VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Hao Chen

Comparative Study of C, C++, C# and Java Programming Languages

Information Technology

2010

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Degree Program of Information Technology

ABSTRACT

Author	Hao Chen
Title	Comparative Study of C, C++, C# and Java Programming Languages
Year	2010
Language	English
Pages	78 + 4 Appendices
Name of Supervisor	Dr. Ghodrat Moghadampour

With the rapid development of software industry, more and more people want to learn programming languages. But nowadays there are more than 200 programming languages available, only a few of them can be applied comparatively widely.

In this thesis, the research in programming language was conducted. Four of the most popular programming languages C, C++, C# and Java are chosen to be the objects to study. The technical features of these four programming languages were summarized and compared with each other. To know the actual performance of these four chosen programming languages, an experiment was carried out by implementing the benchmark for each programming language. The result from the experiment was recorded and analyzed.

The research concluded the most suitable application fields for these four of the most popular programming languages according to the technical features and the result from the experiment. C is suitable for systems-programming applications, hardware related applications, embedded device, chip designing, and industrial automation products. C++ is appropriate for the software development such as application software, device drivers and high-performance server. C# is proper for application development and the development of web application. Java has three different forms, Java2 Standard Edition (J2SE), Java2 Micro Edition (J2ME), and Java2 Enterprise Edition (J2EE). J2SE is suitable for the desktop applications. J2ME is proper for embedded systems development for mobile phones, wireless application and PDA programming. Finally, J2EE is appropriate for the development of server programming.

Key word: Programming language, C, C++, C#, Java

Content

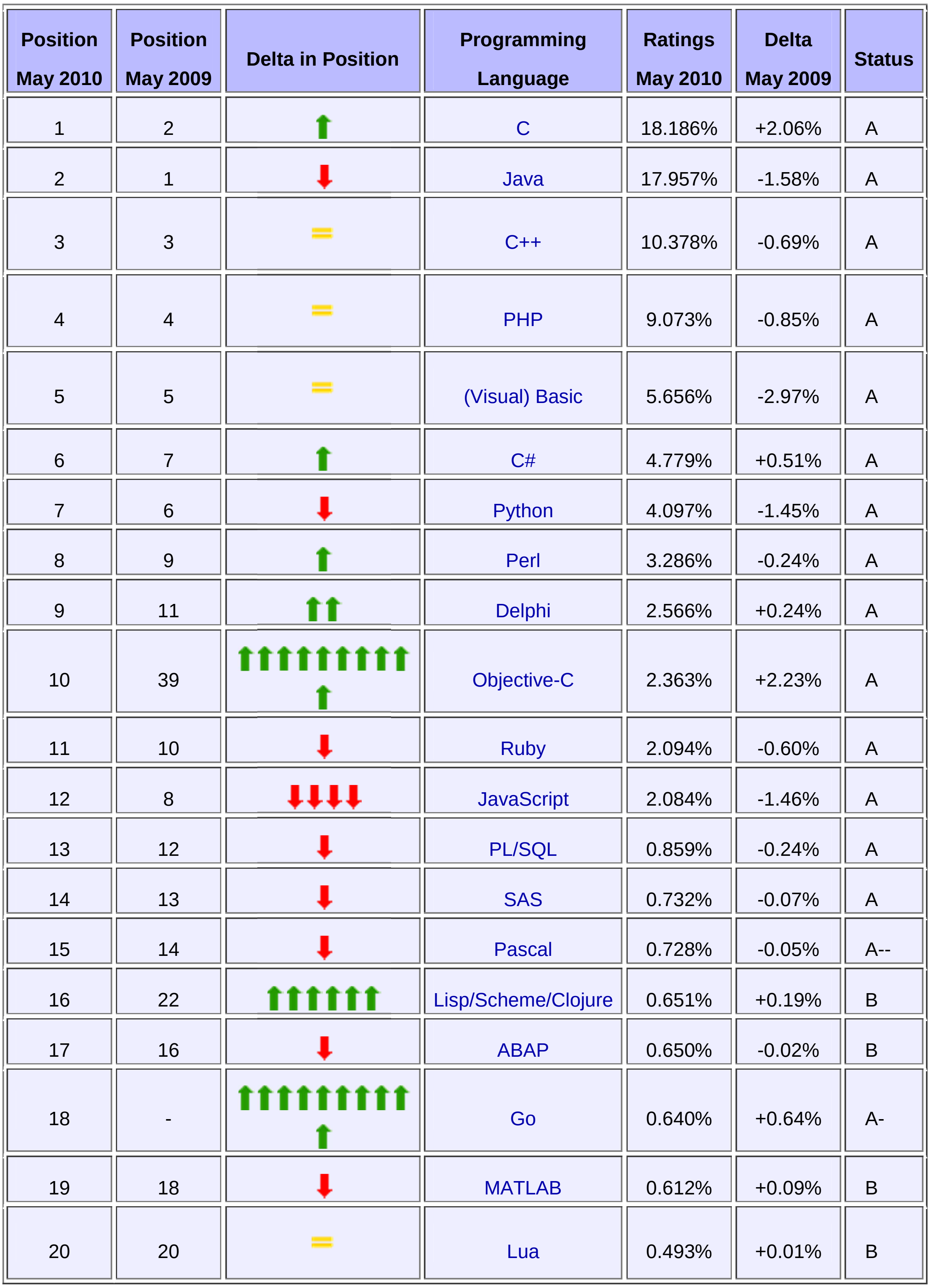
1. Introduction

1.1 Introductory paragraphs

Software has extended many areas of modern life. People might use the software to write the article in the office, communicate with each other and play games on the computer or on the phone to relax. In recently years, the applications of software have dominated in many fields of our life. First, there is office software such as Microsoft Office and Open Office which brought us a huge improvement in the efficiency of study and work related to the text editor. Secondly, with the wide application of communication software such as Skype, MSN, Yahoo Message, the communication between people become more and more convenient. Furthermore, it is also became cheaper and cheaper for people to contact with each other. For example, by using one software call “Fring”, people can make free mobile call through the wireless network. Last but not at least, more and more vivid and attractive games are available for the game machines such as “play-station 3” and “WII”. By playing the games, people can not only relax but also do exercise.

Clearly the public has experienced the power of the software. And all the software is written by the programming languages. The development in the technology of programming languages is also rapid. Since the first generation of programming languages emerges in early 1950, programming languages have five generations. The latest generation of programming languages is aimed to “make the computer solve a given problem without the programmers.” [1] However, according to the TIOBE Programming Community index that “gives an indication of the popularity of programming languages” [2], third generation of programming language is still be widely used. The table 1 shows the TIOBE Programming Community index for May 2010.

Table 1. The TIOBE Programming Community index for May 2010. [2]



Position May 2010	Position May 2009	Delta in Position	Programming Language	Ratings May 2010	Delta May 2009	Status
1	2	↑	C	18.186%	+2.06%	A
2	1	↓	Java	17.957%	-1.58%	A
3	3	=	C++	10.378%	-0.69%	A
4	4	=	PHP	9.073%	-0.85%	A
5	5	=	(Visual) Basic	5.656%	-2.97%	A
6	7	↑	C#	4.779%	+0.51%	A
7	6	↓	Python	4.097%	-1.45%	A
8	9	↑	Perl	3.286%	-0.24%	A
9	11	↑↑	Delphi	2.566%	+0.24%	A
10	39	↑↑↑↑↑↑↑↑↑↑	Objective-C	2.363%	+2.23%	A
11	10	↓	Ruby	2.094%	-0.60%	A
12	8	↓↓↓↓	JavaScript	2.084%	-1.46%	A
13	12	↓	PL/SQL	0.859%	-0.24%	A
14	13	↓	SAS	0.732%	-0.07%	A
15	14	↓	Pascal	0.728%	-0.05%	A--
16	22	↑↑↑↑↑↑	Lisp/Scheme/Clojure	0.651%	+0.19%	B
17	16	↓	ABAP	0.650%	-0.02%	B
18	-	↑↑↑↑↑↑↑↑↑↑	Go	0.640%	+0.64%	A-
19	18	↓	MATLAB	0.612%	+0.09%	B
20	20	=	Lua	0.493%	+0.01%	B

According to the table above, this thesis chooses C, C++, C#, Java that are the four of the most popular programming languages as the subjects to study.

1.2 Statement of the problem

The number of the programming languages is thousand from general purpose programming languages to special purpose programming languages that are used in one application. People always have difficulties about which programming language should be learn and use to develop certain software.

The main problem of this thesis:
Among the four of the most popular programming languages C, C++, C# and Java, which programming language should be adopted when starting to build a new software system to meet the general requirements of this new software system such as stable performances with fast speed and the limited time for developing?

This thesis is aimed to solve the problem above by the theoretical analysis and the practical experiment.

1.3 The purpose

By the introduction of programming languages and the deep analysis of four chosen programming languages C, C++, C# and Java from theatrical aspect to practical aspect, this thesis expresses the general idea about the whole programming languages, the deep insight in technical features and actual performance of these four chosen programming languages. The main purpose of this thesis is to guide people to make a strategic decision to use the most suitable programming language to develop the new software system.

1.4 Significance of study

The importance of this study is significant.

A software system built with wrong programming language can lead to some big problems. The time spent on the development of the new software system can be very long. The maintenance of the new software system can be very difficult. Even worse thing can happen. Because of the poor performance of the new software system, it has to be rebuilt with a new programming language. This is really a big waste of time, money and human resource.

This thesis can help people avoid these problems mentioned above. With a correct programming language, the time spent on developing a new software system can be reduced significantly and the performance of the new software system can be improved obviously.

1.5 The restriction of the thesis

In this thesis, only four of the most programming languages C, C++, C# and Java are chosen to be the subjects to introduce and compared with each other from theoretical aspect, such as language grammars and technical features to the practical aspect-running benchmark of these four chosen programming languages to know the actual performance of these four chosen programming languages.

The other programming languages were not discussed and compare in this thesis.

1.6 Outline of study

The outline of the study follows a structural pattern with five chapters after this, concerning:

Chapter II – Brief history of programming languages, C, C++, C#, and Java

Chapter III- Comparison of C, C++, C#, and Java from theoretical aspects, including summarizing the technical features of these four chosen programming languages and comparing these features with each other

Chapter IV- Experiment on the benchmark of C, C++, C#, and Java, including the description of experiment of running the benchmark of four chosen programming languages and the result of experiment and their deep analysis.

Chapter V- Conclusion, including summering the advantages and disadvantages of four chosen programming languages and the recommendation of their most suitable fields to apply

2. Brief history of programming languages, C, C++, C#, and Java

In this chapter, the definition of programming languages and the brief history of their development were introduced by the time sequence. Then the brief history of four of the most popular general-purposed programming languages C, C++, C# and Java were presented.

2.1 Brief introduction of programming languages

"A programming language is an artificial language designed to express computations that can be performed by a machine, particularly a computer." [3]

Programming languages has developed very rapid since early 1950's which lead to over hundreds of different programming languages being invented. With the rapid technological development in the hardware with faster processors, more and more powerful programming languages can be used to meet the demands of designing more efficient programs for various applications. In this chapter, five generations of programming languages were briefly described by time sequence.

2.1.1 Machine languages

The first generation of programming languages is machine languages appearing in early 1950's. It was written in binary, a series of zeros and ones. Binary is difficult to understand for human-being and was very prone to errors. The main problem of machine languages is machine dependency because "Machine languages were created differently for different for each CPU". [4]

2.1.2 Symbolic assembly languages

The second generation of programming languages is symbolic assembly programming languages which was “written in a more simplistic form and at a higher level of abstraction machine languages, instead of a series of zeros and ones there were symbols (percent, dollar and portions of a word and number combination used to make commands.” [4] But symbolic assembly programming languages still have the problems of high hardware dependency and lack of portability which means assembly codes could not implemented on the different machines.

2.1.3 Problem-oriented languages

The period from the early 1960's till 1980’s brought us third generation programming languages. Third generation of the programming languages is called problem-oriented languages and is also considered as the high level languages. “Third generation programming languages were converted from English into machine languages; compilers were used to convert these instructions.” C, C++, C and Java are all examples of third generation programming languages. Most of third generations of programming languages have the compilers or the interpreters. The advantage of this is that the programs can run very fast after compiling. The problem of “machine dependency” which was encountered by the first and second generation of programming languages is no longer for the third generation of the programming languages any more. The main problem of third generation programming languages is that different type of processors need different source codes of third generation programming languages and third generation programming languages are quite hard to work out. [4]

2.1.4 Non-procedural languages

Fourth generation programming languages are more focused on problem solving. The main difference between other generation programming languages is that they are more concerned with what is to be done than the actual how. “Features evident in fourth generation languages quite clearly are that it must be user friendly, portable and independent of operating systems, usable by non-programmers, having intelligent default options about what the user wants and allowing the user to obtain results fasts using minimum requirement code generated with bug-free code from high-level expressions (employing a data-base and dictionary management which makes applications easy and quick to change).” The most well-known examples of fourth generation programming languages are SQL, MYSQL. [4]

2.1.5 Fifth-generation programming languages

“A fifth-generation programming language (abbreviated 5GL) is a programming language based around solving problems using constraints given to the program, rather than using an algorithm written by a programmer.” [5] With using the fifth generation programming languages, the computers can have their own ability to think and their own inferences can be work out by using the programmed information in large databases. The dream of a robot with artificial intelligence and fuzzy logic came true by using this generation programming languages.

2.2 Brief history of C language

“C is a programming language which born at ‘AT & T’s Bell Laboratories’ of USA in 1972. It was written by Dennis Ritchie. This language was created for a specific purpose: to design the UNIX operating system (which is used on many computers). From the beginning, C was intended to be useful--to allow busy programmers to get things done.”[6]

After that, C began to be used by more and more people outside the Bell Laboratories because it is more efficient than other programming languages at that time. In the late 70’s, C took the dominant position of programming languages. “The committee formed by the American National Standards Institute (ANSI) approved a version of C in 1989 which is known as ANSIC. With few exceptions, every modern C compiler has the ability to adhere to this standard. ANSI C was then approved by the International Standards Organization (ISO) in 1990.”[6]

There is something interesting about the name of C. It was named C because its predecessor was called B which was also developed by Ken Thompson of Bell Labs. [6]

2.3 Brief history of C++ language

“C++ was written by Bjarne Stroustrup at Bell Labs during 1983-1985. C++ is an extension of C. Prior to 1983, Bjarne Stroustrup added features to C and formed what he called ‘C with Classes’. He had combined the Simula's use of classes and object-oriented features with the power and efficiency of C. The term C++ was first used in 1983.” [7]

“C++ was designed for the UNIX system environment, it represents an enhancement of the C programming language and enables programmers to improve the quality of code produced, thus making reusable code easier to write.”[8]

2.4 Brief history of C# language

“The primary architects of C# were Peter Golde, Eric Gunnerson, Anders Hejlsberg, Peter Sollichy and Scott Wiltamuth. Of course, the principal designer of the C# language was Anders Hejlsberg, a lead architect at Microsoft.” [9]

C# was designed to be a pure object-oriented programming language. “C# debuted in the year 2000 at the Professional Developers Conference (PDC) where Microsoft founder Bill Gates was the keynote speaker. At the same time, Visual Studio .NET was announced.” [9]

2.5 Brief history of Java language

Java started to be developed in 1991 by James Gosling from Sun Microsystems and his team. The original version of Java is designed for programming home appliances.

In 1994, James Gosling started to make a connection between Java and internet. "In 1995, Netscape Incorporated released its latest version of the Netscape browser which was capable of running Java programs."[10]

The original name of Java is Oak. But it had to change its original name because Oak had been used by another programming language. The new name Java was inspired by a coffee bean.

"While Java is viewed as a programming language to design applications for the Internet, it is in reality a general all-purpose language which can be used independent of the Internet."[10]

3. Comparison of C, C++, C#, and Java from theoretical aspects

3.1 Data Types and Sizes

“Some common types of data types are used in the programming languages called as the primitive types like characters, integers, floating point numbers etc.”[11]

In C, C++, C# and Java, all variables must be declared before they are used, usually at the beginning of the function before any executable statements.

C

In C Language, There are four basic Date type.

