provides a very high performance yet it is very popular among developers for many of its characteristics. The performance of said language can be increased by following some of the common practices. Table-2 is showing the detail and effect of these practices using stopwatch function to calculate elapsed ticks.[7]

TABLE 2: EFFECT OF BEST PRACTICES ON PERFORMANCE OF C# CODE

Practice	Scenario-1	Elapsed Ticks	Scenario-2	Elapsed Ticks
Select your data type before using it	Using ArrayList<Object>	559	Using Value <Integer Array >	212
Use For loop instead of foreach	For Loop	494	Foreach Loop	737
Select structure instead of class when possible	Using Structure	44	using class	532
Prefer Stringbuilder over String for concatenation operations	Using String	463	Using String builder	24
best way to assign class data member	Using property	367	Direct assign	4

3. C++

C++ is an extension of C by Bjarne Stroustrup to integrate Simula structures into C to improve distributed computing in 1979 with first commercial implementation released in 1985. It was one of the most dominant languages of all time, C++ started as an extension to C. A subsequent release in 1989 added more advanced features dealing with extending classes, many function and variable types as well as error handling.

a. Syntax of C++

```
#include<iostream>
```

```
using namespace std;
// main() is where program execution begins.
int main()
{cout<<"Hello World";// prints Hello World
return 0;
}
```

The C++ language defines several headers, which contain information that is either necessary or useful to your program. For this program, the header `<iostream>` is needed. The line **using namespace std;** tells the compiler to use the std namespace. Namespaces are a relatively recent addition to C++.

The next line **// main () is where program execution begins.** is a single-line comment available in C++. Single-line comments begin with `//` and stop at the end of the line.

The line **int main ()** is the main function where program execution begins.

The next line **cout<< "This is my first C++ program.";** causes the message "This is my first C++ program" to be displayed on the screen.

The next line **returns 0;** terminates main () function and causes it to return the value 0 to the calling process. [8]

- b. Efficiency:** C and C++ are very fast and powerful languages. The primary reason is that they have already been compiled to machine code. The focus of these languages is building code that will run on a specific machine extremely efficiently. C/C++ assumes a good programmer, and focuses on the machine's efficiency.
- c. Ease of use:** Because of more efficiency, in these languages the programmer has a lot of low level controls of what is going on in the computer. The exchange is that the programmer must have considerable knowledge of the system with which they are working with. So, Low level control and overall processing speed decreases the ease of use for the programmer.
- d. Hardware implementation:** The language is general-purpose with a mix of high and low level features giving the programmer choice of implementation, even if the programmer may choose incorrectly. Still C++ is one of the prior choices for hardware implementation.
- e. Full object oriented support:** C++ integrated both object-oriented programming and procedural programming in its design. The original version of C++, called "C with Classes" was first used in 1980. C++ was developed to be a "better C" with support for techniques such as data abstraction and full object-oriented programming. The goal of the new language was to make programming more enjoyable.
- f. Platform independence:** Since C++ is so system dependent, the easiest solution is to use a specific library for that system (in this case Microsoft C++ .NET) or connect to existing system utilities to do the work (as can easily be done on UNIX systems). The same program can be developed quickly in Java and still be cross platform compatible simply by using the included standard networking libraries. Java indisputably wins any debate between it and the standard C++ language as far as networking libraries are concerned.
- g. Security:** C++, are weak to pointer hacks. That is, if a user gets access to a pointer to one part of the program, and they know where it is stored in memory in relation to the rest of the program, they only need to slightly change the pointer to gain access to potentially private parts of the program.[12]

3. General Comparison of above said languages in previous studies

A number of papers comparing the features of different high level languages have been published recently the comparison is done generally. The following table (Table-3) shows a general comparison of chosen languages (JAVA, C#, C++) [13].

