IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 3, Issue 11, November 2016

Analytical Comparison of the Programming Languages

Ruchi Sharma[1], A.J. Singh[2], Pankaj Gupta[3]

Research Scholar, Department of Computer Science, Himachal Pradesh University, Shimla, India [1]

Professor, Department of Computer Science, Himachal Pradesh University, Shimla, India[2]

DIO, National Informatics Centre, Shimla, India[3]

Abstract: A programming language is a language developed to communicate instructions to a computer. Programming languages are used to create programs which are used for solving our day to day problems and/or to express algorithms briefly. There are two categories of programming languages the Low level Languages and High Level languages. This paper, presents a comparative study of three programming languages, C, C# and JAVA, with respect to the following criteria: memory usage and CPU usage.

Keywords: C, Java, C#, Hardware, Memory Usage and CPU Usage.

I. INTRODUCTION

A programming language is a systematic notation by which we describe computational processes to computers. A computational process means a set of steps which a machine can perform for solving a task [1]. A programming language is taken into consideration to be a fixed of characters and regulations for combining them which have the subsequent traits: (1) machine code knowledge makes no sense, (2) there is good potential for conversion to different computers, and (3) there is an instruction explosion (from one to many) [2].

The C Programming Language

The C language is a robust language. As it combines the capabilities of high level languages with the skills of low-level languages, it is appropriate for writing business packages, as well as system software and applications. C is a influential, proficient, well-known cause established programming language. It is dependable, simple and easy to use and consequently it turns into very popular programming language. It isn't always simplest restrained to programming experts and experts however any computer beginner who has expertise of its syntax and programming method can make packages in it. It's been chosen for this study as it's far the foundational language of many different programming languages.

The C# Programming Language

The c# programming language is a multi-paradigm, object-oriented programming language. The goal of the language is to enhance a programmer's productivity. It is getting in popularised because of its efficiency and ease of coding. The framework manages the execution of programs and internet offerings. In addition, it offers several other functionalities that include memory management and protection enforcement. Like java, c# additionally has automated garbage collection and has a comparable syntax structure.

The JAVA Programming Language

The Java programming language has been chosen for this study since it is dynamic and it is designed to adapt to an evolving environment. Java programs can bring vast amount of run-time records that may be used to confirm and resolve accesses to gadgets on run-time. Java compiler generates an architecture-neutral code which makes the compiled code to be executable on different processors with the presence of java runtime device. When an application written in Java is compiled, it generates an intermediate code which is known as byte code. The byte code can be executed directly by any computer. It is object oriented, distributed, multithreaded, and strongly typed.

II. CRITERIA OF LANGUAGE COMPARISON

The following criteria have been considered for comparing various languages:

Reusability: Does the language support effective reuse of program units if so then the project can be accelerated by reusing tried-and-tested program units. Relevant concepts here are packages, abstract types, classes, and particularly generic units. Reason for popularity of Object oriented languages such as JAVA etc. is reusability.

Portability: Does the language support portability or not? In other words, can the code be moved from one platform to a dissimilar platform without major changes? Portability is the big issue now a day's.

Reliability: Can the language be designed in such a way that programming errors can be detected and eliminated as quickly as possible? Errors detected during compile-time are guaranteed to be absent in the running program, which is ideal. Errors detected during run-time checks are guaranteed to cause no harm other than throwing an exception or at worst terminating the program, which is second-best. Errors not detected at all can cause unlimited harm before the program crashes. Reliability is always important especially in safety-critical systems.

Efficiency: Is the language capable of being implemented efficiently? Some aspects of object-oriented programming entail run-time overheads, such as class tags and dynamic dispatch. Garbage collection is also costly, slowing the program down at unpredictable times. Interpretive code is about ten times slower than native machine code.

Readability: Does the language help or hinder good programming practice? A language that enforces difficult syntax, very short identifiers, default declarations, and an absence of type information makes it difficult to write readable code. The significant point is that code is read (by its author and other programmers) more often than it is written.

Expressiveness: This factor reflects the ability of a language to express complex computations or complex data structures in appealing, intuitive ways [3].

III. EVALUATION OF PROGRAMMING LANGUAGES

• C Programming Language

Its definition should be independent of any particular hardware or operating system: C language in the beginning was defined as a language for the system software program on Unix structures. But, it has advanced to a language independent of Unix or any precise platform.
It should effectively support the application domain(s) of interest: Even though it was initially developed to assist device software program, C has tested to be a very versatile language, supporting any domain in which it has been tried.
It should support the required level of system reliability and safety: C provides little support for reliability. Safety-critical systems, those on which human life may depend, are also not effectively supported by C because of its lack of support for software engineering technology.