Table 2. Basic data type in C language. [35]

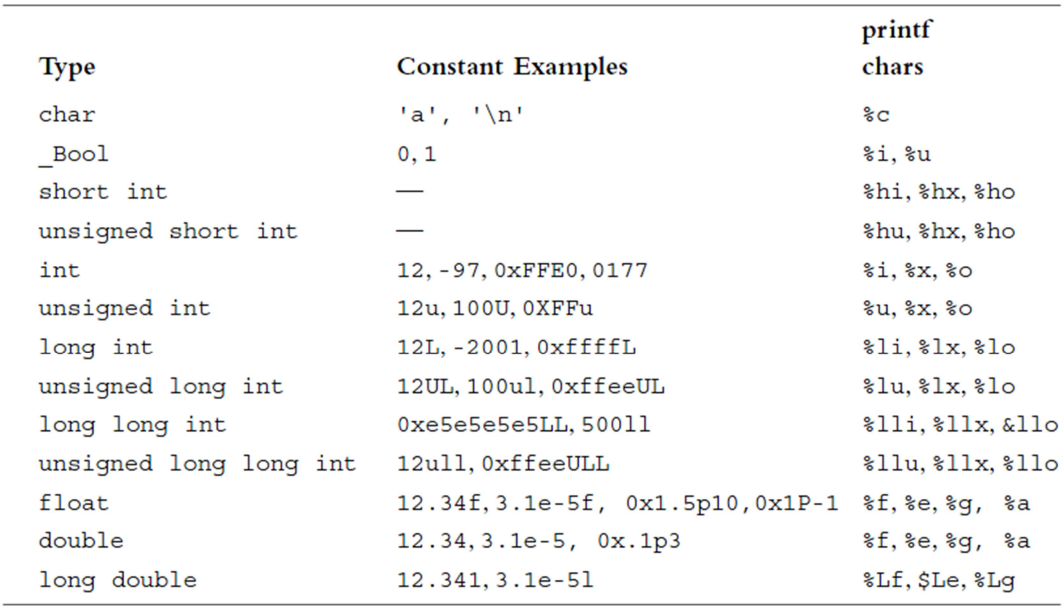
int	“an integer, typically reflecting the natural size of integers on the host machine”
float	“single-precision floating point”
double	“double-precision floating point”
char	“a single byte, capable of holding one character in the local character set”

There are five Type Specifiers in C programming [34]

a) long
b) long long
c) short
d) unsigned
e) signed

The table 3 summarizes the basic data types and qualifiers in C.

Table 3. The basic data types and qualifiers in C. [34]



Type	Constant Examples	printf chars
char	'a', '\n'	%c
_Bool	0, 1	%i, %u
short int	—	%hi, %hx, %ho
unsigned short int	—	%hu, %hx, %ho
int	12, -97, 0xFFE0, 0177	%i, %x, %o
unsigned int	12u, 100U, 0XFFu	%u, %x, %o
long int	12L, -2001, 0xffffL	%li, %lx, %lo
unsigned long int	12UL, 100ul, 0xffeeUL	%lu, %lx, %lo
long long int	0xe5e5e5e5LL, 500ll	%lli, %llx, &llo
unsigned long long int	12ull, 0xffeeULL	%llu, %llx, %llo
float	12.34f, 3.1e-5f, 0x1.5p10, 0x1P-1	%f, %e, %g, %a
double	12.34, 3.1e-5, 0x.1p3	%f, %e, %g, %a
long double	12.341, 3.1e-5l	%Lf, $Le, %Lg

The table 4 lists all the reserved keywords in C language. “These keywords cannot be abbreviated, used as variable names, or used as any other type of identifiers.”[36]

Table 4. Reserved keywords in C language. [36]

auto	else	long	switch
break	enum	register	typedef
case	extern	return	union
char	float	short	unsigned
const	for	signed	void
continue	goto	sizeof	volatile
default	if	static	while
do	int	struct	_Packed
double			

In C, every data type such as a character, integer, or floating-point number has a range of values associated with it. The range is decided by the amount of storage that is allocated to store a particular type of data in the memory of the computer. It depends on the computer you’re running. This feature for C language is called “machine-dependent”. For example, an integer might take up 32 bits on your computer, or perhaps it might be stored in 64 bits on another computer. Don’t write any program that assumes the size of the data types in C. [35]

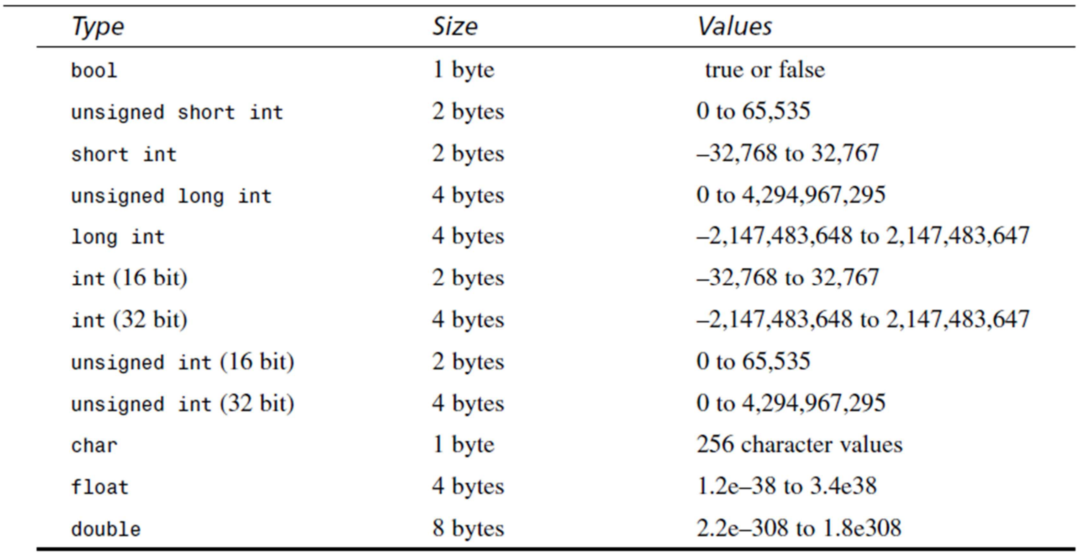
C++

In C++, the basic data type is almost the same as C.

In C++, there is one more type Booleans. A Boolean, bool, can have one of the two values true or false. A Boolean is used to express the results of logical operations.

The table 5 summarizes the basic data types and qualifiers in C++.

Table 5. The basic data types and qualifiers in C++. [37]



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 32 keywords both in C and C++ as shown in table 6:

Table 6. Keyword in both C and C++.

auto	else	long	switch
break	enum	register	typedef
case	extern	return	union
char	float	short	unsigned
const	for	signed	void
continue	goto	sizeof	volatile
default	if	static	while
do	int	struct	double

Only keyword _Packed is not used in C++ any more.

There are 30 keywords that are not in C, but they are used in C++.

Table 7. New keyword in C++.

asm	dynamic_cast	namespace	reinterpret_cast
bool	explicit	new	static_cast
catch	false	operator	template
class	friend	private	this
const_cast	inline	public	throw
delete	mutable	protected	true
try	typeid	typename	using
virtual	wchar_t		

C#

The Boolean type in C# has a little difference between that in C++. In C#, the Boolean type can only have two values: true or false. But in C++, the Boolean type can also have 0 as the value which means false and anything else means true. [39]

Integral Types in c# is also different from C and C++. In C and C++, the integral type is only one type. But "in C#, an integral is a category of types. They are whole numbers, either signed or unsigned, and the char type. The char type is a Unicode character, as defined by the Unicode Standard." Table 8 shows the integral types, their size, and range. [39]

In C#, the long data type is 64 bits, while in C++, it is 32 bits.

Table 8. Integral types, size, and range. [39]

Type	**Size (in bits)**	**Range**
sbyte	8	-128 to 127
byte	8	0 to 255
short	16	-32768 to 32767
ushort	16	0 to 65535
int	32	-2147483648 to 2147483647
uint	32	0 to 4294967295
long	64	-9223372036854775808 to 9223372036854775807
ulong	64	0 to 18446744073709551615
char	16	0 to 65535

"The floating point type in C# is either a float or double, which the same as that in C and C++ is. But there is one more data type decimal that should be used when representing financial or money values.” [39] Table 9 shows the floating point and decimal types, their size, precision, and range.

Table 9. Float types and Decimal type. [39]

Type	**Size (in bits)**	**precision**	**Range**
float	32	7 digits	1.5 x 10-45 to 3.4 x 1038
double	64	15-16 digits	5.0 x 10-324 to 1.7 x 10308
decimal	128	28-29 decimal places	1.0 x 10-28 to 7.9 x 1028

There are much more key words in C# than those in C and C++. There are total 87 reserved words in C#. Table 10 shows all the keywords in C#.

Table 10. C# keywords. [38]

C# keywords / C# reserved words			
abstract	as	base	bool
break	by [2]	byte	case
catch	char	checked	class
const	continue	decimal	default
delegate	do	double	descending [2]
explicit	event	extern	else
enum	false	finally	fixed
float	for	foreach	from [2]
goto	group [2]	if	implicit

in	int	interface	internal
into[2]	is	lock	long
new	null	namespace	object
operator	out	override	orderby[2]
params	private	protected	public
readonly	ref	return	switch
struct	sbyte	sealed	short
sizeof	stackalloc	static	string
select[2]	this	throw	true
try	typeof	uint	ulong
unchecked	unsafe	ushort	using
var[2]	virtual	volatile	void
while	where[2]	yield[1]	
[1, 2] These are not actually keywords, thus (unlike actual keywords) it is possible to define variables and types using these names, but they act like keywords in certain new language constructs introduced in C# 2.0[(1)] and 3.0[(2)].			

Java

The Boolean type in Java is as the same as that in C#. There are only two values available for Boolean type, true and false.

Integral Types in Java is different from C and C++ and it is similar to C#. In C#, an integral is a category consists of 9 data types. In Java, there are five data types that can have the integral value. There is no sbyte, uint, ulong data type in Java any more. Table 11 shows Java Primitive Data Types. [11]

Table 11. Java Primitive Data Types. [11]

Data Type	**Description**	**Size**	**Default Value**
boolean	true or false	1-bit	false
char	Unicode Character	16-bit	\u0000
byte	Signed Integer	8-bit	(byte) 0
short	Signed Integer	16-bit	(short) 0
int	Signed Integer	32-bit	0
long	Signed Integer	64-bit	0L
float	Real number	32-bit	0.0f
double	Real number	64-bit	0.0d

Java has two kinds of floating-point numbers: float and double. The default type when you write a floating-point literal is double. Table 12 shows Java floating-point.

Table 12. Java floating-point. [11]

Type	Size		Range	Precision
name	bytes	bits	approximate	in decimal digits
float	4	32	+/- 3.4 * 1038	6-7
double	8	64	+/- 1.8 * 10308	15

There are total 50 keywords in Java. Some keywords only appear in Java, such as synchronized, instanceof and strictfp. Table 13 lists all the keywords in Java.

Table 13. Key words in Java. [12]

abstract	continue	For	new
switch	assert	default	goto
package	synchronized	boolean	do
if	private	this	break
double	implements	protected	throw
byte	else	import	public
throws	case	enum	instanceof
return	transient	catch	extends
int	short	try	char
final	interface	static	void
class	finally	long	
volatile	const	float	native
super	while		

3.2 String type

In C language, there is no string type. The char type of array is used instead of String type.

In C++ language, there are two types of strings, C-style strings, and C++-style strings. “A C-style strings, which consists of an array of characters terminated by the null character '\0', and which have properties over and above those of an ordinary array of characters, as well as a whole library of functions for dealing with strings represented in this form. Its header file is ‘cstring’. A C++ style string is a ‘class’ data type. The objects of C++ style string are instances of the C++ ‘string’ class. There is a library of C++ string functions as well, available by including the ’string’ header file.” [14]

“In C# and Java, the data type String is treated as reference type. Instance of Strings are treated as (immutable) objects in both languages, but support for string literals provides a specialized means of constructing them. C# also allows verbatim strings for quotation without escape sequences, which also allow newlines.”[15]

3.3 Struct

The structure in C language has two differences from structure in C++ language.

1. C Structure can only contain data items while C++ structure can contain both data and functions.
2. In C language, using 'struct' keyword is necessary to create a variable. But in C++ Only Structure name is needed to identify a variable.

The difference between structure in C++ and C# is significant. “In C++, a struct is exactly like a class, except that the default inheritance and default access are public rather than private.” On the contrary, in C#, structs have significant difference from classes. “Structs in C# are designed to encapsulate lightweight objects. They are value types (not reference types), so they're passed by value. In addition, they have limitations that do not apply to classes. For example, they are sealed, which means they cannot be derived from or have any base class other than System.ValueType, which is derived from Object. Structs cannot declare a default (parameterless) constructor.” [16]

On the other hand, struct is very suitable for using a class with very small instance if you don't care that “the struct is sealed” and you don't care “value semantics”. “Structs are more efficient than classes so they're perfect for the creation of lightweight objects.”[16]

In C#, a struct can also be a value type as an as part of object or store in an array.
Java does not have struct value type.

3.4 Inherence

Inherence means creating a new class derives from the existing class. The derived class inherits all the variables and methods from base class. Overriding methods and properties can extend the functionality of base class.

C language does not support inheritance because it is not object-oriented language.

Neither C# nor Java allows multiple inheritance which means each class can only inherit from only one class. However, both C# and Java supports multiple implementations of interfaces.

C++ supports multiple inheritance of class. In C++, classes and structs are almost identical. “Furthermore, C# structs do not support inheritance, and do not support explicit default constructors (one is provided by default)”. [13] On the contrary, C++ structs support multiple inheritances.

3.5 Array

“In C and C++, each sub-array of a multidimensional array must have the same dimensions. In Java and C# arrays do not have to be uniform because jagged arrays can be created as one-dimensional arrays of arrays. In a jagged array the contents of the array are arrays which may hold instances of a type or references to other arrays.”[18]

In C#，multidimensional and jagged arrays have the significant difference. “A multidimensional array is akin to a multidimensional array in C or C++ that is a contiguous block containing members of the same type. A jagged array is akin to an array in Java which is an array of arrays, meaning that it contains references to other arrays which may contain members of the same type or other arrays depending on how many levels the array has.” [18]

3.6 Reference and Value Types

“Primitive types are the basic types of data, such as byte, short, int, long, float, double, boolean, char. Reference types are any instantiable class as well as arrays, such as String, Scanner, Random, Die, int[], String[], etc.”[13]

Both in C# and Java, the idea of reference type is supported. In C#, value types and reference types which have significant difference. The situation of primitive types and reference types is exactly the same as that in C#. Simple types (int, long, double, and so on) and structs are value types in C#. Both value types in C# and the primitive types in Java as shown in table 7 hold their value on the stack, like variables in C++, unless they are embedded within a reference type. All classes are reference types in both C# and Java. “Value types and primitive type in C# and Java Reference type variables sit on the stack, but they hold the address of an object on the heap, much like pointers in C++. Value types are passed to methods by value (a copy is made), while reference types are effectively passed by reference.”[16]

Neither C nor C++ has value types and reference types. In C++, basic data type, classes or user-defined value types have the same access to memory.

3.7 Pointer

“A pointer is a variable containing the address of another variable.”[18]

In C and C++, pointer can be used as arrays, Strings, writeable function parameters and optimization.

In C# pointer can only be declared to hold the memory address of value types and arrays.

For safety reasons, Java does not support pointers or pointer-arithmetic to be used at all.

3.8 Partial classes

"C# allows a class definition to be split across several source files using a feature called partial classes. Each part must be marked with the keyword partial." [23]

This function is not available in C, C++, and Java. In C++ and Java, an error about redefining already-defined classes was given. However, in C++ and Java, using inheritance can reach the same goal.

3.9 Compiler Technology

There are two ways to implement programming languages: Interpretation and compilation.

"Compilation is a program that reads your program and translates it into machine code. Then the computer obeys the machine code. Interpretation is a program that looks at each line of your program in turn, works out what it means, obeys it, and then goes onto the next line." In other word, Compilation and interpretation like two ways of reading. Compilation is like translating an English article into Chinese totally and then reading the Chinese article. Interpretation is like translate the every word in the article from English to Chinese one by one by using Chinese dictionary. [29]

Java is a dynamically compiled language, while C and C++ are statically compiled languages.

The difference between dynamically compiled languages and statically compiled languages is significant. "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. 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 are two generations of JVM (Java Virtual Machine). The first generation of JVM is slow because “the JVM interpreted the byte codes rather than compiling them to machine code and executing the machine code directly.” This is of course not good approach because “the system spent more time executing the interpreter than the program it was supposed to be running”. The second generation of JVM improves the performance by using “just-in-time (JIT) compilers”. “Strictly defined, a JIT-based virtual machine converts all byte codes into machine code before execution, but does so in a lazy fashion: The JIT only compiles a code path when it knows that code path is about to be executed (hence, the name, just-in-time compilation). This approach allows the program to start up more quickly, as a lengthy compilation phase is not needed before any execution can begin.” [30]

The Compilation Process of Java is quite different between that of C/C++. We can call Java the "semi-interpreted" language because “Java programs execute in the Java Virtual Machine (or JVM), which makes it an interpreted language.” “On the other hand Java unlike pure interpreted languages passes through an intermediate compilation step.” Java code compile to the “intermediate byte code” that run in the virtual machine rather than “native code that the operating system executes on the CPU”. [31]

The figure shows the Compilation Process of Java.