TABLE 3: GENERIC COMPARISON OF JAVA, C# AND C++

	Java	C#	C++
Object-Orientation	Hybrid	Hybrid	Hybrid / Multi-Paradigm
Static / Dynamic Typing	Static	Static	Static
Generic Classes	No	No	Yes
Inheritance	Single class, multiple interfaces	Single class, multiple interfaces	Multiple
Feature Renaming	No	No	No
Method Overloading	Yes	Yes	Yes
Garbage Collection	Mark and Sweep or Generational	Mark and Sweep or Generational	None
Class Variables / Methods	Yes	Yes	Yes

Access Control	public, "package", private	protected, protected internal	public, protected, private, internal, protected internal	public, protected, private, "friends"
Multithreading	Yes	Yes	Yes	Libraries
Regular Expressions	Standard Library	Standard Library	Standard Library	No
Language Integration	C, some C++	All .NET Languages	All .NET Languages	C, Assembler
Built-In Security	Yes	Yes	Yes	No

4. Research Methodology:

Comparison of programming languages on selected criteria

To compare the behaviour of selected languages in some specific Interest we have selected a common program "Printing Fibonacci Series" to be executed in Java# and C++. The sample programs code is given below in all the above said languages in their respective development environment.

Java code for Fibonacci Series program

```
public class FibnoccSeries {
    public static void main(String[] args) {
        //number of elements to generate in a series
        int limit = 20;

        long[] series = new long[limit];

        series[0] = 0;
        series[1] = 1;

        for(int i=2; i < limit; i++){
            series[i] = series[i-1] + series[i-2];
        }

        System.out.println("Fibonacci Series upto " + limit);
        for(int i=0; i< limit; i++){
            System.out.print(series[i] + " ");
        }
    }
}
```

C# code for Fibonacci Series program

```
class FibnoccSeries
{
    static void Main(string[] args)
    {
        int p = 0;
        int q = 1;
        int n, r;
        n = 20;
        Console.WriteLine("-----The series is-----");
        Console.WriteLine(p);
        for (int i = 1; i <= n; i++)
        {
            r = p + q;
            p = q;
            q = r;
            Console.WriteLine(r);
        }
        Console.ReadKey();
    }
}
```

C++ code for Fibonacci Series program

```

#include<iostream.h>
#include<conio.h>
// Display Fibonacci Series
intmain()
{
inta=0,b=1, c,n;
cout<< "Enter the value of n ";
cin>> n; // n store the integer value that series will shown up to n terms
cout<<endl<< a <<" " <<b << " ";
for(int i =0; i<(n-2); i++)
{
c = a+b;
cout<< c << " ";
a=b;
b=c
}

```

For **Syntax**, from the above given code it is quite obvious that JAVA and C# program is more near to natural language as their readability level is more high when compare with the program written in C++. The syntax of C++ is a much more complex than the Java syntax.

For our comparison we have rate these languages as C# at number One (30 points), JAVA at number Two (20 point) and C++ at number Three (10 point) as to near to natural language. The Fig-2 gives a comparison analysis of the syntax of these languages

