ISSN:0975-9646

Comparative Analysis of C, FORTRAN, C# and Java Programming Languages

Keshav Biswa[1], Bishal Jamatia[2] , Deepjyoti Choudhury[3], Pallabi Borah[4]

[1,2,3,4]Department of Computer Science and Engineering, Assam down town University
Gandhi Nagar, Panikhaiti, Guwahati – 26, Assam, India

***Abstract*— This paper compares and analyses four chosen programming languages i.e. C, Fortran, C# and Java based on certain runtime tests to understand the basic differences among the programming languages. We have also measured the performances of the chosen languages in various cases. The result obtained in various cases is also shown in this paper.**

***Keywords*— Programming Languages, C, FORTRAN, C#, Java.**

I. Introduction

Programming languages has developed very rapidly since early 1950's which lead to over hundreds of different programming languages being invented [1]. With the rapid advancement in technology, more and more programming languages started to come into picture. This leads to a need to understand which language is suitable in which situation or circumstances. Some languages clearly create a spark, which causes the languages to become popular--sometimes to a level of fanaticism [2], whereas some languages although good but remained in the shadows of these popular languages.

The paper discusses the overview of different programming languages in Section II, its practical comparison based on different cases in Section III, its overall results and analysis in Section IV and then we conclude the paper with the future work.

II. Overview of Programming Languages

Programming language in the most basic term can be defined as a language used by the humans to communicate with the machines. From the moment we turn on our computer some programs are always running which carries out instructions, loads the operating system, etc. Each and every program in a computer is written by a programmer in some programming languages.

A. FORTRAN

One of the oldest programming language in existence developed by a team of programmers at IBM led by John Backus in 1957. Its name derives as "FORmula TRANslation" is generally used for numeric and scientific computing. It is a general-purpose, imperative programming language. The first standard definition of this language was adopted in 1966 and a major revision was made in this standard in the 1970s, leading to the FORTRAN 77 and FORTRAN 90 in the year 1990 [5]. In its time FORTRAN was one of the most popular programming languages in the area of high performance computing and was developed "for execution on the IBM 704 computer" [5]. It opened the door to practical usage of computers by large numbers of scientific and engineering personnel [2].

B. C

One of the most popular languages till date is the C programming language used by both novice and expert programmers. It was developed in 1972 by Dennis Ritchie and Ken Thompson at AT & T Bell Labs. Although a general-purpose programming language, its compact syntax and efficient execution characteristics have made it popular as a system programming language [5]. C is an imperative (procedural) systems implementation language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal run-time support. A standards-compliant and portably written C program can be compiled for a very wide variety of computer platforms and operating systems with little or no change to its source code [3].

C. C#

C# is a modern, general-purpose, object-oriented programming language developed by Microsoft and approved by European Computer Manufacturers Association (ECMA) and International Standards Organization (ISO). C# was developed by Anders Hejlsberg and his team during the development of .Net Framework. C# is designed for Common Language Infrastructure (CLI), which consists of the executable code and runtime environment that allows use of various high-level languages on different computer platforms and architectures [4].

D. Java

The Java language was developed in 1991 by James Gosling of Sun Microsystems and released in 1995. The latest release of Java is the Java SE 8. Java is a write once run anywhere type of programming language. It is a secure, high performance and portable object oriented programming language. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities [3]. Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture [3].

III. Comparison of FORTRAN, C, C# and Java based on practical cases

Here in this paper, we compared and analysed four programming languages i.e. FORTRAN, C, C# and Java based on certain cases or algorithms. For this purpose, we chose four cases under which all these programming language performs its compilation and execution and its run-time was measured based on which comparison was made.

A. Bubble sorting algorithm

A simple program to bubble sort 11111 numbers was compiled and executed on all the languages and its result is given below:

GRAPH I
PERFORMANCE OF LANGUAGES ON BUBBLE SORTING ALGORITHM

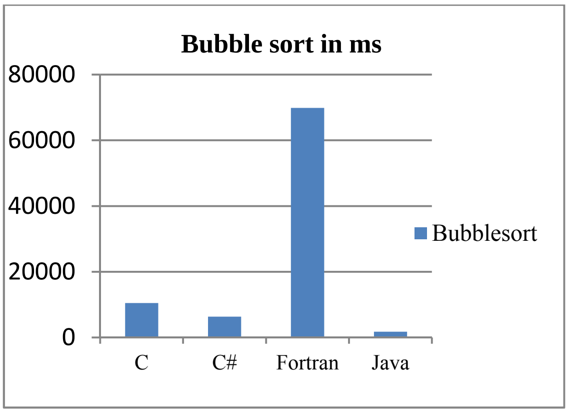


TABLE I
RUNTIME OF LANGUAGES (IN MILLISECONDS) ON BUBBLE SORT ALGORITHM

Test	Programming Languages (in ms)			
	Fortran	C	C#	Java
Mean calculation	26393	4900	2645	827

Here, the comparison was made in milliseconds. In this program, firstly the numbers to sort was assigned its values. All 11111 numbers' values were assigned a random number and then were sorted in ascending order using a bubble-sorting algorithm and then the time taken to execute was compared.