Readability: Although it is possible to write C code which is understandable, it is not common practice to use a verbose, understandable style. C provides cryptic shortcuts that run counter to clarity, and they are commonly used.
Maintainability: A C programmer must work very carefully to write maintainable code because the language provides little inherent support.
Portability: The existence of a standard for C makes portability possible. However, common practice does not necessarily adhere to this standard. There are also no inherent language features that facilitate portability, such as the encapsulation of dependencies. The tremendous popularity of C has spawned tools and tool sets that are widely available on many platforms, enhancing portability.
Reliability: C provides little in the way of inherent language features to support reliability. It readily allows inconsistencies to show up in compiled code.
Reusability: Support for reusability requires support for code clarity, encapsulation, maintainability, and portability. C provides little inherent support for any of these characteristics. Hence, it does not support development reuse on a large scale. On the other hand, reuse of specialized C libraries, such as graphics libraries, is very effective.
Safety: C does not provide good support for any safety features.

• C# Programming Language

Its definition should be independent of any particular hardware or operating system: C# is an object-oriented programming language from Microsoft that aims to combine the computing power of C++ with the programming ease of Visual Basic. C# is based on C++ and contains features similar to those of Java. It requires .Net platform.
It should effectively support the application domain(s) of interest: C# is designed to work with Microsoft's .Net platform. Microsoft's aim is to facilitate the exchange of information and services over the Web, and to enable developers to build highly portable applications.

Readability: C# is strictly object oriented, so its form is very well defined. And also is backed by Microsoft's well structured code editor.
Maintainability: Many features of C# support maintainability, such as those which support code clarity, encapsulation, and object orientation. Object-oriented capabilities can have both good and bad effects on maintainability, but if used properly, object-oriented programming will improve maintainability.
Reliability: C# is based on .Net Technology. Microsoft addressed and cleared many reliability and stability issues with .NET technology, and now this technology is reliable.
Reusability: C# simplifies programming through its use of Extensible Markup Language (XML) and Simple Object Access Protocol (SOAP) which allow access to a programming object or method without requiring the programmer to write additional code for each step. Because programmers can build on existing code, rather than repeatedly duplicating it, C# is expected to make it faster and less expensive to get new products and services to market.
Safety: With the managed code in C#, the CLR can perform checks for type safety, memory overwrites, memory management and garbage collection. Type safety is another feature that promotes robust, safe programs. Microsoft incorporated several features to promote proper code execution in C#.

• Java Programming Language

Its definition should be independent of any particular hardware or operating system: Java is completely independent of any particular hardware or operating system.

It should effectively support the application domain(s) of interest: Java was developed specifically to support WWW applications. However, it is also a general-purpose language. Although it is still a very young language, it has proven to provide good support for any domain in which it has been tried.

It should support the required level of system reliability and safety: Java provides many features which support system reliability. For safety-critical systems, those on which human life may depend, no current language other than java is entirely satisfactory. Java has not been around long enough to be studied for suitability for safety-critical systems. However, since it does not provide formal analysis features, it would require a combination with mathematical specification, rigorous analysis, and formal proofs of correctness before it could be appropriate for use in safety-critical systems.

Readability: Java is strictly object oriented, so its form is very well defined. The code suffers somewhat from the cryptic C syntax forms.

Portability: Java was built for complete portability. Its compiler produces source code in a platform-independent byte code. The byte code is then translated at runtime into native machine code for the given platform.

Reliability: Java requires the specification of information, the omission of which can make a program unreliable, such as type specifications.

Reusability: Java supports reusability with language features supporting code clarity (making code understandable), encapsulation (making code adaptable), maintainability, and portability.

Safety: Java was not developed for safety-critical systems, and its capabilities in that area are unproven.

IV. RESEARCH METHODOLOGY

The analytical approach was used for obtaining the results. Visual studio has been employed as the platform for implementation of programming languages for linear search algorithm and bubble sort algorithm. Then, the results were compared for CPU usage and memory usage. Visual studio also has a function known as profiler for calculating the CPU usage and memory usage.

Features of Microsoft Visual Studio:

1. It consists of a code editor that supports syntax highlighting and code completion with the use of IntelliSense for variables, capabilities, techniques, loops and LINQ queries.
2. Visual studio consists of a debugger that works each as a source-level debugger and as a machine-level debugger. It works with both managed code in addition to local code and may be used for debugging packages written in any language supported through visual studio.

V. RESULTS AND ANALYSIS

This section presents the results of comparison study of the programming languages C, C# and JAVA, with respect to the following criteria: memory usage and CPU usage on the Bubble Sort and Linear Search algorithms. Three different data sizes 10000, 50000 and 100000 have been used.

For the algorithm implementation, Microsoft Visual Studio 2015 has been used.