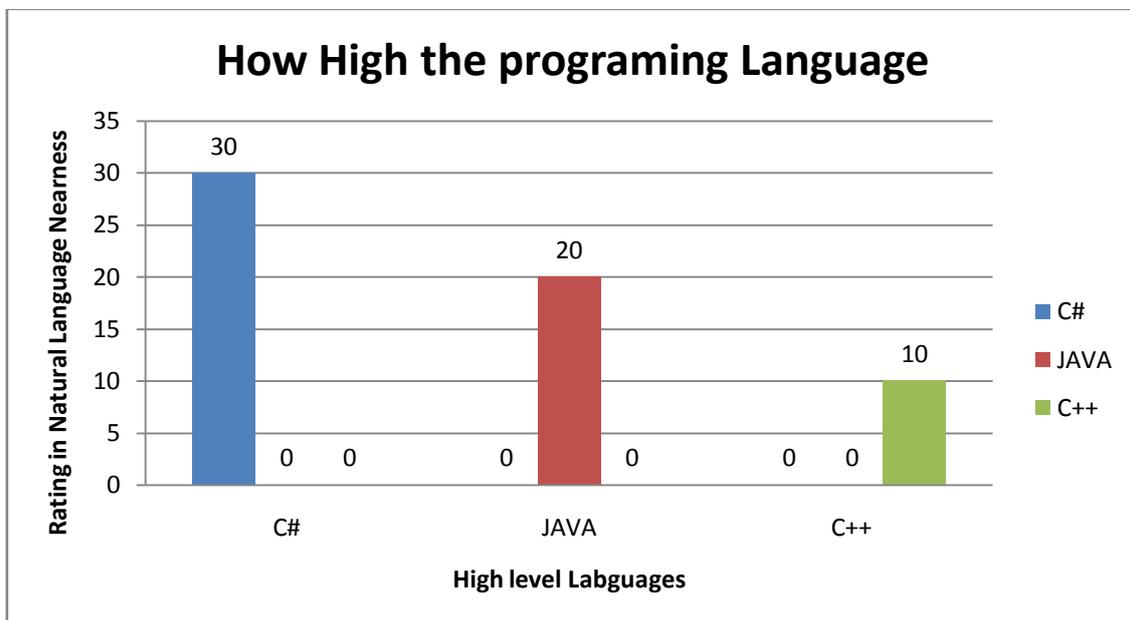


Fig-2 Syntax Comparison with Natural Language

Lines of code (LOC), is a software metric used to determine the size of a computer program. The size of program can be estimated by counting the number of lines of the code in its textual form, the size of program can help in estimating the programming productivity or maintainability [10]. More the number of lines of code more the efforts needed for maintainability. To calculate the lines of code used in the above written programs first we used separate programs for each selected language but the results were not

satisfactory so a common program “LocMetric” was used to count the number of lines in Fibonacci series program. The result is shown in the below given Figs.

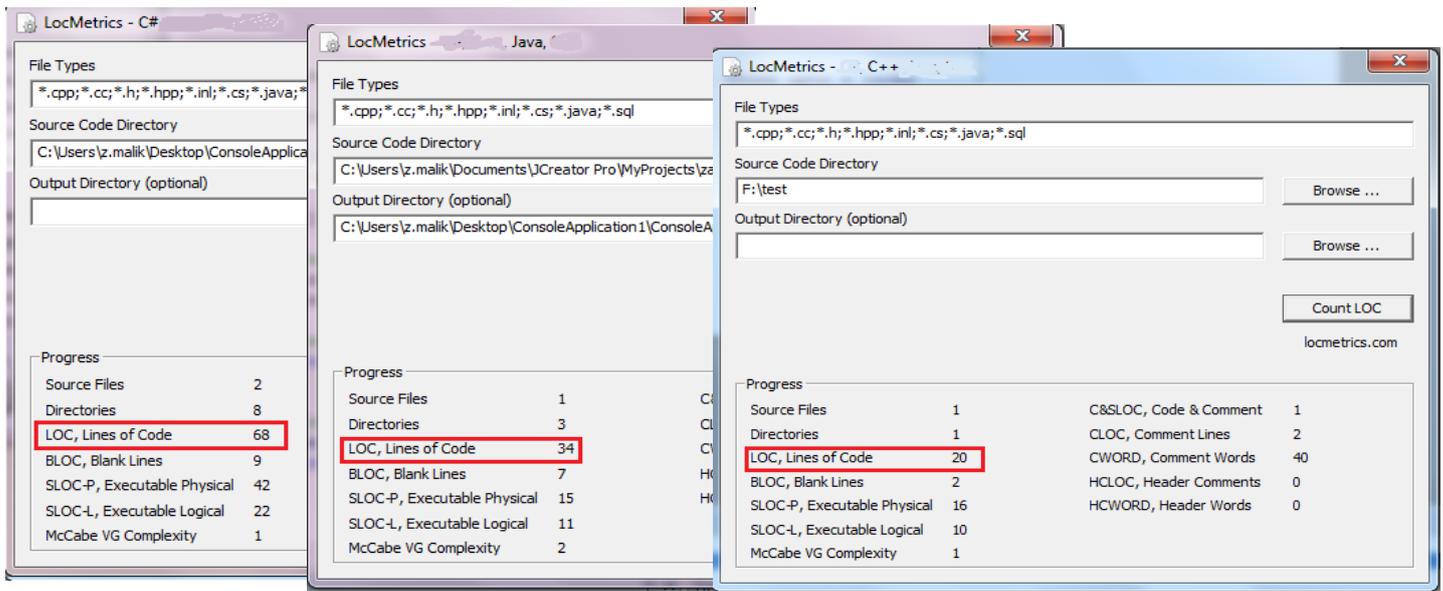


Fig-3-Lines of Code of program written in C#, JAVA, C++ using Loc Metric Tool

The output of LocMetric clearly shows the lines of code used in the above said program and there is noticeable difference in the length of code in all the languages. Fig-4 shows a comparison of these languages in respect to lines of code for the same program in the form of a bar chart.