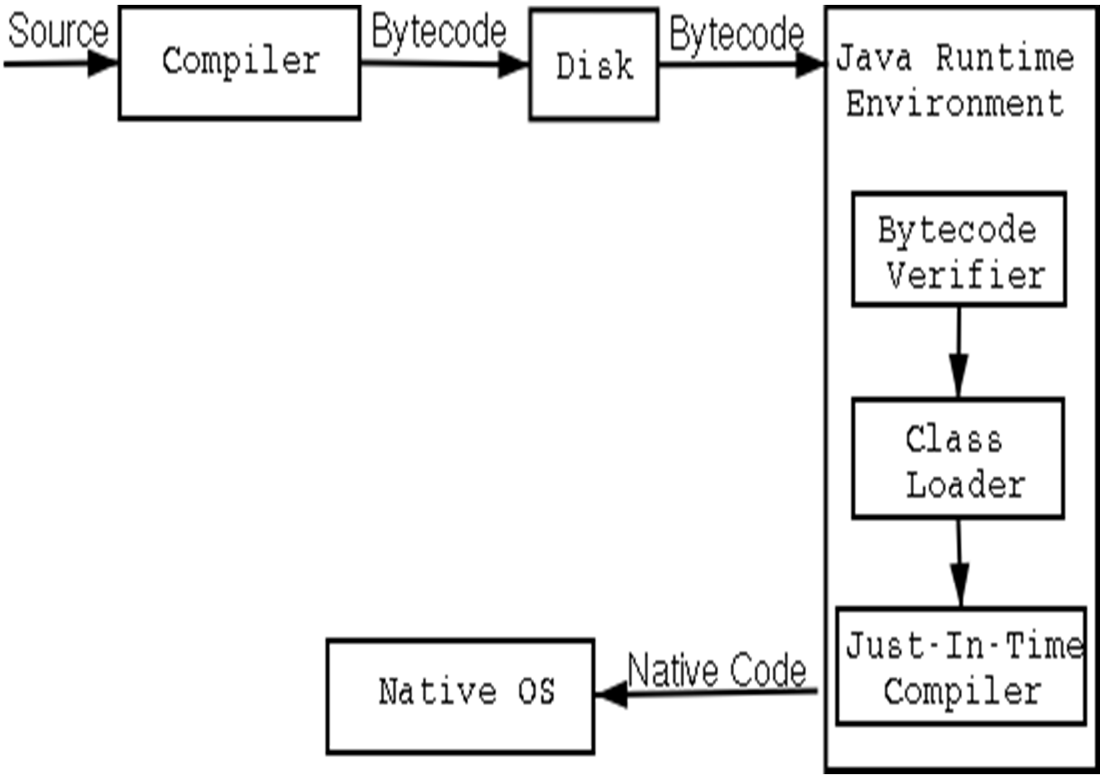


Figure 1. Java Compile/Execute Path. [31]

There are four main steps of compilation in C and C++ as figure show below.

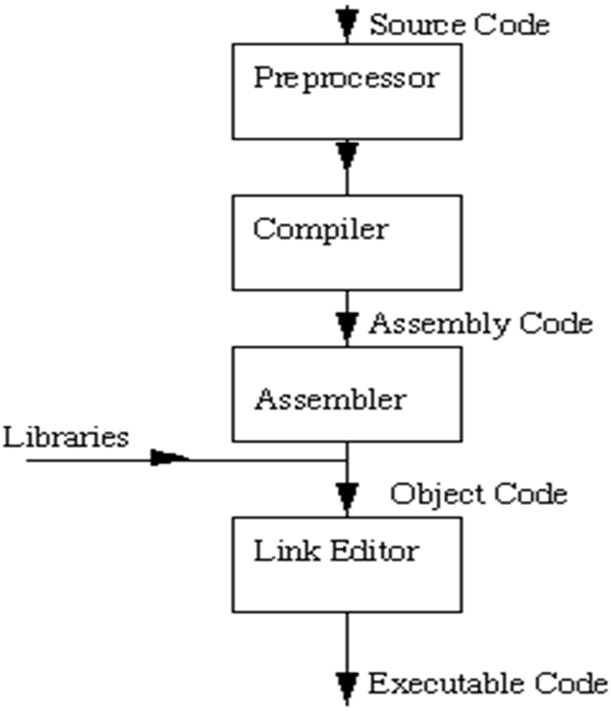


Figure 2. C and C++ Compilation Model. [32]

The Preprocessor is responsible for two functions: “Removing comments” and “Interpreting special preprocessor directives denoted by #.” [32]

The output of the Preprocessor becomes the input of the C compiler. And then “C compiler translates the received code to the assembly code”. [32]

“Assembler creates object code. On a UNIX system you may see files with a .o suffix (.OBJ on MSDOS) to indicate object code files.” [32]

Link Editor: “If a source file references library functions or functions defined in other source files the link editor combines these functions (with main()) to create an executable file. External Variable references resolved here also.” [32]

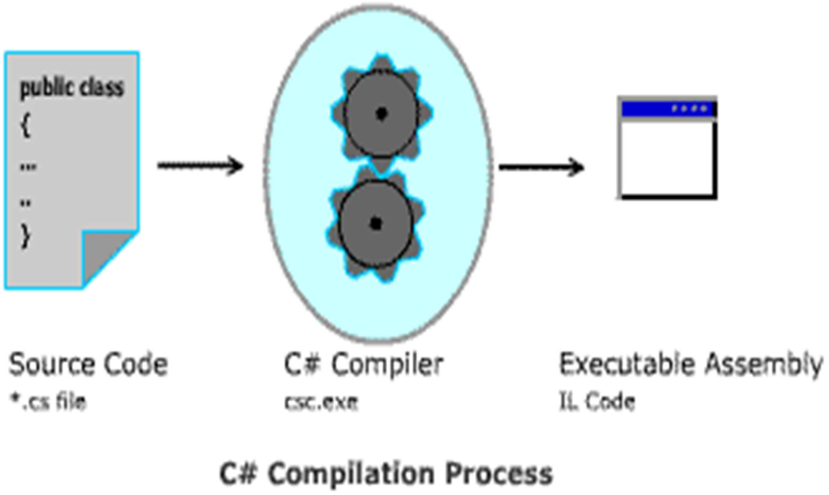


Figure 3. C# compilation Process. [32]

The first step of C# compilation is that the source code is translated to the “Microsoft Intermediate Language (MSIL) code”. “It packages the MSIL code as a Win32 executable file with some extended features. The header table of the executable has been expanded to accommodate additional metadata about the assembly. Also the code contained within it is not assembly language but its MSIL.”[32]

4 Experiment on the benchmark of C, C++, C#, and Java

This chapter presented the recording data from running the benchmark of C, C++, C# and Java. And the recording data was analyzed as well.

The definition of benchmark: “in computing, a benchmark is the act of running a computer program, a set of programs, or other operations, in order to assess the relative performance of an object, normally by running a number of standard tests and trials against it.”[40]

4.1 Object

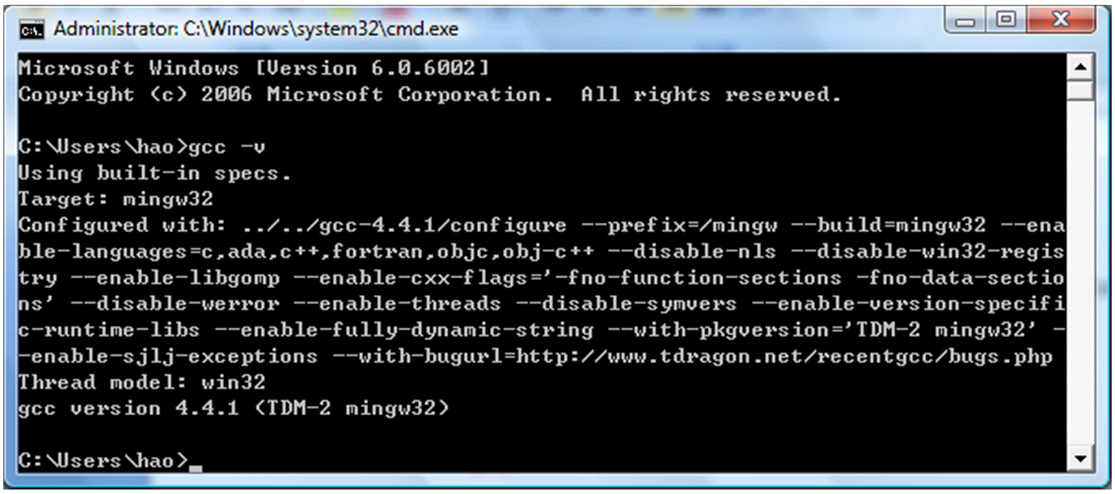
Implement the benchmark testing to compare the actual performance of the four chosen programming languages C, C++, C# and Java.

4.2 Test System

- Processor: Intel(R) Core(TM)2 Duo CPU P7350 @ 2.00GHZ
- Memory: 3072 MB DDR2 800MHZ(PC6400)
- Hard Disk: TOSHIBA mk3252GSX 320G
- Chipset: Intel Mobile 4 Series Chipset
- Operating System: Microsoft Windows Vista Home Premium (SP2)

C compiler version:

- gcc version “4.4.1”

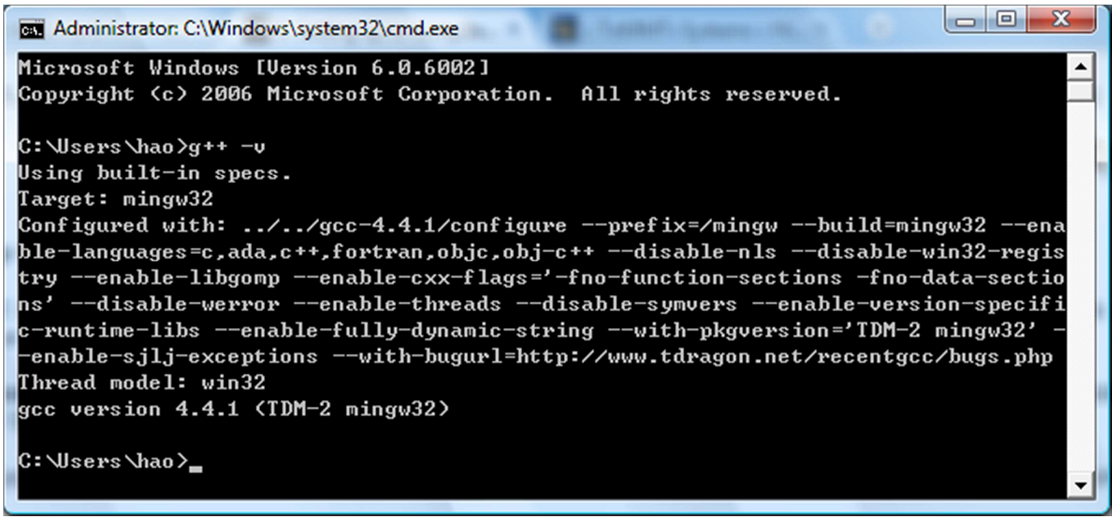


```
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\hao>gcc -v
Using built-in specs.
Target: mingw32
Configured with: ../../gcc-4.4.1/configure --prefix=/mingw --build=mingw32 --ena
ble-languages=c,ada,c++,fortran,objc,obj-c++ --disable-nls --disable-win32-regis
try --enable-libgomp --enable-cxx-flags='-fno-function-sections -fno-data-sectio
ns' --disable-werror --enable-threads --disable-symvers --enable-version-specifi
c-runtime-libs --enable-fully-dynamic-string --with-pkgversion='TDM-2 mingw32' -
-enable-sjlj-exceptions --with-bugurl=http://www.tdragon.net/recentgcc/bugs.php
Thread model: win32
gcc version 4.4.1 (TDM-2 mingw32)

C:\Users\hao>_
```

Figure 4. C compiler version.

C++ compiler version:

- g++ version “4.4.1”

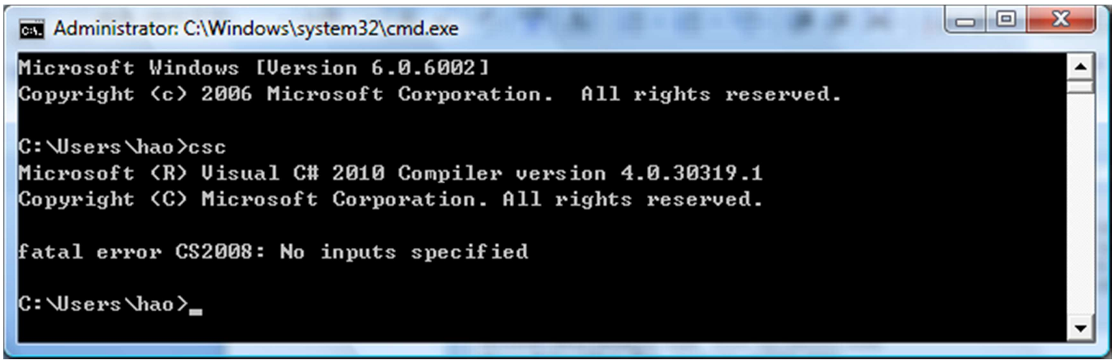


```
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\hao>g++ -v
Using built-in specs.
Target: mingw32
Configured with: ../../gcc-4.4.1/configure --prefix=/mingw --build=mingw32 --ena
ble-languages=c,ada,c++,fortran,objc,obj-c++ --disable-nls --disable-win32-regis
try --enable-libgomp --enable-cxx-flags='-fno-function-sections -fno-data-sectio
ns' --disable-werror --enable-threads --disable-symvers --enable-version-specifi
c-runtime-libs --enable-fully-dynamic-string --with-pkgversion='TDM-2 mingw32' -
-enable-sjlj-exceptions --with-bugurl=http://www.tdragon.net/recentgcc/bugs.php
Thread model: win32
gcc version 4.4.1 (TDM-2 mingw32)

C:\Users\hao>_
```

Figure 5. C++ compiler version.

C# compiler version:

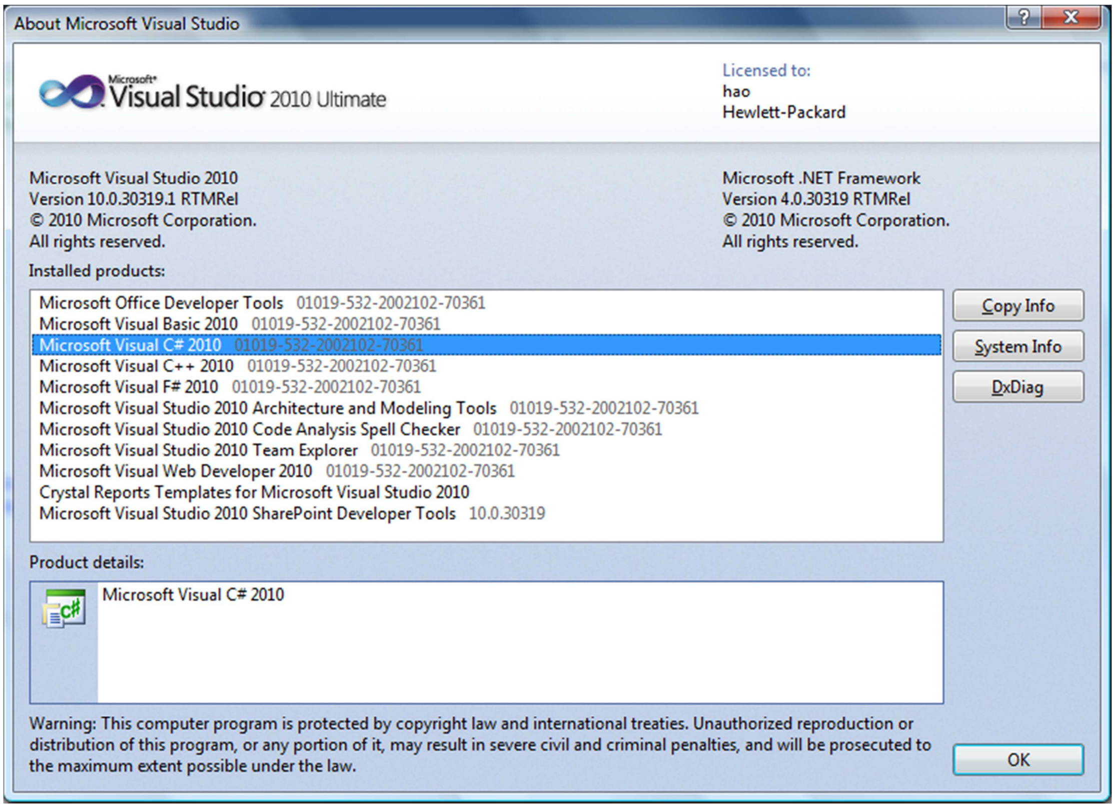
- Microsoft (R) Visual C# 2010 Compiler version 4.0.30319.1



```
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\hao>csc
Microsoft (R) Visual C# 2010 Compiler version 4.0.30319.1
Copyright (C) Microsoft Corporation. All rights reserved.

fatal error CS2008: No inputs specified

C:\Users\hao>_
```

Figure 6. C# compiler version.

- Microsoft (R) .NET Framework version 4.0.30319 RTMRel



About Microsoft Visual Studio
Microsoft Visual Studio 2010 Ultimate

Licensed to:
hao
Hewlett-Packard

Microsoft Visual Studio 2010
Version 10.0.30319.1 RTMRel
© 2010 Microsoft Corporation.
All rights reserved.

Microsoft .NET Framework
Version 4.0.30319 RTMRel
© 2010 Microsoft Corporation.
All rights reserved.