1) Algorithm of bubble-sort program

```
Begin BubbleSort(list)
   for 11111 element of list
         print unsorted list
         if list(i)>list(i+1)
              swap(list(i), list(i+1))
         end if
    end for
         print sorted list
return list
end BubbleSort
```

B. Mean of numbers

A program to take mean of 11111 integer numbers and display the numbers was made and executed in all the languages and comparison was made. Its graph and result is given below:

GRAPH II
PERFORMANCE OF LANGUAGES ON MEAN OF NUMBERS

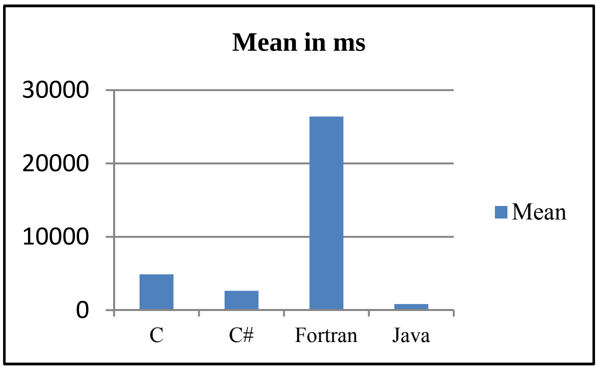


TABLE II
RUNTIME OF LANGUAGES (IN MILLISECONDS) ON MEAN OF NUMBERS

Test	Programming Languages (in ms)			
	Fortran	C	C#	Java
Bubble-sort	69849	10490	6311.3	1747.6

Here, the 11111 numbers used for mean calculation was stored in an array and its values were assigned by its array position. For example, the array (2) stored value '2' and so on. Then its mean was calculated using a variable 'sum' to store the sum of these numbers and then dividing it by 11111.00. The time taken for this program to finish its execution was calculated.

1) Algorithm for mean of 11111 numbers

```
Begin Mean( float avg)
      for i=1 to 11111 numbers
            arr(i)=i
            sum=sum+i
            print arr(i)
      end for
      avg=(double) sum/11111
      return(avg)
end Mean
```

C. Read and Display of Files

A simple program to create, read and display a data structure i.e. a file was made in all four program languages and its runtime was compared.

GRAPH IIII
PERFORMANCE OF LANGUAGES ON FILES

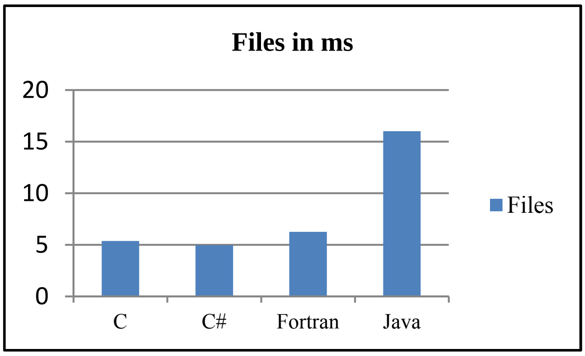


TABLE III
RUNTIME OF LANGUAGES (IN MILLISECONDS) ON FILES

Test	**Programming Languages (in ms)**			
	Fortran	**C**	**C#**	**Java**
Files	6.256	5.367	4.968	16

Here, firstly a simple text file was created with a number of random names. Then a program was made which would open this file, read it, display its contents and then close it.

The calculation was made based on how long the language took to display its contents and to close it in milliseconds.

1) Algorithm for reading and displaying a file

```
Begin Files
        line="", counter =0
        Open file(textfile.txt)
        While (line=readLine()!=null)
                Print line
                Counter++
        end while
end Files
```

D. Basic Benchmark Program

A simple benchmark program was made to test the compiler performance of the four programming languages.

The program was a simple looping of a variable up to 100000 times and displaying its values after every iteration.