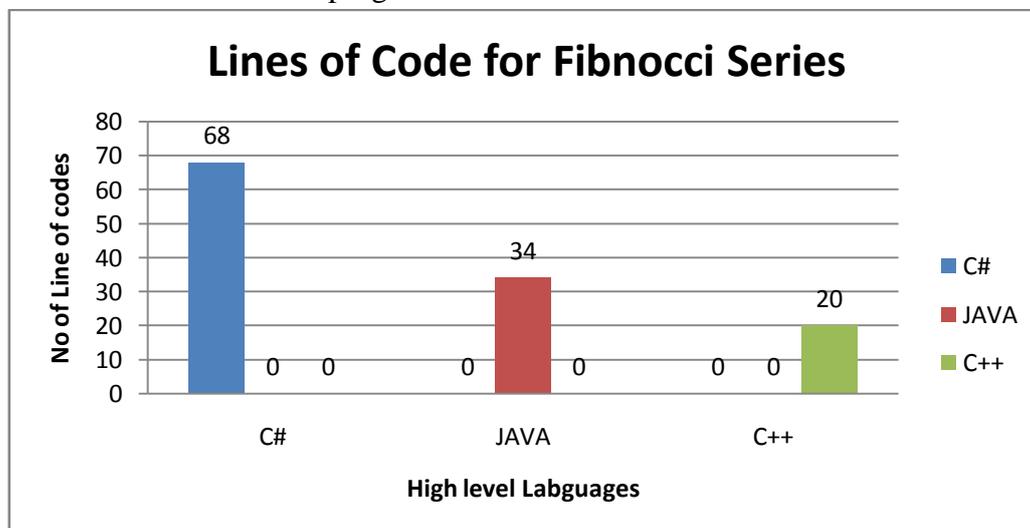


Fig-4 Comparison of Length of Code using C#, JAVA and C++

From the graph it can be concluded that C++ have the advantage of having the minimum lines of code java have the medium but C# having the maximum lines of code.

Compilation is a process of converting the source code into executable file using compiler. Compilation time is the amount of time taken by the compiler to convert course code to an executable file. Different high level programming languages use different processes for compilation, for reference Fig-1

shows the compilation and execution process of C# program. Depending on the process each language takes different amount of time for compilation.

Execution or running is a process in which executable file takes input from user and generate the output. Execution time is the amount of time taken to execute the executable file to produce output.

To compare the compilation and execution time of said languages many different methods have been applied, different methods produced different results. Even in some cases the results were different when execute at different times.

For **C#** calculation of execution time was done using Stopwatch class of using System namespace Diagnostics.the Fibnocci function was called after the start of stopwath and right after the end of method the stopwatch was stoped. The execution time was calculated as eight(8) Miliseconds.For **Javaa** simple getTime() method of Date() is used, the time before the execution of function was taken and also was taken at the end then by subtracting the times the final result was calculated as sixteen(16) Miliseconds. For **C++** also same method is usded like JAVA the compile time was calculated as 18 Milisecond.

The simple illustration of the comparison of selected languages in terms of execution time taken to compile and esxecute the Fibnocci series program is given in Fig-5 below.