Installed products:

Microsoft Office Developer Tools 01019-532-2002102-70361
Microsoft Visual Basic 2010 01019-532-2002102-70361
Microsoft Visual C# 2010 01019-532-2002102-70361
Microsoft Visual C++ 2010 01019-532-2002102-70361
Microsoft Visual F# 2010 01019-532-2002102-70361
Microsoft Visual Studio 2010 Architecture and Modeling Tools 01019-532-2002102-70361
Microsoft Visual Studio 2010 Code Analysis Spell Checker 01019-532-2002102-70361
Microsoft Visual Studio 2010 Team Explorer 01019-532-2002102-70361
Microsoft Visual Web Developer 2010 01019-532-2002102-70361
Crystal Reports Templates for Microsoft Visual Studio 2010
Microsoft Visual Studio 2010 SharePoint Developer Tools 10.0.30319

Product details:

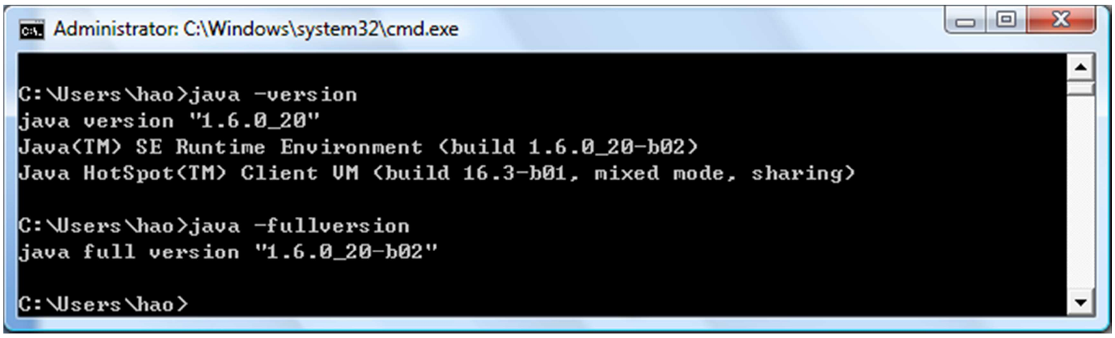
Microsoft Visual C# 2010

Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Figure 7. Microsoft (R) .NET Framework version.

Java Version:

- Java full version "1.6.0_20-b02"



```
C:\Users\hao>java -version
java version "1.6.0_20"
Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
Java HotSpot(TM) Client VM (build 16.3-b01, mixed mode, sharing)

C:\Users\hao>java -fullversion
java full version "1.6.0_20-b02"

C:\Users\hao>
```

Figure 8. Java compiler version.

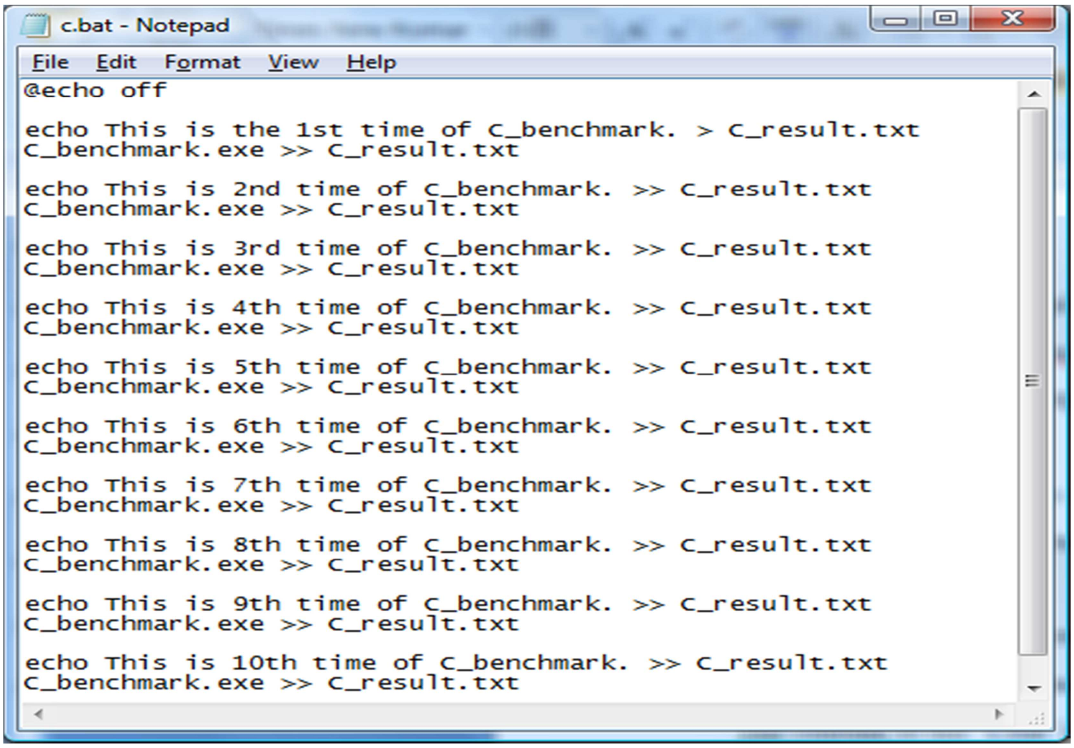
4.3 Source Code

The source code for the benchmark for C, C++, C# and Java is from Christopher W. Cowell-Shah http://www.cowell-shah.com/research/benchmark/code.

4.4 Implementation

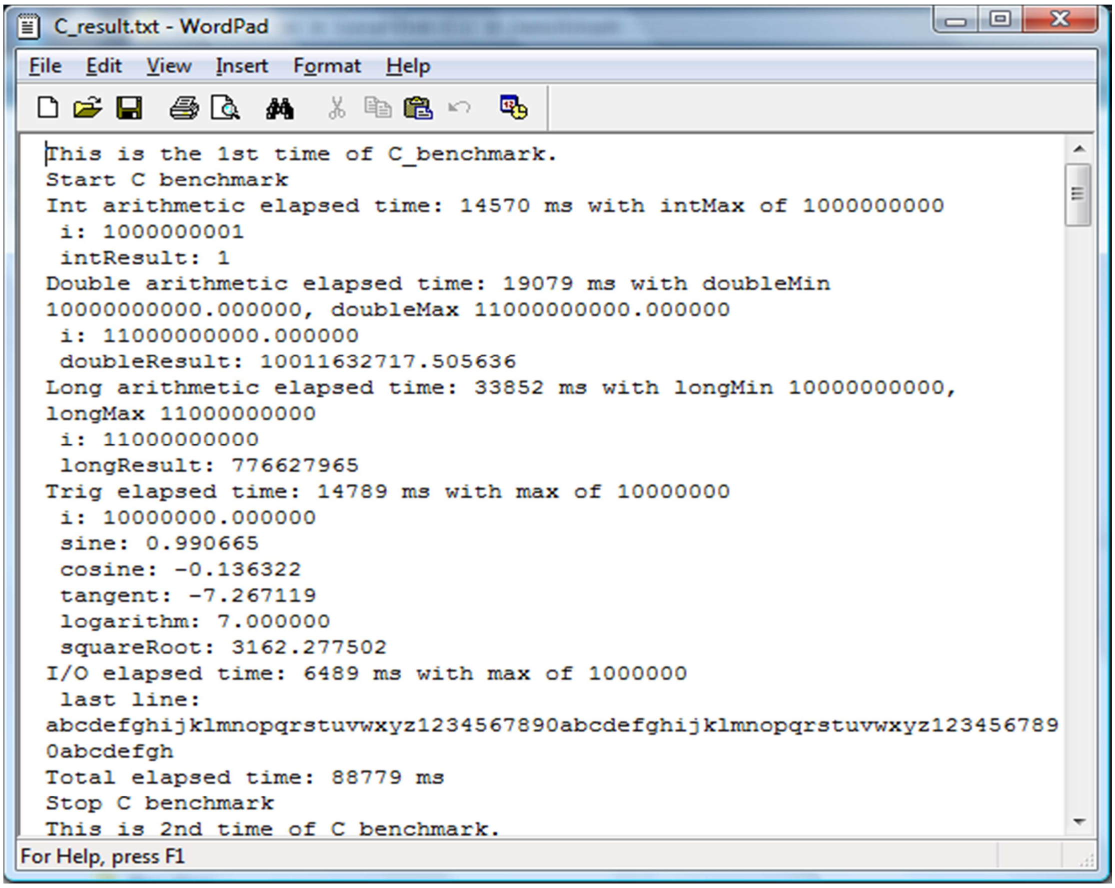
First, the codes of benchmark for four chosen programming languages were compiled to machine code and stored in the executable file. Then four executable bat files are created to run every executable files for 10 times. This is to give a better overall picture of performance for four chosen programming languages. All the output of the executable bat files is stored in the files and the files are used to analyze the performance of four chosen programming languages.

There are five parts in the benchmark of all four chosen programming languages.
Int arithmetic: Math benchmark using ints.
Double arithmetic: Math benchmark using doubles.
Long arithmetic: Math benchmark using longs.
Trig: Calculate sin, cos, tan, log, square root for all numbers up to a max.
I/O: Write max lines to a file, then read max lines back in from file.



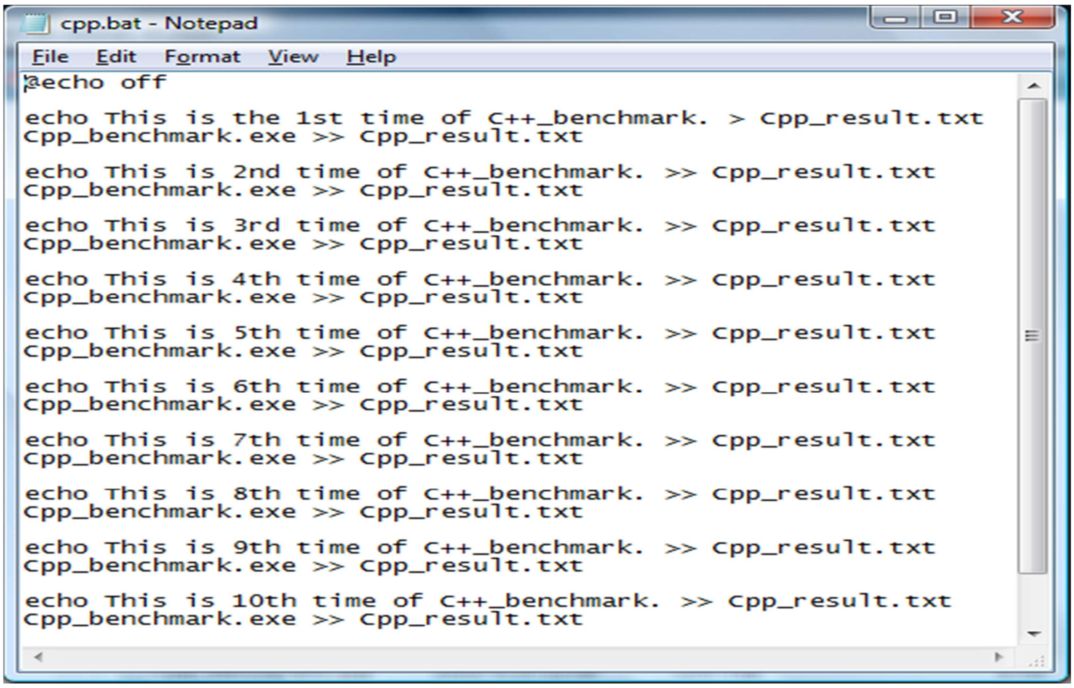
```
@echo off

echo This is the 1st time of C_benchmark. > C_result.txt
C_benchmark.exe >> C_result.txt

echo This is 2nd time of C_benchmark. >> C_result.txt
C_benchmark.exe >> C_result.txt

echo This is 3rd time of C_benchmark. >> C_result.txt
C_benchmark.exe >> C_result.txt

echo This is 4th time of C_benchmark. >> C_result.txt
C_benchmark.exe >> C_result.txt

echo This is 5th time of C_benchmark. >> C_result.txt
C_benchmark.exe >> C_result.txt

echo This is 6th time of C_benchmark. >> C_result.txt
C_benchmark.exe >> C_result.txt

echo This is 7th time of C_benchmark. >> C_result.txt
C_benchmark.exe >> C_result.txt

echo This is 8th time of C_benchmark. >> C_result.txt
C_benchmark.exe >> C_result.txt

echo This is 9th time of C_benchmark. >> C_result.txt
C_benchmark.exe >> C_result.txt

echo This is 10th time of C_benchmark. >> C_result.txt
C_benchmark.exe >> C_result.txt
```

Figure 9. Screenshot of executable bat file for C language.



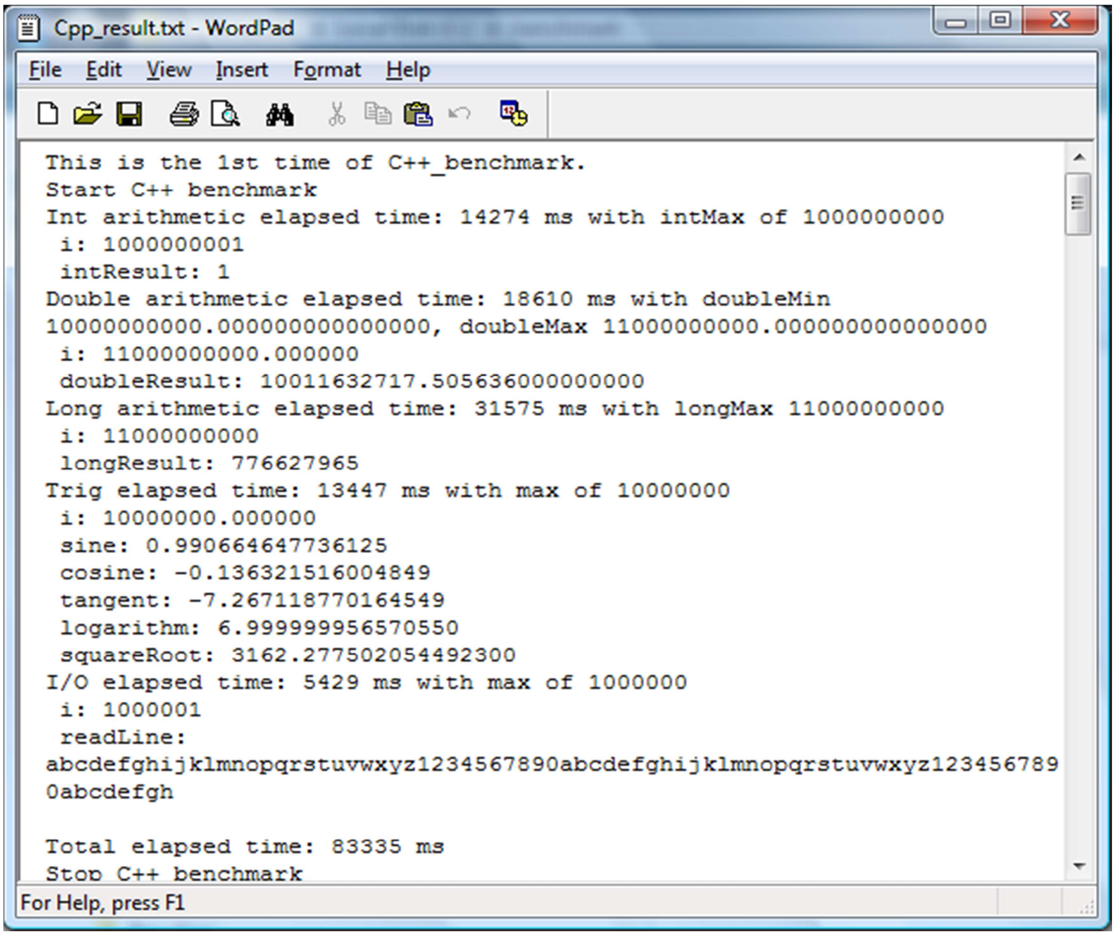
```
This is the 1st time of C_benchmark.
Start C benchmark
Int arithmetic elapsed time: 14570 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 19079 ms with doubleMin
10000000000.000000, doubleMax 11000000000.000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636
Long arithmetic elapsed time: 33852 ms with longMin 10000000000,
longMax 11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 14789 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990665
 cosine: -0.136322
 tangent: -7.267119
 logarithm: 7.000000
 squareRoot: 3162.277502
I/O elapsed time: 6489 ms with max of 1000000
 last line:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz123456789
0abcdefgh
Total elapsed time: 88779 ms
Stop C benchmark
This is 2nd time of C benchmark.
```

Figure 10. Screenshot of output of the executable bat files for C language.



