

Evaluation of C++ as object-oriented programming language compared to Java

Hafiza Elbadi Ahmed

Princess Nora Bint Abdelrahman University, Riyadh, Saudi Arabia

Abstract

C++ is an object-oriented Programming (OOP) language, which in addition, supports the procedure oriented codes of C. Program coding in C++ codes provides the advantage of objected oriented programming as well as the advantage of C and in-line assembly. C++ is rapidly becoming one of the most important programming languages in the world and widely-used in many real world applications. C++ is widespread adoption and support – cross platform, industry, and academia, Each C++ application (complete code) has a main features ,C++ provides all the advantages of 'C' and of object oriented programming and is suitable for embedded systems; it helps in optimizing the generated codes. . Using embedded C++ compilers or the special compilers make the C++ more powerful coding language than C for embedded systems

Keywords: **Features, Data types, syntax**

1. Introduction

C++ is rapidly becoming one of the most important programming languages in the world. Yet it is an extremely complex language and, therefore, one that is very difficult to learn. C++ is a complex programming language and, consequently, it is difficult to learn [1 , 2,3,4,5].However, it is quickly becoming the predominant software systems development language throughout the world [2, 3, 5,6,7,8,9,10].

Therefore, it is important for software development practitioners, educators, and trainers to become acquainted with this complicated language. For the same reason, students and other individuals who are learning to develop software also need to become familiar

with C++.The standard definition of C++ has not yet been formally completed or accepted [11]

It currently takes the form of a working paper for a Draft Proposed International Standard for Information Systems Programming Language C++ [12],(hereafter called "draft proposed standard"). Nevertheless, its effect on programming languages and programming has been profound and relatively long-term, For instance, even with traditional C as its base language, the earliest published description of C++ [13] had an important influence on the definition of standard C [14], and It may also have an impact on the next version of standard C[15]. Furthermore, C++ has already had and is continuing to have a significant impact on systems and application software development[2,3,6,5,7,8,9,10,15,16,17,18,19].

It has become one of the most important and widely-used programming languages. Accordingly, it is increasingly being adopted, primarily in place of Pascal, as the programming language of choice for the beginning college level computer science course [3, 20 , 21,22].Much of the Java language builds on C/C++. Some of the language keywords and behavior is very similar [23].

C++ is an object oriented Program (OOP) language, which in addition, supports the procedure oriented codes of C. Program coding in C++ codes provides the advantage of objected oriented programming as well as the advantage of C and in-line assembly[24].

2. General Features of C++

The program is written in a file using a text editor such as vi, gedit, or emacs. A file

containing C++ code is conventionally designated by one of the Suffixes:

.c .cc .cpp .cxx

The C++ source files are compiled and linked through a C++ compiler to

Produce the corresponding binary executable

Free C++ compilers are available for the Linux platform. Cygwin and Borland offer complimentary compilers for other operating systems. Some compilers are bundled in an integrated development environment (IDE) offering dazzling graphical user interfaces (GUI).

2.1 The main function

Each C++ application (complete code) has a main function that is first loaded into memory and then transferred to the CPU for execution. When execution has been concluded, the main function returns the integer 0. This practice is motivated partly by issues of backward compatibility

```
int main()
{
...
return 0;
}
```

2.2 Grammar and syntax

Where:

- Int indicates that an integer will be returned on completion. The penultimate line sets this integer to 0, signaling the success of the execution.
- The parentheses after main enclose the arguments of the main function; in this case, there are no arguments.
- The curly brackets mark the beginning and the end of the enclosed main program consisting of various instructions.
- The dots stand for additional lines of code.
- The semicolon is a delimiter, marking the end of the preceding command return 0, which concludes the execution. A simplified version of

the main program that returns nothing on execution is:

```
main()
{
...
}
```

However, the previous structure with the return statement included is highly recommended as a standard practice. If an error occurs during compilation or execution, this list should

Serve as a first checkpoint.

- C++ is (lower and upper) case sensitive For example, the variable echidna is different from the variable echiDna, and the C++ command return is not equivalent to the non-existent command Return.

- Beginning of a statement

A C++ statement or command may begin at any place in a line and continue onto the next line. In fact, a statement may take several lines of code.

- End of a statement

The end of a statement is indicated by a semicolon “;” (statement delimiter.) Thus:

```
a=5;
```

If the semicolon is not included, the compiler will assume that the statement in the next line is a continuation of the statement in the present line.

- Multiple commands in a line

Two or more statements can be placed in the same line provided they are separated with semicolons. Thus:

```
a=5; b=10;
```

- White space

An empty (blank) space separates two words. The compiler ignores more than one empty space between two words. A number cannot be broken up into pieces separated by white space; thus, it may not be 92 093 instead of 92093.

- Statement and command blocks

Blocks of statements or commands defining procedures are enclosed by curly brackets (block delimiters)

```
{
```

```
...
}
```

Note that it is not necessary to put a semicolon after the closing bracket.

- In-line comments

In-line comments may be inserted following the double slash “//”. For Example:

```
a = 10.0; // ignore me
```

The text: // ignore me is ignored by the compiler.