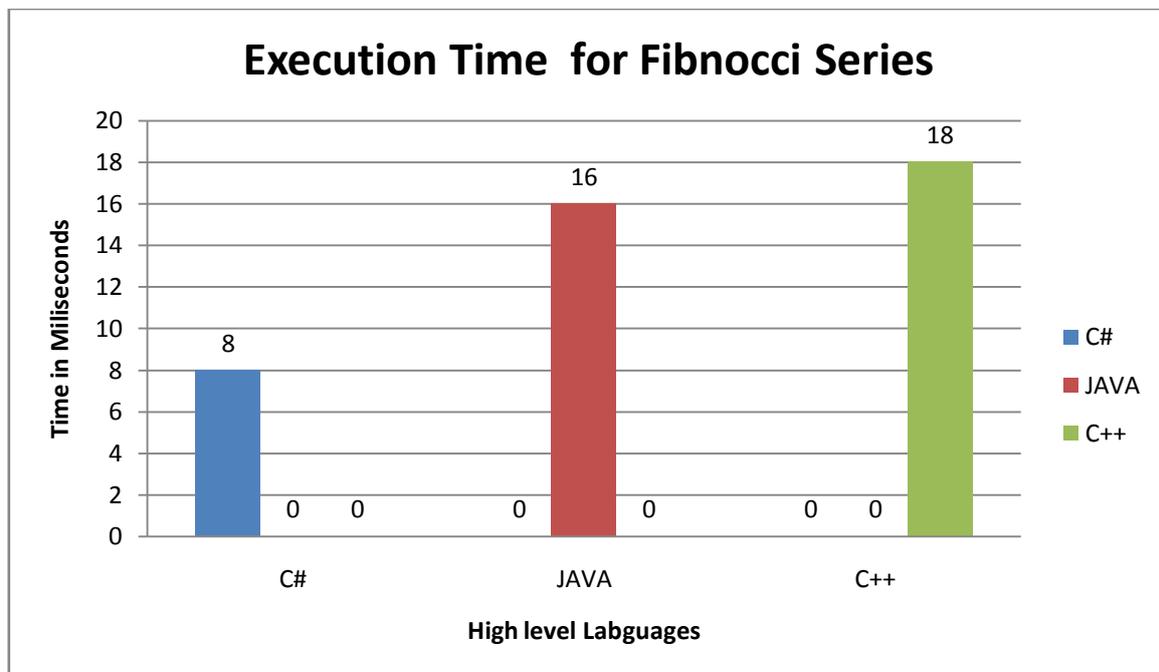


Fig-5 Comparison of Compile and execution Time

The clear difference in the compile time of the said languages can be observed from the above chart. There are many reasons for the difference mainly these causes are Header files, Linking, Parsing, Templates, Optimization and Machine code. Java compilation produces simple byte code for fast compilation whereas C++ produced Machine code that is very complicated to be made.

The fourth selected criterion to compare these languages is the most important one that is **Memory Management**. In this respect C# also take the higher place the second place is for JAVA and C++ again lack in memory management. C++ there are many objects that need to allocate the recourses when they are declared, after the utilization of these recourses programmer has to manually released the resources to free the memory pool. It is also discovered that there is a common problem of memory leakage in C++ (utilization of more and more resources needlessly).On the other hand if the resources are released before

the objects are destroyed the problem of loss of data or even corruption in some other memory area and null pointer exceptions can occur.[9]

Both Java and C# avoid these dangers by autonomously controlling the life span of all objects in use by an application. Java uses its special feature Java Virtual Machine (JVM) that is responsible for keeping the track of the reference of allocated resources and releasing the unused memory. The technique used for this purpose is known as garbage collection. So memory is managed automatically. C# uses feature common language runtime (CLR) to automatically manage the memory, the functionality is same as of JVM but an additional feature that is overflow protection. But JAVA has an extra advantage of memory safe due to memory management and not supporting of pointers.

For our comparison we have rate these languages as JAVA at number One (30 points), C# at number Two (20 points) and C++ at number Three (10 points) as to the memory management approach. The Fig-6 gives a comparison analysis of the memory management methods. [11]