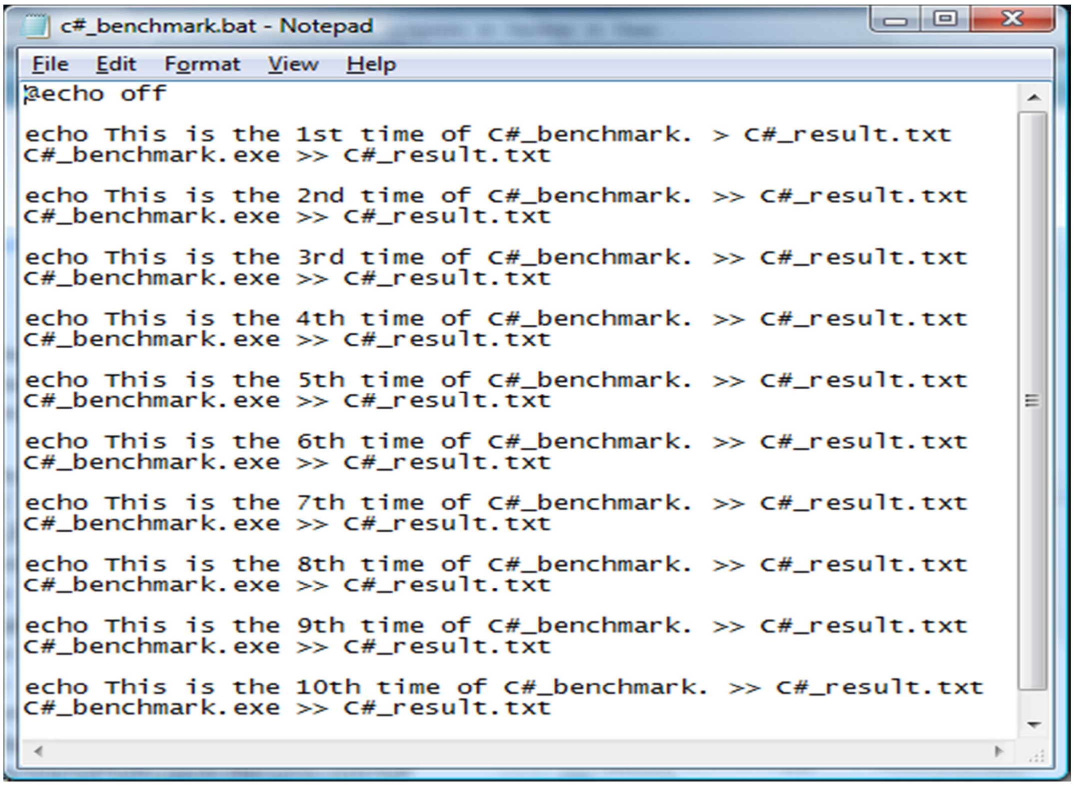
```
@echo off

echo This is the 1st time of C++_benchmark.  > Cpp_result.txt
Cpp_benchmark.exe >> Cpp_result.txt

echo This is 2nd time of C++_benchmark. >> Cpp_result.txt
Cpp_benchmark.exe >> Cpp_result.txt

echo This is 3rd time of C++_benchmark. >> Cpp_result.txt
Cpp_benchmark.exe >> Cpp_result.txt

echo This is 4th time of C++_benchmark. >> Cpp_result.txt
Cpp_benchmark.exe >> Cpp_result.txt

echo This is 5th time of C++_benchmark. >> Cpp_result.txt
Cpp_benchmark.exe >> Cpp_result.txt

echo This is 6th time of C++_benchmark. >> Cpp_result.txt
Cpp_benchmark.exe >> Cpp_result.txt

echo This is 7th time of C++_benchmark. >> Cpp_result.txt
Cpp_benchmark.exe >> Cpp_result.txt

echo This is 8th time of C++_benchmark. >> Cpp_result.txt
Cpp_benchmark.exe >> Cpp_result.txt

echo This is 9th time of C++_benchmark. >> Cpp_result.txt
Cpp_benchmark.exe >> Cpp_result.txt

echo This is 10th time of C++_benchmark. >> Cpp_result.txt
Cpp_benchmark.exe >> Cpp_result.txt
```

Figure 11. Screenshot of executable bat file for C++ language.



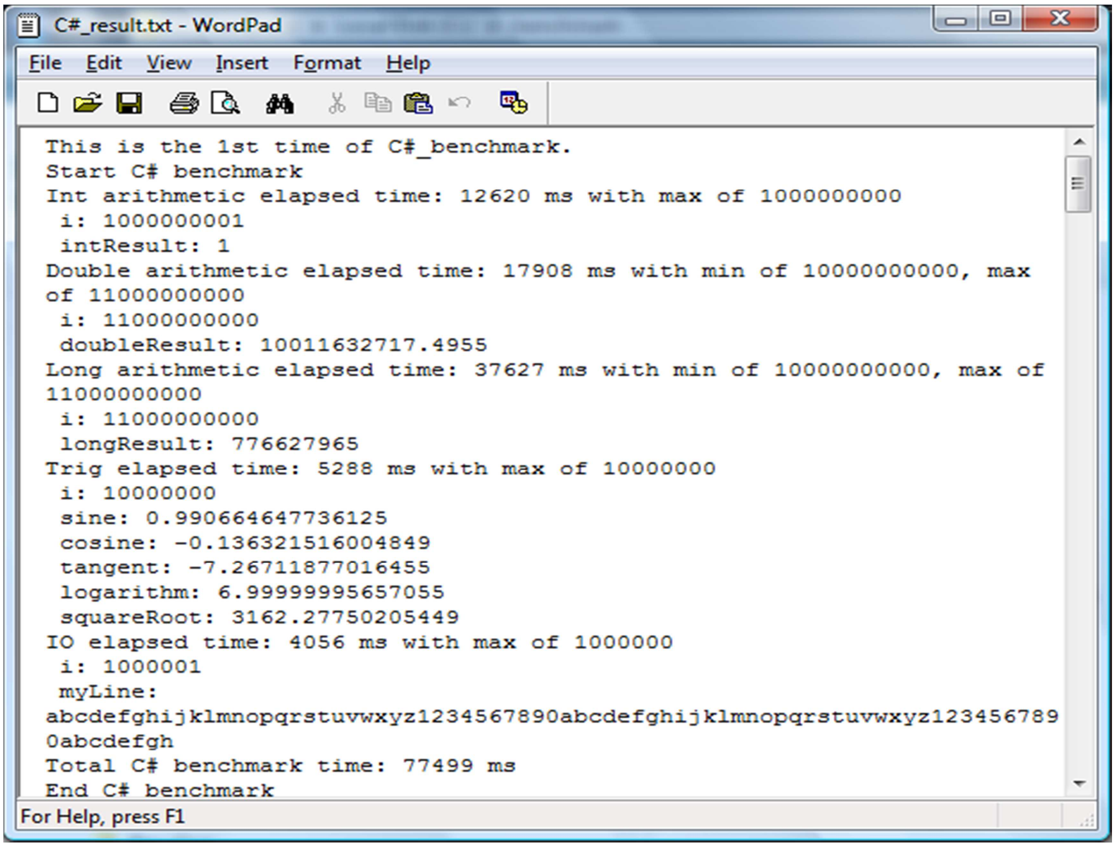
```
This is the 1st time of C++_benchmark.
Start C++ benchmark
Int arithmetic elapsed time: 14274 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18610 ms with doubleMin
10000000000.000000000000000, doubleMax 11000000000.000000000000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636000000000
Long arithmetic elapsed time: 31575 ms with longMax 11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13447 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.267118770164549
 logarithm: 6.999999956570550
 squareRoot: 3162.277502054492300
I/O elapsed time: 5429 ms with max of 1000000
 i: 1000001
 readLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz123456789
0abcdefgh

Total elapsed time: 83335 ms
Stop C++ benchmark
```

Figure 12. Screenshot of output of the executable bat files for C++ language.



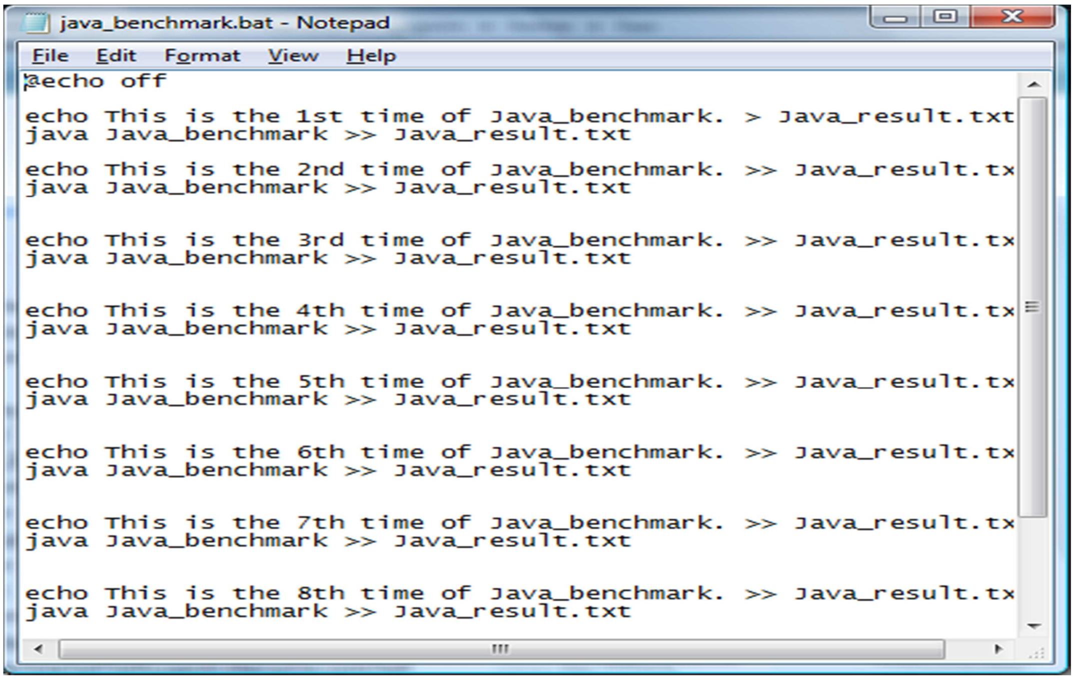
```
@echo off

echo This is the 1st time of C#_benchmark. > C#_result.txt
C#_benchmark.exe >> C#_result.txt

echo This is the 2nd time of C#_benchmark. >> C#_result.txt
C#_benchmark.exe >> C#_result.txt

echo This is the 3rd time of C#_benchmark. >> C#_result.txt
C#_benchmark.exe >> C#_result.txt

echo This is the 4th time of C#_benchmark. >> C#_result.txt
C#_benchmark.exe >> C#_result.txt

echo This is the 5th time of C#_benchmark. >> C#_result.txt
C#_benchmark.exe >> C#_result.txt

echo This is the 6th time of C#_benchmark. >> C#_result.txt
C#_benchmark.exe >> C#_result.txt

echo This is the 7th time of C#_benchmark. >> C#_result.txt
C#_benchmark.exe >> C#_result.txt

echo This is the 8th time of C#_benchmark. >> C#_result.txt
C#_benchmark.exe >> C#_result.txt

echo This is the 9th time of C#_benchmark. >> C#_result.txt
C#_benchmark.exe >> C#_result.txt

echo This is the 10th time of C#_benchmark. >> C#_result.txt
C#_benchmark.exe >> C#_result.txt
```

Figure 13. Screenshot of executable bat file for C# language.



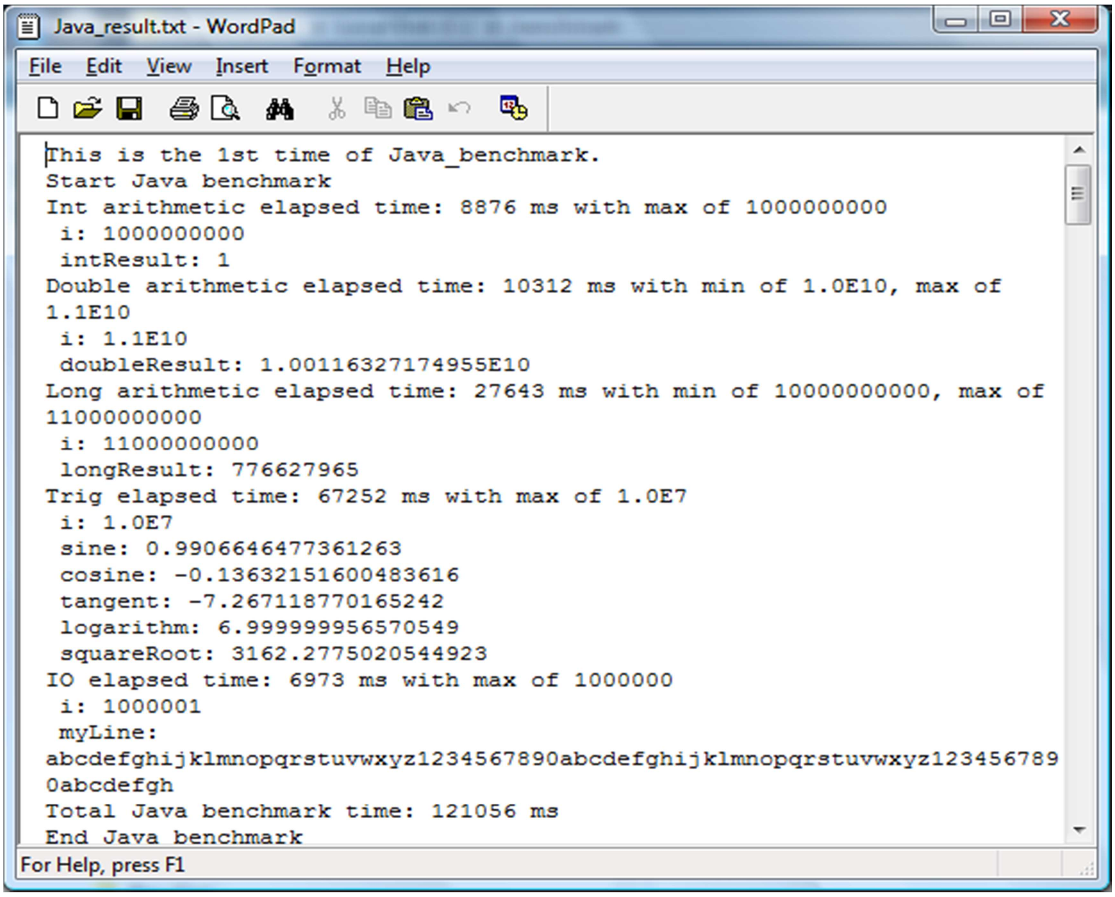
```
This is the 1st time of C#_benchmark.
Start C# benchmark
Int arithmetic elapsed time: 12620 ms with max of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 17908 ms with min of 10000000000, max
of 11000000000
 i: 11000000000
 doubleResult: 10011632717.4955
Long arithmetic elapsed time: 37627 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 5288 ms with max of 10000000
 i: 10000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.26711877016455
 logarithm: 6.99999995657055
 squareRoot: 3162.27750205449
IO elapsed time: 4056 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz123456789
0abcdefgh
Total C# benchmark time: 77499 ms
End C# benchmark
```

Figure 14. Screenshot of output of the executable bat files for C# language.



```
@echo off

echo This is the 1st time of Java_benchmark. > Java_result.txt
java Java_benchmark >> Java_result.txt

echo This is the 2nd time of Java_benchmark. >> Java_result.tx
java Java_benchmark >> Java_result.txt


echo This is the 3rd time of Java_benchmark. >> Java_result.tx
java Java_benchmark >> Java_result.txt


echo This is the 4th time of Java_benchmark. >> Java_result.tx
java Java_benchmark >> Java_result.txt


echo This is the 5th time of Java_benchmark. >> Java_result.tx
java Java_benchmark >> Java_result.txt


echo This is the 6th time of Java_benchmark. >> Java_result.tx
java Java_benchmark >> Java_result.txt


echo This is the 7th time of Java_benchmark. >> Java_result.tx
java Java_benchmark >> Java_result.txt


echo This is the 8th time of Java_benchmark. >> Java_result.tx
java Java_benchmark >> Java_result.txt
```

Figure 15. Screenshot of executable bat file for Java language.



```
This is the 1st time of Java_benchmark.
Start Java benchmark
Int arithmetic elapsed time: 8876 ms with max of 1000000000
 i: 1000000000
 intResult: 1
Double arithmetic elapsed time: 10312 ms with min of 1.0E10, max of
1.1E10
 i: 1.1E10
 doubleResult: 1.00116327174955E10
Long arithmetic elapsed time: 27643 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 67252 ms with max of 1.0E7
 i: 1.0E7
 sine: 0.9906646477361263
 cosine: -0.13632151600483616
 tangent: -7.267118770165242
 logarithm: 6.999999956570549
 squareRoot: 3162.2775020544923
IO elapsed time: 6973 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz123456789
0abcdefgh
Total Java benchmark time: 121056 ms
End Java benchmark
```

Figure 16. Screenshot of output of the executable bat files for Java language.

4.5 Results

The tables below show the results after running executable bat files for four chosen programming languages.

Table 14. C language Performance in ms.

	Int arithmetic	Double arithmetic	Long arithmetic	Trig	I/O	Total elapsed time
1st Time	14570	19079	33852	14789	6489	88779
2nd Time	14258	18673	33103	13526	7410	86970
3rd Time	14320	18798	32714	13572	6240	85644
4th Time	14211	18673	33072	13479	6084	85519
5th Time	14180	18642	33088	13494	6754	86158
6th Time	14258	18658	33103	13463	5990	85472
7th Time	14258	18658	33040	13494	5429	84879
8th Time	14211	18658	33056	13494	5367	84786
9th Time	14242	18674	33040	13479	5382	84817
10th Time	14227	18673	33041	13478	5367	84786
Average	14273.5	18718.6	33110.9	13626.8	6052.1	85781

Table 15. C++ performance in ms.