The result of this program is given below:

GRAPH IIIV
PERFORMANCE OF LANGUAGES ON THE BENCHMARK PROGRAM

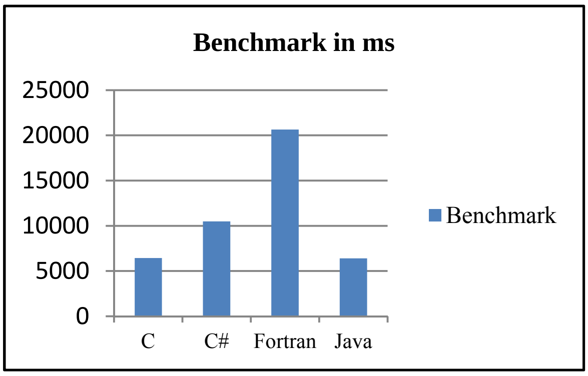


TABLE IV
RUNTIME OF LANGUAGES (IN MILLISECONDS) ON THE BENCHMARK PROGRAM

Test	**Programming Languages (in ms)**			
	Fortran	**C**	**C#**	**Java**
Benchmark	20640	6430	10491.1	6396

Here, the program consisted of two variables. One variable was used to loop 100000 times and the other variable was assigned to calculate the sum of the former variable every iteration and to display it. Calculation was based on how long it took to finish its execution until the last loop.

1) Algorithm for benchmark program

```
Begin BenchMark
        For i=1 to 100000 elements
                j=i+1
                print j
        end for
end BenchMark
```

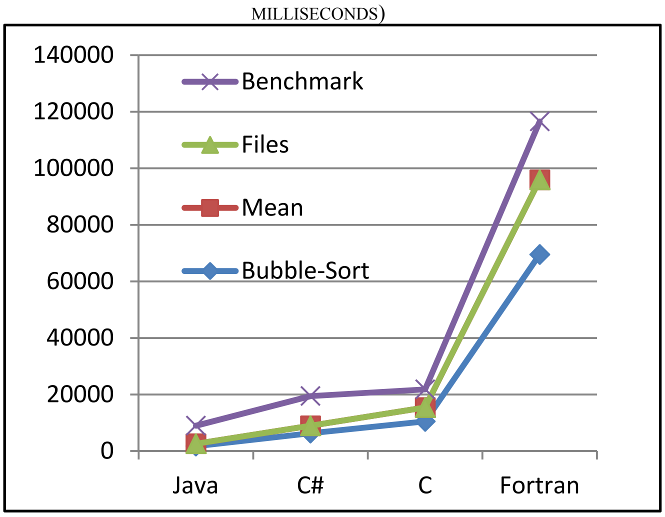
IV. RESULTS AND ANALYSIS

After the comparison of the four chosen programming languages, we get the result as shown in table:

TABLE V
FINAL RESULT TABLE OF THE LANGUAGES

Tests	**Programming Languages (in ms)**			
	Fortran	**C**	**C#**	**Java**
Bubble-sort	69849	10490	6311.3	1747.6
Mean	26393	4900	2645	827
Files	6.256	5.367	4.968	16
Benchmark	20640	6430	10491.1	6396

GRAPH V
FINAL GRAPH DENOTING THE PERFORMANCE OF LANGUAGES (IN MILLISECONDS)



Here we observed that Java performed the best in the bubble sorting test and FORTRAN performed the worst and also in the mean calculation test whereas in the benchmark test C took the least execution time and FORTRAN had the highest execution time. In file execution tests, C# performed the best and again FORTRAN performed the worst.

V. PLATFORM

All the analyses were made on windows 8.2 OS systems with 4 GB RAM with Intel CORE i3 processor of 1.70 GHz. The compilers used for the languages were:

A. For Fortran - Silverfrost FTN95 for Microsoft Visual Studio .NET.
B. For C - Turbo C/C++ 4.0 Windows 8 64-bit version.
C. For C# - Microsoft Visual C# 2010.
D. For Java – Eclipse Mars.2 (4.5.2) version.

VI. CONCLUSION

The comparative study done so far provided us a basic knowledge of how different programming language executes and how long it takes for it to compile and execute. Here we can conclude that object oriented programming language like C# and Java performs better in these cases as compared to procedure oriented programs such as C and FORTRAN. This can help a novice programmer to choose its language more efficiently based on these concepts.

VII. FUTURE WORK

In future, we would like to extend our comparative analysis by comparing languages based on its features, orientation and its field of applications and also analyse on the fifth generation programming languages.

ACKNOWLEDGEMENT

We would like to thank and show our gratitude towards all these persons who have provided us help throughout our work.

We would like to thank Assam down town University for providing us the facilities which was needed for us to finish our work.

REFERENCES

[1] H. Chen, “Comparative Study of C , C ++ , C # and Java Programming Languages Degree Program of Information Technology”, vol. 2, no. 5, pp. 11–39, 2010.

[2] J. E. Sammet, “Programming Languages : History and Future”, Communication of the ACM, vol. 15, no. 7, July 1972.

[3] J. Paquet and S. A. Mokhov, “Comparative Studies of Programming Languages; Course Lecture Notes”, p. 66, August 2010.

[4] K. E. Jawa and K. E. Jawa, “C. Overview”, pp. 64–91,2009.

[5] Terrence W.Pratt, Marvin V. Zelkowitz and Tadepalli V. Gopal, "PROGRAMMING LANGUAGES, Design and Implementation", 4th Edition, Pearson, 2001.