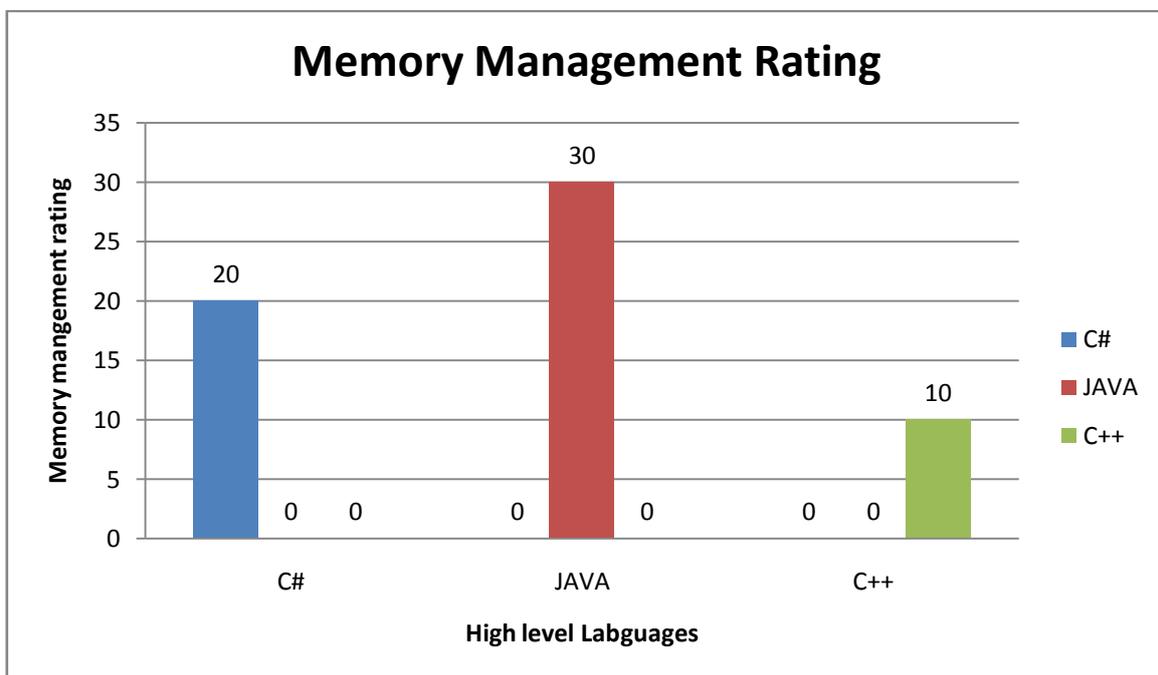


Fig-6 Memory Management Rating according to the techniques

The summarize comparison of the above stated characteristics of selected languages can be noticed in a combined graph that would show a clear comparative analysis. Fig-7 stated the selected characteristics of the languages ,the whole idea of our research can be clarify from the Fig below.