	Int arithmetic	Double arithmetic	Long arithmetic	Trig	I/O	Total elapsed time
1st Time	14274	18610	31575	13447	5429	83335
2nd Time	14258	18673	31933	13463	5445	83772
3rd Time	14242	18658	31699	13463	5756	83818
4th Time	14196	18673	32198	13479	5772	84318
5th Time	14352	18704	31668	13463	5382	83569
6th Time	14274	18657	31684	13447	5834	83896
7th Time	14258	18642	31590	13463	5382	83335
8th Time	14336	18673	32323	13494	5398	84224
9th Time	14289	18658	31574	13463	5429	83413
10th Time	14274	18642	31575	13447	5803	83741
Average	14275.3	18659	31781.9	13462.9	5563	83742.1

Table 16. C# performance in ms.

	Int arithmetic	Double arithmetic	Long arithmetic	Trig	I/O	Total elapsed time
1st Time	12620	17908	37627	5288	4056	77499
2nd Time	12589	17940	38313	5304	5382	79528
3rd Time	12636	17971	37658	5319	6708	80292
4th Time	12620	17955	37611	5397	3978	77561
5th Time	12573	17908	38188	5288	3400	77357
6th Time	12589	17924	38188	5288	3572	77561
7th Time	12589	17908	38188	5304	3369	77358
8th Time	12589	17908	38204	5304	4789	78794
9th Time	12604	17893	37596	5304	3946	77343
10th Time	12604	17893	38173	5288	3400	77358
Average	12601.3	17920.8	37974.6	5308.4	4260	78065.1

Table 17. Java performance in ms.

	Int arithmetic	Double arithmetic	Long arithmetic	Trig	I/O	Total elapsed time
1st Time	8876	10312	27643	67252	6973	121056
2nd Time	8939	10343	27752	67205	6038	120277
3rd Time	8892	10281	27690	67220	6162	120245
4th Time	8908	10296	27628	67173	6240	120245
5th Time	8907	10312	27643	67158	6240	120260
6th Time	8908	10311	27659	67455	5850	120183
7th Time	8970	10374	27924	67533	5444	120245
8th Time	8907	10312	27705	67704	5476	120104
9th Time	8939	10343	27752	67798	5507	120339
10th Time	8923	10343	27768	67517	7051	121602
Average	8916.9	10322.7	27716.4	67401.5	6098.1	120455.6

4.6 Analysis

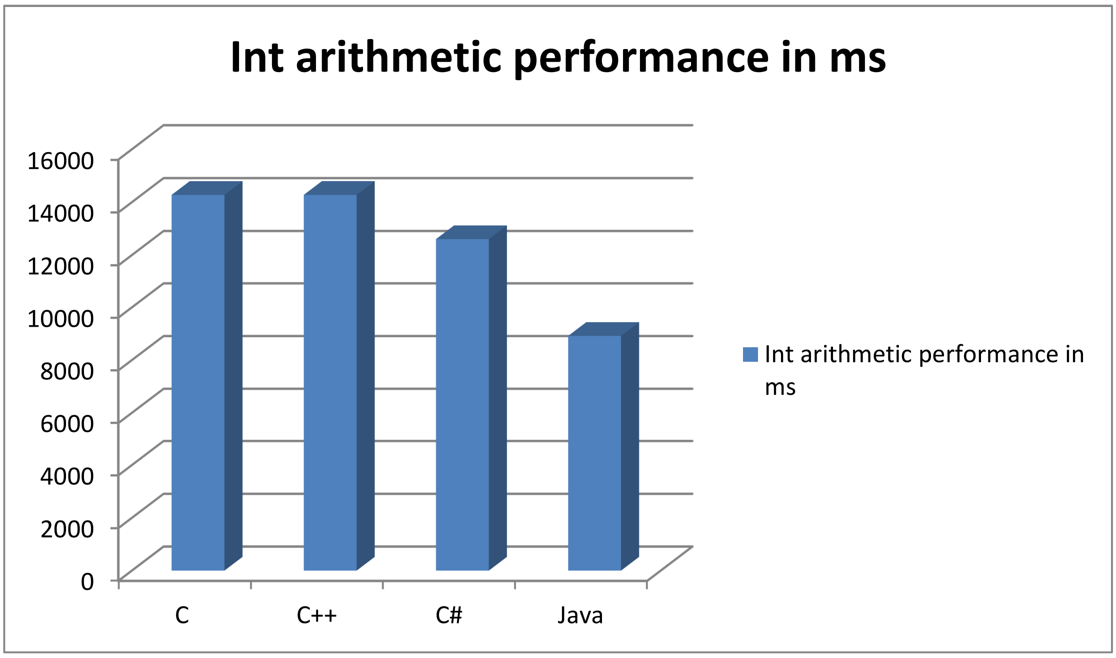


Figure 17. Int arithmetic performances in ms.

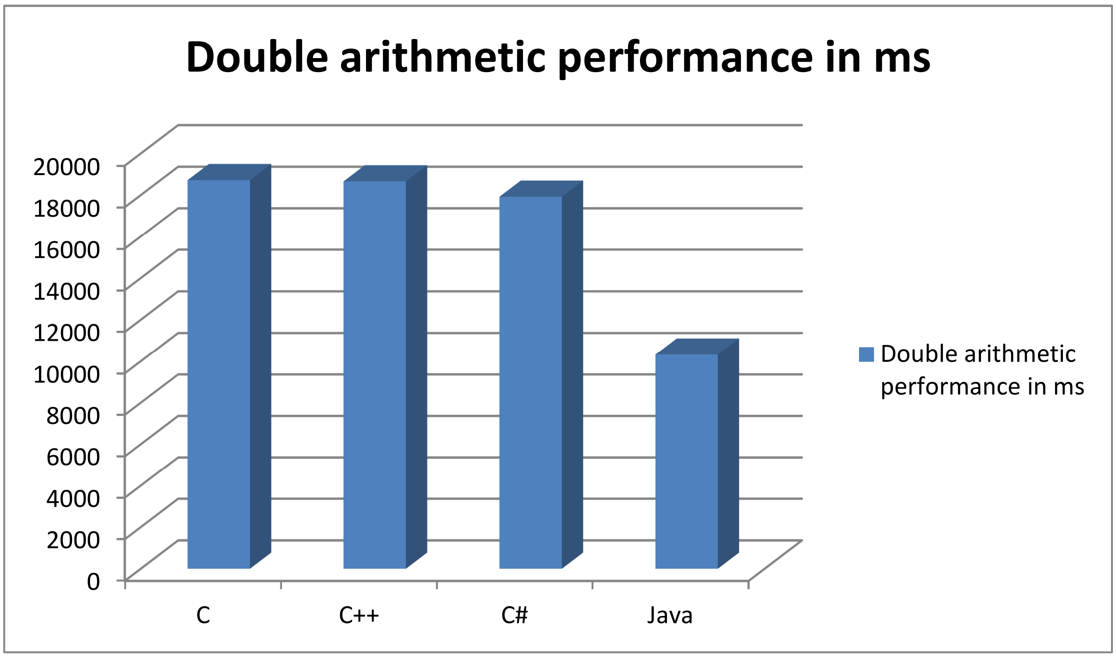


Figure 18. Double arithmetic performances in ms.

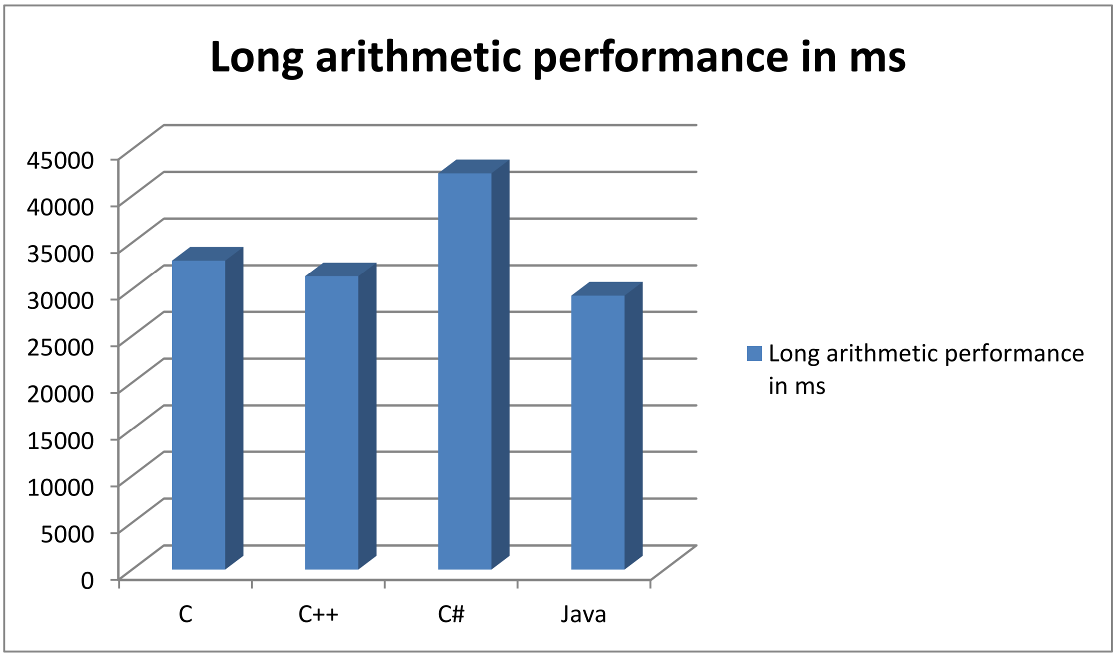


Figure 19. Long arithmetic performances in ms.

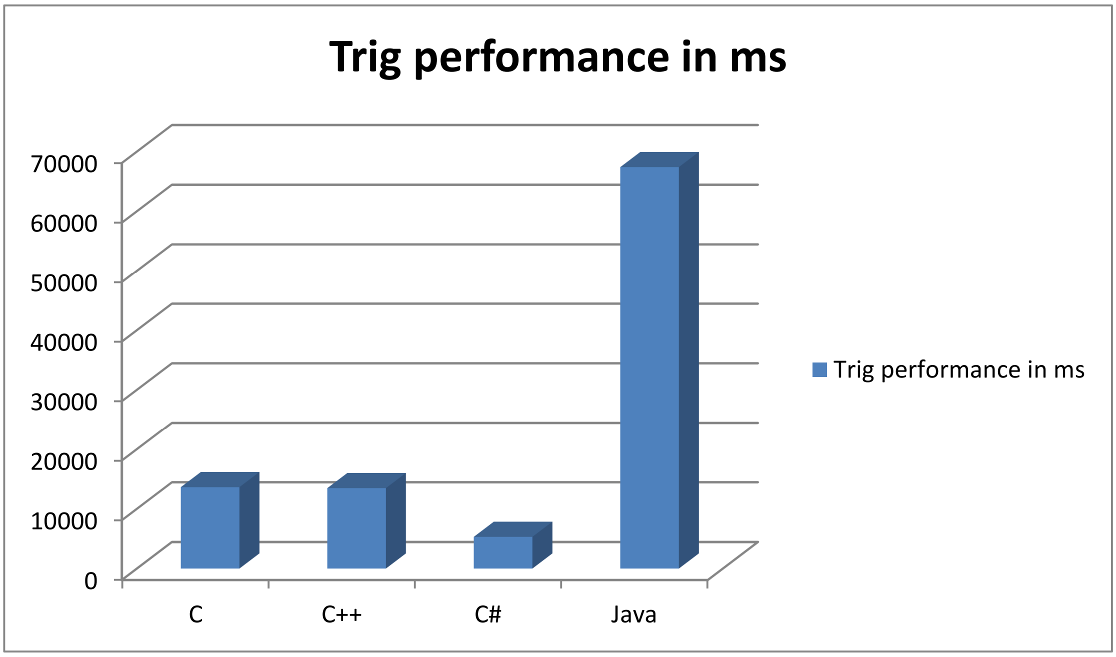


Figure 20. Trig arithmetic performances in ms.

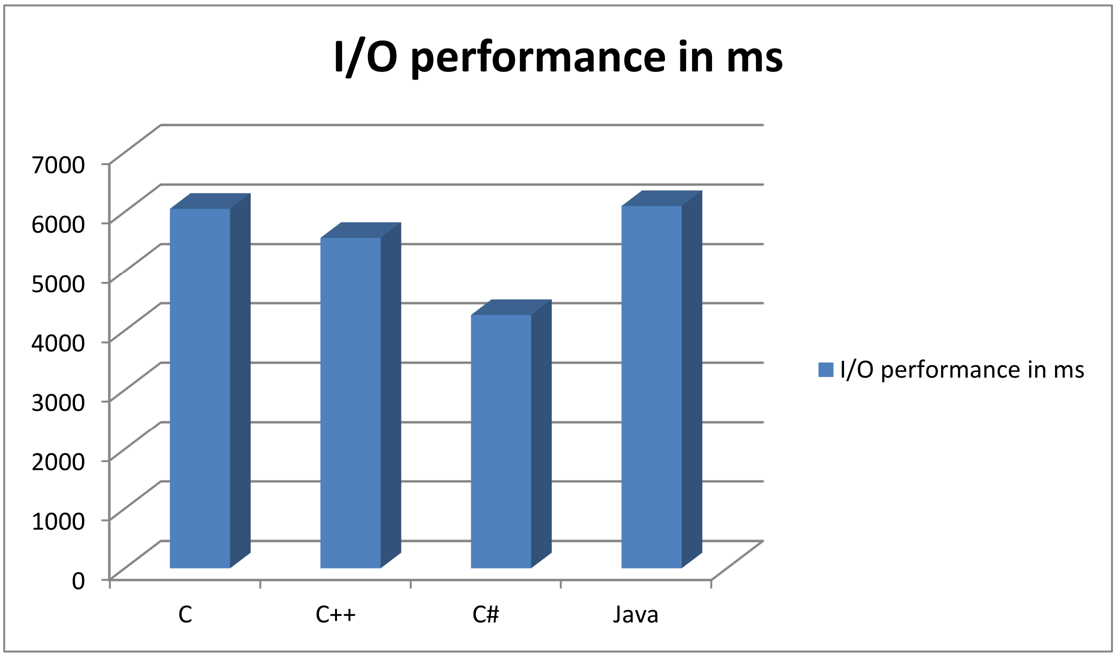


Figure 21. I/O performances testing in ms.

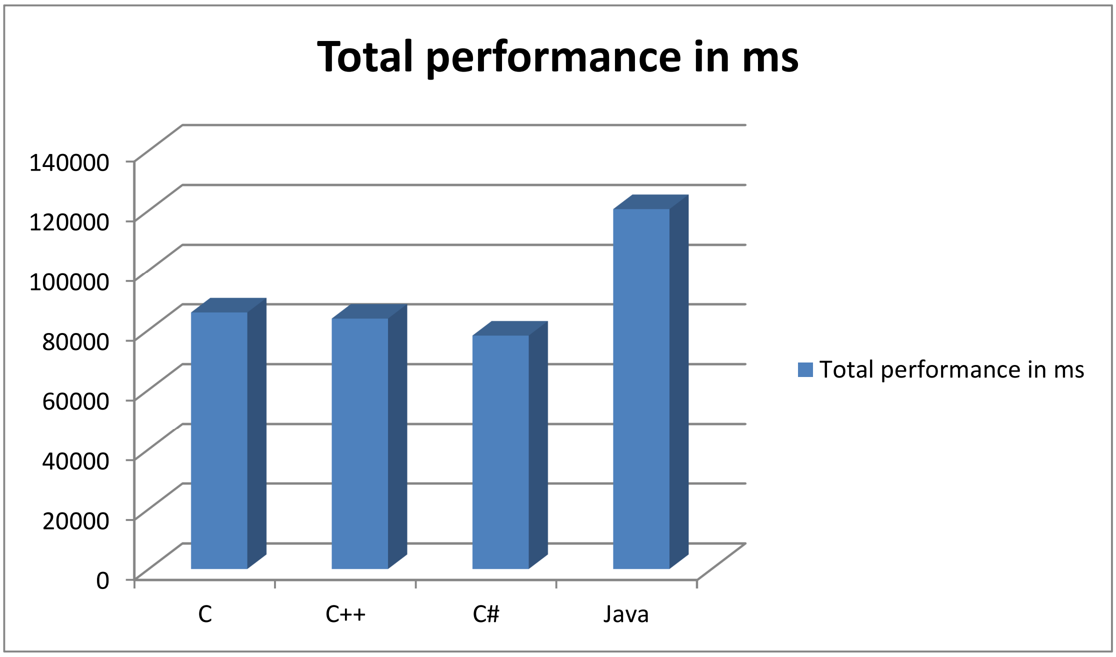


Figure 22. Total performances in ms.

C has the third best performance in all five parts of the benchmark and the total performance is the third best as well.

C++ did the good job. C++ has the second best performance in all five parts of the benchmark and the total performance is the second best as well.

C# is the best in the test of Int arithmetic, Double arithmetic, Trig arithmetic and I/O and total performance. Especially, in tri arithmetic and I/O test, the performance of C# is much better than other three programming languages. However, it has the worst performance in the Long arithmetic.