After implementing algorithms, following results have been obtained:

Table 1.1: CPU Usage Time in Seconds of Bubble Sort

Data Size	C	C#	JAVA
10000	5.226	11.526	4.853
50000	24.659	42.79	15.369
100000	61.8	130.2	57.29

Table 1.2: CPU Usage Time in Seconds of Linear Search

Data Size	C	C#	JAVA
10000	5.858	7.893	5.694
50000	10.803	11.636	7.638
100000	17.422	18.589	11.668

Table 1.3: Memory Usage in MB of Bubble Sort

Data Size	C	C#	JAVA
10000	0.83789	7	0.31738
50000	1	8	0.4658
100000	1	8	0.6621

Table 1.4: Memory Usage in MB of Linear Search

Data Size	C	C#	JAVA
10000	0.85546	6	0.44726
50000	1	8	0.5996
100000	1	8	0.78515

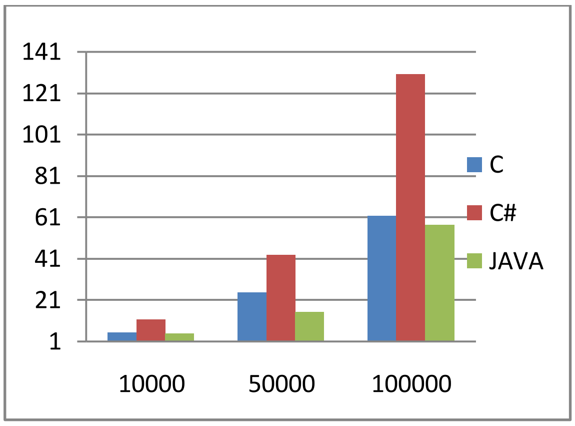

Figure 1.1: CPU Usage of bubble sort

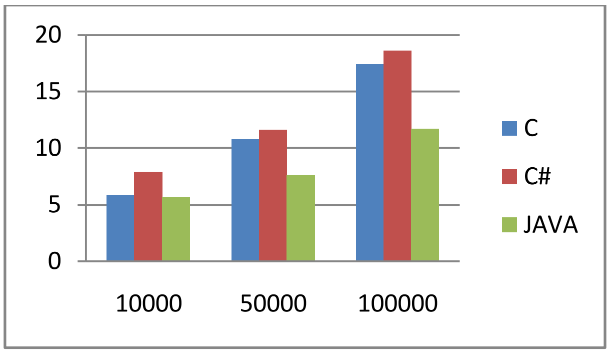

Figure 1.2: CPU Usage of linear search

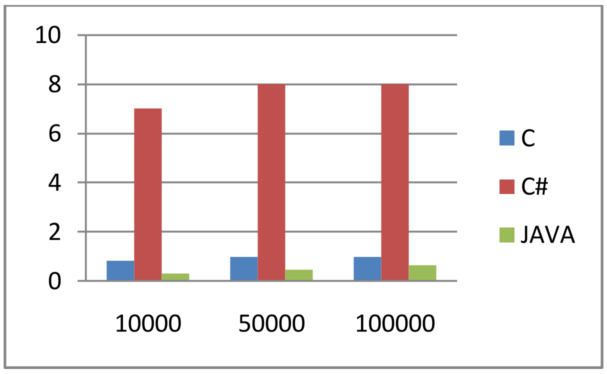

Figure 1.3: Memory usage of bubble sort

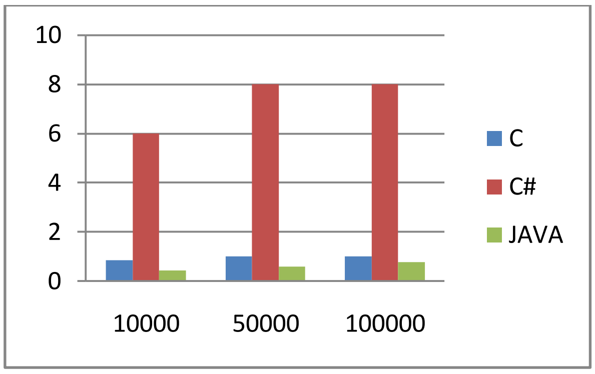

Figure 1.4: Memory Usage of linear search

VI. CONCLUSION AND FUTURE SCOPE

In this study, the results are based on measurement values taken from the following categories: CPU Usage and Memory Usage. Some languages performed as anticipated and others performed with some unexpected outcomes. The research provided a well-defined perspective of how each language performed based on the memory consumption, CPU utilization, of three 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 leader. In case of CPU utilization in Bubble Sort and Linear Search C# is third, C is second and java becomes the first.

Further writings and experimentation on this subject could include comparing programming languages across platforms using this study. Perhaps a researcher could compare these languages result obtained using the same categories in a LINUX to those obtained here in the Microsoft environment, and see how the programming languages behave differently.

REFERENCES

[1] Ellis Horowitz, Fundamentals of programming languages, second edition, 1994.
[2] Jean E. Sammet, “Programming Languages: History and Future”, Communication of ACM, July 1972, Volume 15.
[3] Terrence W. Pratt and Marvin V. Zelkowitz, Programming Languages Design and Implementation, Prentice Hall, Inc, Fourth Edition, 2001.
[4] P. Kulkarni, H.D. Kailash, V. Shankar, S. Nagarajan and D.L. Goutham, “Programming Languages: A Comparative Study”, Information Security Research Lab, NITK, Surathkal.
[5] MICHAEL MCMILLAN, Data Structures and Algorithms Using C#, First Edition, 2007.
[6] Michael T. Goodrich, Data Structures and Algorithms in Java, Fourth Edition John Wiley & Sons, Inc, 2010.
[7] Yashavant P. Kanetkar, Data Structures Through C, Second Edition, 2009.