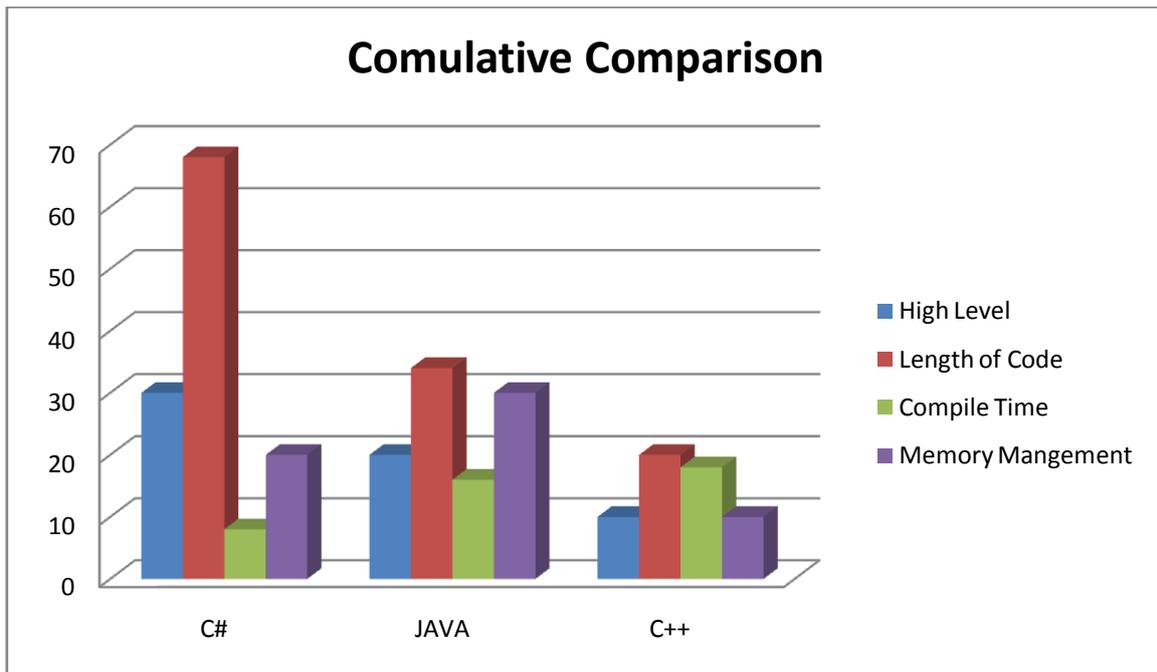


Fig-7: Comparison of different characteristics of high level languages

5. Conclusion

In this research paper we have done a comparative analysis of three most common high level languages C#, JAVA and C++ based on specific criteria. We used simple and most talked about features to provide the readers a better insight of the said languages. We considered features like emergence, platform dependency, speed and efficiency, performance in general and syntax, length of code, compilation and execution time and memory management in specific. Our goal is to raise the reader's awareness in the programming language while selecting them as their development tool.

Each of the languages considered have their own specific characteristics that distinguish them from the others. C# has the simplest and most nearest to natural language syntax while comparing with others. C# is a not an efficient language, it also not fully independent of the platform as compared to JAVA. The good thing about C# is its compilation time that is for less from C++ and JAVA. The length of code is again worst when comparing with others discovered in our research. Memory management is also reasonable in contrast to C++. Java one of the most popular and older language have a distinction of being fully independent from the platform and a complete object oriented language. The syntax is complicated when compare with C# but rather simple comparing with C++. JAVA is said to be the most secured language. It was concluded that JAVA have the best memory management techniques and also to write same function in JAVA a far less lines of code is required comparing with C#. The execution time is a little bit less than C++ but in our research methodology was double as compare to C#. The length of code is also reason able while comparing with C#. C++ one of the basic high level language, many other languages like C# adopt the syntax features from C++. Now a day's C++ is lacking the popularity because of the emergence of more easy to use high level languages but still C++ is providing the least number of code while comparing with other. C++ is the one of the most suitable language for hardware implementation.

The research concluded that there is no an ideal language to choose for all purpose, rather each language has its own distinguish characteristics depending upon the programmers requirements. If a highly secured system is needed JAVA can be preferred, high speed and easy to use application preferred C# and if the hardware implementation is the concern C++ preferred language.

REFERENCES

- [1] Nell Dale & John Lewis, "High level programming languages", 2000
- [2] Matthew Cook B.S. Software Engineering, Student Syed (Shawon) M. Rahman, Ph.D., Assistant Professor, 2001
- [3] <http://www.programmerinterview.com/index.php/java-questions/jvm-platform-dependent>
- [4] Pieter J. Schoenmakers<tiggr at gerbil.org> Programmers Without Deadlines Eindhoven, the Netherlands(2002)
- [5] Tutorials point, http://www.tutorialspoint.com/csharp/csharp_overview.htm.
- [6] C Sharp Programming : wiki books , 2015.
SouravKayal, " Improve Performance of C# Code", 2013
- [7] http://www.tutorialspoint.com/cplusplus/cpp_basic_syntax.htm
- [8] R. Henderson and B. Zorn,"A Comparison of Object-oriented Programming in Four Modern Languages", 1994
- [9] Maya Hristakeva, RadhaKrishnaVuppala,"A Survey of Object Oriented Programming Languages", 2011
- [10] S.H. Edwards. Inheritance: One mechanism, many conflicting uses. In Proc. 6th Ann. Workshop on Software Reuse.
- [11] Margaret A. Ellis and BjarneStroustrup. The annotated C++ reference manual. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.
- [12] D. Flanagan and Y. Matsumoto. The ruby programming language. O'Reilly Media, Inc., 2008.
- [13] Jerry Gao, Chris Chen, Y. Toyoshima, David Kung, and Pei Hsia. Identifying polymorphism change and impact in object-orientated software maintenance. Journal of Software Maintenance.