The result of implementing the benchmark of Java is quite interesting. Java has the best performances in Int arithmetic test, double arithmetic test and long arithmetic test, which is three parts out of six parts of benchmark. However, it has the worst performance in other two parts of benchmark, Trig arithmetic test and I/O performances test. Java also ranks the last position it the total performance, which means Java spends much more time to implement the benchmark than other three programming languages.

5 Conclusion

In this chapter, the advantages and disadvantages of the four chosen programming languages C, C++, C# and Java were summarized and the recommendation of their most suitable fields to apply was presented.

C is a minimalistic programming language because it could be compiled in a straightforward manner by a relatively simple compiler. C offers low-level access to memory via pointers and the ability to access specific hardware addresses. C generates only a few instructions of machine languages for each of its core language elements and does not require extensive run-time support. It can be concluded that C language is suitable for many systems-programming applications that had traditionally been implemented in assembly languages.

However, as C is structured oriented programming language and focuses on the procedural programming paradigm, it is relatively hard to control the large-scale program.

As C language has high level and machine level mixed programming capacity, it is used in most hardware related applications. It is very suitable for writing programs in embedded device, chip designing, industrial automation products and so forth and so on. Meanwhile, Software such as “Unix”, “windows”, and other antivirus can also be created by C language. Last but not at least, algorithm can also be implemented in C language easily.

C++ was originally designed to be an enhancement to C language. Basically it inherits all the advantages of C language. In addition, it has more features than C, such as encapsulation, multiple inheritance, and Polymorphism. It can be concluded that it is relatively easy to use C++ to develop a large or huge system compared with C language as C++ supports the object-oriented (OO) features.

However, C++ has some disadvantages. C++ code is easily prone to errors related to data types because C++ does not offer very strong type-checking. C++ does not support platform independent. It can't run on the all kinds of platforms. The main disadvantage is that C++ is not a pure object oriented programming language as it doesn’t have the feature of garbage collection. C++ adopts the pointers which lead to no security for the data.

C++ can be widely used in the software industry. As C++ can be a very fast programming language after compiled, the software such as application software, device drivers and high-performance server can be designed by C++.

C# is designed for programming the Microsoft .NET Framework. C# is a combination of all the other programming languages in an almost perfect balance. C# is a pure object-oriented language. The concise syntax of C is also added to it. C# syntax is more similar to Java rather than to C++. Pointer memory management in C# is not a problem anymore because the garbage collector takes care of this, much like Java. The relational database management system (RDBMS) such as Mysql, Oracle, and Microsoft SQL Server can work with C# by the simple connection procedures.

On the other hand, C# has the disadvantages. C# is not flexible. C# depends greatly on .NET framework. Without the component in the .NET framework, C# is difficult to implement.

C# can be applied to the application development because C# is a rapid application development (RAD) language. It can reduce the period of the application development significantly. Furthermore C# is also very suitable for the development of web application because C# consists of a large framework of pre-developed components which can simplify the code of web applications.

Java is a pure object-oriented programming language. It makes modular programs available in order to reuse the code. Java is open source. People can use it for free. It is also platform-independent, which is one of the most significant advantages of Java. Programs written in Java can easily move from one computer system to another.

Java also has some disadvantages. Java is a memory-consuming programming language. Java is slow because it has an extra layer between the systems and the programs. The extra layer is Java Virtual Machine (JVM). Anything done by the Java programs has to be executed by the Java Virtual Machine. Then it makes the system to do the actual instructions.

Java has three different forms, Java 2 Standard Edition (J2SE), Java2 Micro Edition (J2ME), and Java2 Enterprise Edition (J2EE) which is quite similar to what we have in Windows operating systems such as Windows Vista Home Basic Edition, Windows Vista Business Edition and Windows Vista Ultimate Edition. Each form of Java has its suitable application field. J2SE, which is also called CORE Java, is suitable for the desktop applications. J2ME is mainly used in embedded systems development, such as mobile phones, wireless application and PDA programming. J2EE, designed for enterprise applications, is mainly used for the development of distributed network program, such as e-commerce website and ERP systems.

All in all, it can be concluded that all four programming languages C, C++; C#, and Java have the advantages and disadvantages. It is really hard to say which one is better than the others and which one is faster than the others. But all four programming languages have their most suitable fields to apply. People can get fast and stable performance from the software written in a suitable programming language.

Reference

[1] Wikipedia, Fifth-generation programming language, available at http://en.wikipedia.org/wiki/Fifth-generation_programming_language (Accessed May 25th, 2010)

[2] TIOBE Programming Community Index for May 2010, available at http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html (Accessed May 25th, 2010)

[3] Wikipedia, Programming language, available at http://en.wikipedia.org/wiki/Programming_language (Accessed May 25th, 2010)

[4] The Evolution of Computer Programming Languages, available at http://www.associatedcontent.com/article/369862/the_evolution_of_computer_programming.html (Accessed May 25th, 2010)

[5] Wikipedia, Fifth-generation programming language, available at http://en.wikipedia.org/wiki/Fifth-generation_programming_language (Accessed May 25th, 2010)

[6] A Brief History of the C Language, available at http://hubpages.com/hub/A-Brief-History-of-the-C-Language (Accessed May 25th, 2010)

[7] A Brief History of C++, available at http://hubpages.com/hub/A-Brief-History-of-the-C-Language (Accessed May 25th, 2010)

[8] Online C++ tutorial, A Brief History of C++, available at http://www.intap.net/~drw/cpp/cpp01_03.htm (Accessed May 25th, 2010)

[9] C# Overview, available at http://en.csharp-online.net/CSharp_Overview#A_Brief_History_of_C.23 (Accessed May 25th, 2010)

[10] Beginning Java, Brief history of Java, available at http://mathbits.com/mathbits/java/Introduction/BriefHistory.htm (Accessed May 25th, 2010)

[11] RoseIndia, Variable in Java, Available at http://www.roseindia.net/java/master-java/variables-in-java.shtml (Accessed April 21th, 2010)

[12] Java Language Keywords, Available at http://java.sun.com/docs/books/tutorial/java/nutsandbolts/_keywords.html (Accessed April 21th, 2010)

[13] Primitive vs. Reference Data Types, Available at http://pages.cs.wisc.edu/~hasti/cs302/examples/primitiveVsRef.html (Accessed April 26th, 2010)

[14] Strings in C and C++, Available at http://cs.smu.ca/~porter/csc/ref/c_cpp_strings.html (Accessed April 26th, 2010)

[15] Wikipedia, Comparison of Java and C Sharp, available at http://en.wikipedia.org/wiki/Comparison_of_Java_and_C_Sharp#Introduction (Accessed April 25th, 2010)

[16] MSDN, C++ -> C#: What You Need to Know to Move from C++ to C#, available at http://msdn.microsoft.com/en-us/magazine/cc301520.aspx (Accessed April 25th, 2010)

[17] MSDN, C# for C++ Developers, available at http://msdn.microsoft.com/en-us/library/yyaad03b (VS.90).aspx (Accessed April 25th, 2010)

[18] A COMPARISON OF FT'S C# PROGRAMMING LMICROSOANGUAGE TO SUN MICROSYSTEMS' JAVA PROGRAMMING LANGUAGE available at http://www.25hoursaday.com/CsharpVsJava.html#multiarray (Accessed April 25th, 2010)

[19] Wikipedia, Comparison of Java and C Sharp, available at http://en.wikipedia.org/wiki/Comparison_of_Java_and_C_Sharp#Arrays (Accessed April 25th, 2010)

[20] About.com, What is an Enum? , available at http://cplus.about.com/od/introductiontoprogramming/p/enumeration.htm (Accessed April 27th, 2010)

[21] Wikipedia, Comparison of Java and C Sharp, available at http://en.wikipedia.org/wiki/Comparison_of_Java_and_C_Sharp#Enumerations (Accessed April 27th, 2010)

[22] Why C has Pointers, available at http://duramecho.com/ComputerInformation/WhyCPointers.html (Accessed April 27th, 2010)

[23] Wikipedia, Comparison of Java and C Sharp, available at

http://en.wikipedia.org/wiki/Comparison_of_Java_and_C_Sharp#Pointers (Accessed April 27th, 2010)

[24] Pointers in C#, available at http://www.c-sharpcorner.com/UploadFile/rajeshvs/PointersInCSharp11112005051624AM/PointersInCSharp.aspx (Accessed April 27th, 2010)

[25] Wikipedia, Comparison of Java and C Sharp, available at http://en.wikipedia.org/wiki/Comparison_of_Java_and_C_Sharp#Inner_classes (Accessed April 27th, 2010)

[26] Nested Class Declarations, available at http://msdn.microsoft.com/en-us/library/x23h0937(VS.80).aspx(Accessed April 27th, 2010)

[27] Wikipedia, Comparison of Java and C Sharp, available at http://en.wikipedia.org/wiki/Comparison_of_Java_and_C_Sharp#Partial_classes (Accessed April 27th, 2010)

[28] Generics in C#, Java, and C++, a Conversation with Anders Hejlsberg, Part VII by Bill Venners with Bruce Eckel available at http://www.artima.com/intv/generics2.html (Accessed April 28th, 2010)

[29] Interpretation versus compilation, available at http://www.j-paine.org/students/lectures/lect6/node20.html (Accessed May 10th, 2010)

[30] Java theory and practice: Dynamic compilation and performance measurement, available at http://www.ibm.com/developerworks/library/j-jtp12214/ (Accessed May 10th, 2010)

[31] The Compilation Process, available at http://www.acm.uiuc.edu/sigmil/RevEng/ch02.html (Accessed May 10th, 2010)

[32] C/C++ Program Compilation, available at available at http://www.cs.cf.ac.uk/Dave/C/node3.html#app:compiler (Accessed May 10th, 2010)

[33] Matercsharp, available at http://mastercsharp.com/article.aspx?ArticleID=90&TopicID=7 (Accessed May 10th, 2010)

[34]Stephen G. Kochan, “Understanding Data Types and Constants” in Programming in C, Third Edition, Sams Publishing, 800 East 96th Street, Indianapolis, Indiana, 46240 USA, 2005, pp 23-39

[35] Brian W. Kernighan and Dennis M. Ritchie，”Chapter2 Types, Operators and Expressions” in the C programming Language, Prentice-Hall Publishing

[36] C reserved keywords Available at http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp?topic=/com.ibm.debugtool.doc_8.1/eqa8ug02426.htm (Accessed April 20th, 2010).

[37] 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

[38] Wikipedia, C Sharp syntax 2010, Available at http://en.wikipedia.org/wiki/C_Sharp_syntax (Accessed April 20th, 2010)

[39] C# Station, C# tutorial, Available at http://www.csharp-station.com/Tutorials/Lesson02.aspx (Accessed April 21th, 2010)

[40] Wikipedia, Benchmark (computing), Available at http://en.wikipedia.org/wiki/Benchmark_(computing) (Accessed May 27th, 2010)

APPENDICES

APPENDIX 1 Result of running C benchmark

```
This is the 1st time of C_benchmark.
Start C benchmark
Int arithmetic elapsed time: 14570 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 19079 ms with doubleMin
10000000000.000000, doubleMax 11000000000.000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636
Long arithmetic elapsed time: 33852 ms with longMin 10000000000, longMax
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 14789 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990665
 cosine: -0.136322
 tangent: -7.267119
 logarithm: 7.000000
 squareRoot: 3162.277502
I/O elapsed time: 6489 ms with max of 1000000
 last line:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total elapsed time: 88779 ms
Stop C benchmark
This is 2nd time of C_benchmark.
Start C benchmark
Int arithmetic elapsed time: 14258 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18673 ms with doubleMin
10000000000.000000, doubleMax 11000000000.000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636
Long arithmetic elapsed time: 33103 ms with longMin 10000000000, longMax
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13526 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990665
 cosine: -0.136322
 tangent: -7.267119
 logarithm: 7.000000
 squareRoot: 3162.277502
I/O elapsed time: 7410 ms with max of 1000000
 last line:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total elapsed time: 86970 ms
Stop C benchmark
This is 3rd time of C_benchmark.
Start C benchmark
Int arithmetic elapsed time: 14320 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18798 ms with doubleMin
10000000000.000000, doubleMax 11000000000.000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636
Long arithmetic elapsed time: 32714 ms with longMin 10000000000, longMax
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13572 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990665
 cosine: -0.136322
 tangent: -7.267119
 logarithm: 7.000000
 squareRoot: 3162.277502
I/O elapsed time: 6240 ms with max of 1000000
 last line:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total elapsed time: 85644 ms
Stop C benchmark
This is 4th time of C_benchmark.
Start C benchmark
Int arithmetic elapsed time: 14211 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18673 ms with doubleMin
10000000000.000000, doubleMax 11000000000.000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636
Long arithmetic elapsed time: 33072 ms with longMin 10000000000, longMax
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13479 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990665
 cosine: -0.136322
 tangent: -7.267119
 logarithm: 7.000000
 squareRoot: 3162.277502
I/O elapsed time: 6084 ms with max of 1000000
 last line:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total elapsed time: 85519 ms
Stop C benchmark
This is 5th time of C_benchmark.
Start C benchmark
Int arithmetic elapsed time: 14180 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18642 ms with doubleMin
10000000000.000000, doubleMax 11000000000.000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636
Long arithmetic elapsed time: 33088 ms with longMin 10000000000, longMax
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13494 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990665
 cosine: -0.136322
 tangent: -7.267119
 logarithm: 7.000000
 squareRoot: 3162.277502
I/O elapsed time: 6754 ms with max of 1000000
 last line:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total elapsed time: 86158 ms
Stop C benchmark
```

```
This is 6th time of C_benchmark.
Start C benchmark
Int arithmetic elapsed time: 14258 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18658 ms with doubleMin
10000000000.000000, doubleMax 11000000000.000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636
Long arithmetic elapsed time: 33103 ms with longMin 10000000000, longMax
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13463 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990665
 cosine: -0.136322
 tangent: -7.267119
 logarithm: 7.000000
 squareRoot: 3162.277502
I/O elapsed time: 5990 ms with max of 1000000
 last line:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total elapsed time: 85472 ms
Stop C benchmark
This is 7th time of C_benchmark.
Start C benchmark
Int arithmetic elapsed time: 14258 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18658 ms with doubleMin
10000000000.000000, doubleMax 11000000000.000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636
Long arithmetic elapsed time: 33040 ms with longMin 10000000000, longMax
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13494 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990665
 cosine: -0.136322
 tangent: -7.267119
 logarithm: 7.000000
 squareRoot: 3162.277502
I/O elapsed time: 5429 ms with max of 1000000
 last line:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total elapsed time: 84879 ms
Stop C benchmark
This is 8th time of C_benchmark.
Start C benchmark
Int arithmetic elapsed time: 14211 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18658 ms with doubleMin
10000000000.000000, doubleMax 11000000000.000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636
Long arithmetic elapsed time: 33056 ms with longMin 10000000000, longMax
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13494 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990665
 cosine: -0.136322
 tangent: -7.267119
 logarithm: 7.000000
 squareRoot: 3162.277502
I/O elapsed time: 5367 ms with max of 1000000
 last line:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total elapsed time: 84786 ms
Stop C benchmark
This is 9th time of C_benchmark.
Start C benchmark
Int arithmetic elapsed time: 14242 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18674 ms with doubleMin
10000000000.000000, doubleMax 11000000000.000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636
Long arithmetic elapsed time: 33040 ms with longMin 10000000000, longMax
```

```
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13479 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990665
 cosine: -0.136322
 tangent: -7.267119
 logarithm: 7.000000
 squareRoot: 3162.277502
I/O elapsed time: 5382 ms with max of 1000000
 last line:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total elapsed time: 84817 ms
Stop C benchmark
This is 10th time of C_benchmark.
Start C benchmark
Int arithmetic elapsed time: 14227 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18673 ms with doubleMin
10000000000.000000, doubleMax 11000000000.000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636
Long arithmetic elapsed time: 33041 ms with longMin 10000000000, longMax
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13478 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990665
 cosine: -0.136322
 tangent: -7.267119
 logarithm: 7.000000
 squareRoot: 3162.277502
I/O elapsed time: 5367 ms with max of 1000000
 last line:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total elapsed time: 84786 ms
Stop C benchmark
```