To deactivate (comment out) a line, such as:

```
// a = 34.5;
```

A distinction should be made between the slash (/) and the backslash (\). These are two different symbols separated by two rows on the keyboard.

- Commentary

All text enclosed between a slash-asterisk pair (/*) and the converse Asterisk-slash pair (*/) is commentary and ignored by the compiler. Such as:

```
/* ---- main program ---- */
```

To provide documentation at the beginning of a code, it can be written in:

```
/* PROGRAM: late
   AUTHOR: Justin Case
   PURPOSE: produce an excuse for being
           late */
```

2.3 Data types

In mathematical modeling and computer programming, introducing variables representing abstract notions and physical objects. In C++, the name of a variable must start with a letter and contain only Letters, numbers, and the underscore (_). Names reserved for C++ grammar and syntax cannot be employed. Acceptable variables obey the rules discussed in this section [25].

- Simple data types include
 - Integers data types include [char, short, Int, long and Bool]
 - Floating point
 - Enumeration

The table [26] below shows the date types and their description.

Table 1: Date Types

Name	Size (bytes)	Description	Range
Char	1	Character or eight bit integer	Signed:128..127 Unsigned: 0..255
short	2	Sixteen bit integer	Signed:-32768..32767 Unsigned: 0..65535
Long	4	Thirty-two bit integer	Signed -231... 231-1 Unsigned: 0... 232
Int	*(4)	System dependent, likely four bytes or thirty-two bits	Signed: -32768..32767 Unsigned: 0..65535
Float	4	Floating point number	3.4e +/-38 (7 digits)
Double	8	Double precision floating point	1.7e +/-308 (15 digits)
Long double	10	Long Double precision floating point	1.2e +/-4932 (19 digits)
Bool	1	Boolean value false 0,true 1	{0,1}

2.4 Compiling in UNIX

To compile the program on a UNIX system, the directory supposed to be navigated where this file resides, and issue the command:

```
C++ addition.cc
```

This statement invokes the C++ compiler with a single argument equal to the file name. The compiler will run and produce an executable binary, which may then be loaded into memory (executed)

Other compilation options are available, as explained in the compiler manual invoked by typing:

```
man gcc
```

For the GNU project C and C++ compilers [25].

3. Application of C++

C++ is used in many real world applications, such as:

1. Games.
2. Graphic User Interface (GUI) based applications.
3. Web Browsers.
4. Advance Computations and Graphics.
5. Database Software.
6. Operating Systems.
7. Enterprise Software.
8. Medical and Engineering Applications.
9. Compilers.

4. Strengths of C++

- Compiled code - capable of high performance comparable with FORTRAN, C
- Flexible coding styles - Functional, object oriented, high level, low level
- Powerful standard library with many functions, more added with time (BOOST)
- Local scoping of variables (more lately)
- Widespread adoption and support – cross platform, industry, and academia

5. Disadvantages of C++

- A powerful and expansive tool - easy to code for coding's sake (over engineering).
- Matrices and arrays are horrible.
- High performance code is harder to write (write for the compiler).
- Cryptic debugging for advanced features, and some not so advanced features [27].

6. Comparing between Java and C++

Java and C++ are the most used languages in programming for most of programmers and system designers. Java has a structure called an "Interface". Java interface is almost identical to

a C++ class that has nothing but pure virtual functions [28].

The differences between Java and C++ can be summarized as in Table 2.

Table 2: Differences between Java and C++ [29,30,31,32]

Item	C++	JAVA
1	While C++ interface is a class.	Java interface is not a class.
2	In C++ the functions can be implemented using inheritance and there is many options of such implementation using regular inheritance between two variables A and B if we need two copies or one copy then the virtual inheritance can be used.	Functions declared within Java interface cannot be implemented using that interface and have no member variables.
3	Where as in C++ it was a class with nothing but pure virtual functions. However the Subject class is quite different.	The Clock in Java is an interface.
4	The new object is referred to by the variable "c". Note that "c" is rather like a reference variable in C++, also C++ is often criticized for its lack of GC. However, many people have added garbage collectors to C++.	Java uses garbage collection. Garbage collection is a scheme of memory management that automatically frees blocks of memory sometime after all references to that memory have been redirected.
5	Templates are a wonderful feature of C++	Java does not have templates, which is of some concern to any programmer. In Java, one cannot create a type-safe container. All containers in Java can hold any kind of object. This can lead to some ugly problems.

7. Discussion

Indeed, Java is a powerful language. While C++ has a relatively easy time to be learned, and will

find that the programmers enjoy using it. But it has some disadvantages. Language design always involves some disadvantages or shortcomings that displease someone[28]. C++ is an interesting language that enables writing codes easily with more flexibility and with little time needed to execute some code comparing to Java.

8. Conclusions

C++ provides all the advantages of 'C' and of object oriented programming and is suitable for embedded systems; it helps in optimizing the generated codes. Using embedded C++ compilers or the special compilers make the C++ more powerful coding language than Java for embedded systems due to the OOP features of software re-usability, extendibility, polymorphism, function overriding and overloading along portability of C codes and in-line assembly codes.

