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 object 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 object 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.

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

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 nonexistent 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:

```c
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:

```c
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:
  
  ```c
  a = 10.0;  // ignore me
  The text: // ignore me is ignored by the compiler. **To deactivate (comment out) a line, such as:**
  ```
  ```c
  // 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:
  ```c
  /* ---- main program ----- */
  
  To provide documentation at the beginning of a code, it can be written in:
  ```c
  /* 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>
<thead>
<tr>
<th>Name</th>
<th>Size (bytes)</th>
<th>Description</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Char</td>
<td>1</td>
<td>Character or eight bit integer</td>
<td>Signed:128..127 Unsigned: 0..255</td>
</tr>
<tr>
<td>short</td>
<td>2</td>
<td>Sixteen bit integer</td>
<td>Signed: -32768..32767 Unsigned: 0..65535</td>
</tr>
<tr>
<td>Long</td>
<td>4</td>
<td>Thirty-two bit integer</td>
<td>Signed :231... 231-1 Unsigned: 0.. 232</td>
</tr>
<tr>
<td>Int</td>
<td>*(4)</td>
<td>System dependent, likely four bytes or thirty-two bits</td>
<td>Signed: -32768..32767 Unsigned: 0..65535</td>
</tr>
<tr>
<td>Float</td>
<td>4</td>
<td>Floating point number</td>
<td>3.4e +/-38 (7 digits)</td>
</tr>
<tr>
<td>Double</td>
<td>8</td>
<td>Double precision floating point</td>
<td>1.7e +/-308 (15 digits)</td>
</tr>
<tr>
<td>Long double</td>
<td>10</td>
<td>Long Double precision floating point</td>
<td>1.2e +/-4932 (19 digits)</td>
</tr>
<tr>
<td>Bool</td>
<td>1</td>
<td>Boolean value false 0,true 1</td>
<td>{0,1}</td>
</tr>
</tbody>
</table>

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.
5. Database Software.
7. Enterprise Software.
8. Medical and Engineering Applications.

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++

<table>
<thead>
<tr>
<th>Item</th>
<th>C++</th>
<th>JAVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>While C++ interface is a class.</td>
<td>Java interface is not a class.</td>
</tr>
<tr>
<td>2</td>
<td>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.</td>
<td>Functions declared within Java interface cannot be implemented using that interface and have no member variables.</td>
</tr>
<tr>
<td>3</td>
<td>Where as in C++ it was a class with nothing but pure virtual functions. However the Subject class is quite different.</td>
<td>The Clock in Java is an interface.</td>
</tr>
<tr>
<td>4</td>
<td>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++.</td>
<td>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.</td>
</tr>
<tr>
<td>5</td>
<td>Templates are a wonderful feature of C++</td>
<td>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.</td>
</tr>
</tbody>
</table>

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


22 Reid, R.J., 1995. The Programming Language of the First Computer Science Course. unpublished informal and on-going survey conducted over the Internet.


30 Sestoft, P., 2010. Numeric performance in C, C# and Java. IT University of Copenhagen Denmark, Version 0.9, 1.