APPENDIX 2 Result of running C++ benchmark

```
This is the 1st time of C++_benchmark.
Start C++ benchmark
Int arithmetic elapsed time: 14274 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18610 ms with doubleMin
10000000000.000000000000000, doubleMax 11000000000.000000000000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636000000000
Long arithmetic elapsed time: 31575 ms with longMax 11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13447 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.267118770164549
 logarithm: 6.999999956570550
 squareRoot: 3162.277502054492300
I/O elapsed time: 5429 ms with max of 1000000
 i: 1000001
 readLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh

Total elapsed time: 83335 ms
Stop C++ benchmark
This is 2nd time of C++_benchmark.
Start C++ benchmark
Int arithmetic elapsed time: 14258 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18673 ms with doubleMin
10000000000.000000000000000, doubleMax 11000000000.000000000000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636000000000
Long arithmetic elapsed time: 31933 ms with longMax 11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13463 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.267118770164549
 logarithm: 6.999999956570550
 squareRoot: 3162.277502054492300
I/O elapsed time: 5445 ms with max of 1000000
 i: 1000001
 readLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh

Total elapsed time: 83772 ms
Stop C++ benchmark
This is 3rd time of C++_benchmark.
Start C++ benchmark
Int arithmetic elapsed time: 14242 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18658 ms with doubleMin
10000000000.000000000000000, doubleMax 11000000000.000000000000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636000000000
Long arithmetic elapsed time: 31699 ms with longMax 11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13463 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.267118770164549
 logarithm: 6.999999956570550
 squareRoot: 3162.277502054492300
I/O elapsed time: 5756 ms with max of 1000000
 i: 1000001
 readLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh

Total elapsed time: 83818 ms
Stop C++ benchmark
This is 4th time of C++_benchmark.
Start C++ benchmark
Int arithmetic elapsed time: 14196 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
```

```
Double arithmetic elapsed time: 18673 ms with doubleMin
10000000000.000000000000000, doubleMax 11000000000.000000000000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636000000000
Long arithmetic elapsed time: 32198 ms with longMax 11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13479 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.267118770164549
 logarithm: 6.999999956570550
 squareRoot: 3162.277502054492300
I/O elapsed time: 5772 ms with max of 1000000
 i: 1000001
 readLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh

Total elapsed time: 84318 ms
Stop C++ benchmark
This is 5th time of C++_benchmark.
Start C++ benchmark
Int arithmetic elapsed time: 14352 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18704 ms with doubleMin
10000000000.000000000000000, doubleMax 11000000000.000000000000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636000000000
Long arithmetic elapsed time: 31668 ms with longMax 11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13463 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.267118770164549
 logarithm: 6.999999956570550
 squareRoot: 3162.277502054492300
I/O elapsed time: 5382 ms with max of 1000000
 i: 1000001
 readLine:
```

```
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh

Total elapsed time: 83569 ms
Stop C++ benchmark
This is 6th time of C++_benchmark.
Start C++ benchmark
Int arithmetic elapsed time: 14274 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18657 ms with doubleMin
10000000000.000000000000000, doubleMax 11000000000.000000000000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636000000000
Long arithmetic elapsed time: 31684 ms with longMax 11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13447 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.267118770164549
 logarithm: 6.999999956570550
 squareRoot: 3162.277502054492300
I/O elapsed time: 5834 ms with max of 1000000
 i: 1000001
 readLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh

Total elapsed time: 83896 ms
Stop C++ benchmark
This is 7th time of C++_benchmark.
Start C++ benchmark
Int arithmetic elapsed time: 14258 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18642 ms with doubleMin
10000000000.000000000000000, doubleMax 11000000000.000000000000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636000000000
Long arithmetic elapsed time: 31590 ms with longMax 11000000000
 i: 11000000000
 longResult: 776627965
```

```
Trig elapsed time: 13463 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.267118770164549
 logarithm: 6.999999956570550
 squareRoot: 3162.277502054492300
I/O elapsed time: 5382 ms with max of 1000000
 i: 1000001
 readLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh

Total elapsed time: 83335 ms
Stop C++ benchmark
This is 8th time of C++_benchmark.
Start C++ benchmark
Int arithmetic elapsed time: 14336 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18673 ms with doubleMin
10000000000.000000000000000, doubleMax 11000000000.000000000000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636000000000
Long arithmetic elapsed time: 32323 ms with longMax 11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13494 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.267118770164549
 logarithm: 6.999999956570550
 squareRoot: 3162.277502054492300
I/O elapsed time: 5398 ms with max of 1000000
 i: 1000001
 readLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh

Total elapsed time: 84224 ms
Stop C++ benchmark
This is 9th time of C++_benchmark.
Start C++ benchmark
```

```
Int arithmetic elapsed time: 14289 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18658 ms with doubleMin
10000000000.000000000000000, doubleMax 11000000000.000000000000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636000000000
Long arithmetic elapsed time: 31574 ms with longMax 11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13463 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.267118770164549
 logarithm: 6.999999956570550
 squareRoot: 3162.277502054492300
I/O elapsed time: 5429 ms with max of 1000000
 i: 1000001
 readLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh

Total elapsed time: 83413 ms
Stop C++ benchmark
This is 10th time of C++_benchmark.
Start C++ benchmark
Int arithmetic elapsed time: 14274 ms with intMax of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 18642 ms with doubleMin
10000000000.000000000000000, doubleMax 11000000000.000000000000000
 i: 11000000000.000000
 doubleResult: 10011632717.505636000000000
Long arithmetic elapsed time: 31575 ms with longMax 11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 13447 ms with max of 10000000
 i: 10000000.000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.267118770164549
 logarithm: 6.999999956570550
 squareRoot: 3162.277502054492300
```

```
I/O elapsed time: 5803 ms with max of 1000000
 i: 1000001
 readLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh

Total elapsed time: 83741 ms
Stop C++ benchmark
```

APPENDIX 3 Result of running C# benchmark

```
This is the 1st time of C#_benchmark.
Start C# benchmark
Int arithmetic elapsed time: 12620 ms with max of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 17908 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 doubleResult: 10011632717.4955
Long arithmetic elapsed time: 37627 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 5288 ms with max of 10000000
 i: 10000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.26711877016455
 logarithm: 6.99999995657055
 squareRoot: 3162.27750205449
IO elapsed time: 4056 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total C# benchmark time: 77499 ms
End C# benchmark
This is the 2nd time of C#_benchmark.
Start C# benchmark
Int arithmetic elapsed time: 12589 ms with max of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 17940 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 doubleResult: 10011632717.4955
Long arithmetic elapsed time: 38313 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 5304 ms with max of 10000000
 i: 10000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.26711877016455
 logarithm: 6.99999995657055
 squareRoot: 3162.27750205449
IO elapsed time: 5382 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total C# benchmark time: 79528 ms
End C# benchmark
This is the 3rd time of C#_benchmark.
Start C# benchmark
Int arithmetic elapsed time: 12636 ms with max of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 17971 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 doubleResult: 10011632717.4955
Long arithmetic elapsed time: 37658 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 5319 ms with max of 10000000
 i: 10000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.26711877016455
 logarithm: 6.99999995657055
 squareRoot: 3162.27750205449
IO elapsed time: 6708 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total C# benchmark time: 80292 ms
End C# benchmark
This is the 4th time of C#_benchmark.
Start C# benchmark
Int arithmetic elapsed time: 12620 ms with max of 1000000000
 i: 1000000001
 intResult: 1
```

```
Double arithmetic elapsed time: 17955 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 doubleResult: 10011632717.4955
Long arithmetic elapsed time: 37611 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 5397 ms with max of 10000000
 i: 10000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.26711877016455
 logarithm: 6.99999995657055
 squareRoot: 3162.27750205449
IO elapsed time: 3978 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total C# benchmark time: 77561 ms
End C# benchmark
This is the 5th time of C#_benchmark.
Start C# benchmark
Int arithmetic elapsed time: 12573 ms with max of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 17908 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 doubleResult: 10011632717.4955
Long arithmetic elapsed time: 38188 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 5288 ms with max of 10000000
 i: 10000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.26711877016455
 logarithm: 6.99999995657055
 squareRoot: 3162.27750205449
IO elapsed time: 3400 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total C# benchmark time: 77357 ms
End C# benchmark
This is the 6th time of C#_benchmark.
Start C# benchmark
Int arithmetic elapsed time: 12589 ms with max of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 17924 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 doubleResult: 10011632717.4955
Long arithmetic elapsed time: 38188 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 5288 ms with max of 10000000
 i: 10000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.26711877016455
 logarithm: 6.99999995657055
 squareRoot: 3162.27750205449
IO elapsed time: 3572 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total C# benchmark time: 77561 ms
End C# benchmark
This is the 7th time of C#_benchmark.
Start C# benchmark
Int arithmetic elapsed time: 12589 ms with max of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 17908 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 doubleResult: 10011632717.4955
Long arithmetic elapsed time: 38188 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 5304 ms with max of 10000000
 i: 10000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.26711877016455
 logarithm: 6.99999995657055
 squareRoot: 3162.27750205449
IO elapsed time: 3369 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total C# benchmark time: 77358 ms
End C# benchmark
This is the 8th time of C#_benchmark.
Start C# benchmark
Int arithmetic elapsed time: 12589 ms with max of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 17908 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 doubleResult: 10011632717.4955
Long arithmetic elapsed time: 38204 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 5304 ms with max of 10000000
 i: 10000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.26711877016455
 logarithm: 6.99999995657055
 squareRoot: 3162.27750205449
IO elapsed time: 4789 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total C# benchmark time: 78794 ms
End C# benchmark
This is the 9th time of C#_benchmark.
Start C# benchmark
```

```
Int arithmetic elapsed time: 12604 ms with max of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 17893 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 doubleResult: 10011632717.4955
Long arithmetic elapsed time: 37596 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 5304 ms with max of 10000000
 i: 10000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.26711877016455
 logarithm: 6.99999995657055
 squareRoot: 3162.27750205449
IO elapsed time: 3946 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total C# benchmark time: 77343 ms
End C# benchmark
This is the 10th time of C#_benchmark.
Start C# benchmark
Int arithmetic elapsed time: 12604 ms with max of 1000000000
 i: 1000000001
 intResult: 1
Double arithmetic elapsed time: 17893 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 doubleResult: 10011632717.4955
Long arithmetic elapsed time: 38173 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 5288 ms with max of 10000000
 i: 10000000
 sine: 0.990664647736125
 cosine: -0.136321516004849
 tangent: -7.26711877016455
 logarithm: 6.99999995657055
 squareRoot: 3162.27750205449
IO elapsed time: 3400 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total C# benchmark time: 77358 ms
End C# benchmark
```

APPENDIX 4 Result of running Java benchmark

```
This is the 1st time of Java_benchmark.
Start Java benchmark
Int arithmetic elapsed time: 8876 ms with max of 1000000000
 i: 1000000000
 intResult: 1
Double arithmetic elapsed time: 10312 ms with min of 1.0E10, max of 1.1E10
 i: 1.1E10
 doubleResult: 1.00116327174955E10
Long arithmetic elapsed time: 27643 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 67252 ms with max of 1.0E7
 i: 1.0E7
 sine: 0.9906646477361263
 cosine: -0.13632151600483616
 tangent: -7.267118770165242
 logarithm: 6.999999956570549
 squareRoot: 3162.2775020544923
IO elapsed time: 6973 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total Java benchmark time: 121056 ms
End Java benchmark
This is the 2nd time of Java_benchmark.
Start Java benchmark
Int arithmetic elapsed time: 8939 ms with max of 1000000000
 i: 1000000000
 intResult: 1
Double arithmetic elapsed time: 10343 ms with min of 1.0E10, max of 1.1E10
 i: 1.1E10
 doubleResult: 1.00116327174955E10
Long arithmetic elapsed time: 27752 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 67205 ms with max of 1.0E7
 i: 1.0E7
 sine: 0.9906646477361263
 cosine: -0.13632151600483616
 tangent: -7.267118770165242
 logarithm: 6.999999956570549
 squareRoot: 3162.2775020544923
IO elapsed time: 6038 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total Java benchmark time: 120277 ms
End Java benchmark
This is the 3rd time of Java_benchmark.
Start Java benchmark
Int arithmetic elapsed time: 8892 ms with max of 1000000000
 i: 1000000000
 intResult: 1
Double arithmetic elapsed time: 10281 ms with min of 1.0E10, max of 1.1E10
 i: 1.1E10
 doubleResult: 1.00116327174955E10
Long arithmetic elapsed time: 27690 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 67220 ms with max of 1.0E7
 i: 1.0E7
 sine: 0.9906646477361263
 cosine: -0.13632151600483616
 tangent: -7.267118770165242
 logarithm: 6.999999956570549
 squareRoot: 3162.2775020544923
IO elapsed time: 6162 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total Java benchmark time: 120245 ms
End Java benchmark
This is the 4th time of Java_benchmark.
Start Java benchmark
Int arithmetic elapsed time: 8908 ms with max of 1000000000
 i: 1000000000
 intResult: 1
Double arithmetic elapsed time: 10296 ms with min of 1.0E10, max of 1.1E10
 i: 1.1E10
 doubleResult: 1.00116327174955E10
```

```
Long arithmetic elapsed time: 27628 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 67173 ms with max of 1.0E7
 i: 1.0E7
 sine: 0.9906646477361263
 cosine: -0.13632151600483616
 tangent: -7.267118770165242
 logarithm: 6.999999956570549
 squareRoot: 3162.2775020544923
IO elapsed time: 6240 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total Java benchmark time: 120245 ms
End Java benchmark
This is the 5th time of Java_benchmark.
Start Java benchmark
Int arithmetic elapsed time: 8907 ms with max of 1000000000
 i: 1000000000
 intResult: 1
Double arithmetic elapsed time: 10312 ms with min of 1.0E10, max of 1.1E10
 i: 1.1E10
 doubleResult: 1.00116327174955E10
Long arithmetic elapsed time: 27643 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 67158 ms with max of 1.0E7
 i: 1.0E7
 sine: 0.9906646477361263
 cosine: -0.13632151600483616
 tangent: -7.267118770165242
 logarithm: 6.999999956570549
 squareRoot: 3162.2775020544923
IO elapsed time: 6240 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total Java benchmark time: 120260 ms
End Java benchmark
```

```
This is the 6th time of Java_benchmark.
Start Java benchmark
Int arithmetic elapsed time: 8908 ms with max of 1000000000
 i: 1000000000
 intResult: 1
Double arithmetic elapsed time: 10311 ms with min of 1.0E10, max of 1.1E10
 i: 1.1E10
 doubleResult: 1.00116327174955E10
Long arithmetic elapsed time: 27659 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 67455 ms with max of 1.0E7
 i: 1.0E7
 sine: 0.9906646477361263
 cosine: -0.13632151600483616
 tangent: -7.267118770165242
 logarithm: 6.999999956570549
 squareRoot: 3162.2775020544923
IO elapsed time: 5850 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total Java benchmark time: 120183 ms
End Java benchmark
This is the 7th time of Java_benchmark.
Start Java benchmark
Int arithmetic elapsed time: 8970 ms with max of 1000000000
 i: 1000000000
 intResult: 1
Double arithmetic elapsed time: 10374 ms with min of 1.0E10, max of 1.1E10
 i: 1.1E10
 doubleResult: 1.00116327174955E10
Long arithmetic elapsed time: 27924 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 67533 ms with max of 1.0E7
 i: 1.0E7
 sine: 0.9906646477361263
 cosine: -0.13632151600483616
 tangent: -7.267118770165242
 logarithm: 6.999999956570549
 squareRoot: 3162.2775020544923
IO elapsed time: 5444 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total Java benchmark time: 120245 ms
End Java benchmark
This is the 8th time of Java_benchmark.
Start Java benchmark
Int arithmetic elapsed time: 8907 ms with max of 1000000000
 i: 1000000000
 intResult: 1
Double arithmetic elapsed time: 10312 ms with min of 1.0E10, max of 1.1E10
 i: 1.1E10
 doubleResult: 1.00116327174955E10
Long arithmetic elapsed time: 27705 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 67704 ms with max of 1.0E7
 i: 1.0E7
 sine: 0.9906646477361263
 cosine: -0.13632151600483616
 tangent: -7.267118770165242
 logarithm: 6.999999956570549
 squareRoot: 3162.2775020544923
IO elapsed time: 5476 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total Java benchmark time: 120104 ms
End Java benchmark
This is the 9th time of Java_benchmark.
Start Java benchmark
Int arithmetic elapsed time: 8939 ms with max of 1000000000
 i: 1000000000
 intResult: 1
Double arithmetic elapsed time: 10343 ms with min of 1.0E10, max of 1.1E10
 i: 1.1E10
 doubleResult: 1.00116327174955E10
Long arithmetic elapsed time: 27752 ms with min of 10000000000, max of
11000000000
```

```
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 67798 ms with max of 1.0E7
 i: 1.0E7
 sine: 0.9906646477361263
 cosine: -0.13632151600483616
 tangent: -7.267118770165242
 logarithm: 6.999999956570549
 squareRoot: 3162.2775020544923
IO elapsed time: 5507 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total Java benchmark time: 120339 ms
End Java benchmark
This is the 10th time of Java_benchmark.
Start Java benchmark
Int arithmetic elapsed time: 8923 ms with max of 1000000000
 i: 1000000000
 intResult: 1
Double arithmetic elapsed time: 10343 ms with min of 1.0E10, max of 1.1E10
 i: 1.1E10
 doubleResult: 1.00116327174955E10
Long arithmetic elapsed time: 27768 ms with min of 10000000000, max of
11000000000
 i: 11000000000
 longResult: 776627965
Trig elapsed time: 67517 ms with max of 1.0E7
 i: 1.0E7
 sine: 0.9906646477361263
 cosine: -0.13632151600483616
 tangent: -7.267118770165242
 logarithm: 6.999999956570549
 squareRoot: 3162.2775020544923
IO elapsed time: 7051 ms with max of 1000000
 i: 1000001
 myLine:
abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567
890abcdefgh
Total Java benchmark time: 121602 ms
End Java benchmark
```