9. References

- 1 Abelson, H., Bruce, K., van Dam, A., Harvey, B., Tucker, A. and Wegner, P., 1995. The first-course conundrum. *Communications of the ACM*, 38(6), pp.116-117.
- 2 Allison, Chuck, , April, 1995, "A Better C," *C/C++ Users Journal*, pp. 67-78.
- 3 Berman, A.M., Decker, R., Nguyen, D.X., Reid, R.J. and Wallingford, E., 1994. Using C++ in CS 1/CS 2. *ACM SIGCSE Bulletin*, 26(1), pp.383-384.
- 4 Engle, C.B., 1995. An Educator's Perspective on Ada. *AdaC News*, 12(1), p.1.
- 5 Vilot, Michael J. October, 1994, "An Introduction to the Standard Template Library," *C++ Report*, pp. 22-28.
- 6 Booch, G., 2006. *Object oriented analysis & design with application*. Pearson Education India.
- 7 Friedman, R.P., 1993. *Creating creating*. *C++ Report*, 5(4), p.4.
- 8 Keffer, T., 1992. Why C++ will replace Fortran. *Dr Dobb's Journal-Software Tools for the Professional Programmer*, 17(12), pp.39-47.
- 9 Reid, R.J., 1993. C++ as a first programming language. *C++ Report*, 5(4), pp.41-44.
- 10 Sebesta, R.W., 1993. *Concepts of programming languages*. Pearson Education India.
- 11 Clamage, S., Status of ANSISO C++ Standard," a comp. std. c++ newsgroup posting# 7383 dated 17 July 1995 16: 54: 39 GMT. Clamage is Acting Chair, ANSI X3J16.
- 12 Koenig, A., 1995. Working Paper for Draft Proposed International Standard for Information Systems—Programming Language C++. Accredited Standards Committee X, 3, pp.95-0087.
- 13 Stroustrup, B., 1986 *The C++ programming language*. Addison-Wesley Publishing Company.
- 14 ANSI Technical Committee X3J11, 1989. *American National Standard for Information Systems Programming Language C*, ANSI 3.159-1989. New York. American National Standards Institute, (The official current version of standard C.)
- 15 Jervis, Bob, October, 1994, "All is Flux," *C/C++ Users Journal*, pp. 39-49.

-
- 16 Barton, John J., and Lee IL Nackman, September, 1993, "Scientific & Engineering Programming in C++," C I I Report, vol. 5, no. 7, pp. 26-31
 - 17 Booch, G., 2006. Object oriented analysis & design with application. Pearson Education India.
 - 18 Buzzi-Ferraris, Guido, 1993. Scientific C! I, Addison-Wesley Publishing Company.
 - 19 Jagadeesh, J. M., (Product Reviews Editor), July,1995, "C++ Class Libraries: Part I," Computer, pp. 83-87.
 - 20 Chandra, K., 1993. C++ in eight weeks. ACM SigPlan Notices, 28(8), pp.29-39.
 - 21 Morton, L. and Norgaard, N., 1993. A survey of programming languages in CS programs. ACM SIGCSE Bulletin, 25(2), pp.9-11.
 - 22 Reid, R.J., 1995. The Programming Language of the First Computer Science Course. unpublished informal and on-going survey conducted over the Internet.
 - 23 Blackburn, S.M., Garner, R., Hoffmann, C., Khang, A.M., McKinley, K.S., Bentzur, R., Diwan, A., Feinberg, D., Frampton, D., Guyer, S.Z. and Hirzel, M., 2006, October. The DaCapo benchmarks: Java benchmarking development and analysis. In ACM Sigplan Notices (Vol. 41, No. 10, pp. 169-190). ACM
 - 24 Kirch-Prinz, U. and Prinz, P., 2002. A complete guide to programming in C++. Jones & Bartlett Learning.
 - 25 Pozrikidis, C., 2007. Introduction to C++ programming and graphics. Springer Science & Business Media.
 - 26 Stroustrup, B., 2000. The C++ programming language. Pearson Education India.
 - 27 Douglas C. Schmidt, January 18, 2012, An Overview of C++.
 - 28 AlHeyasat, O., Abu-Ein, A.A.K., Hatamleh, H. and Sharadqeh, A.A., 2012. Time comparing between Java and C++ software. Journal of Software Engineering and Applications, 5(08), p.630.
 - 29 Prechelt, L., 1999. Comparing Java vs. C/C++ Efficiency Differences to Interpersonal Differences. Commun. ACM, 42(10), pp.109-112.
 - 30 Sestoft, P., 2010. Numeric performance in C, C# and Java. IT University of Copenhagen Denmark, Version 0.9, 1.
 - 31 Wu, P., Midkiff, S., Moreira, J. and Gupta, M., 1999, June. Efficient support for complex numbers in Java. In Proceedings of the ACM 1999 conference on Java Grande (pp. 109-118). ACM.
 - 32 Moreira, J.E., Midkiff, S.P., Gupta, M., Artigas, P.V., Snir, M. and Lawrence, R.D., 2000. Java programming for high-performance numerical computing. IBM Systems Journal, 39(1), pp.21-56