ANALYSIS

5(14), August 1, 2013

ISSN 2278–5469 EISSN 2278–5450

Discovery

Comparative study of C, Objective C, C++ programming language

Surbhi Gupta, Puneet Bhalla

Dept. of Computer science & Engineering, Dronacharya College of Engineering, Gurgaon, India

Publication History
Received: 14 June 2013
Accepted: 17 July 2013
Published: 1 August 2013

Citation
Surbhi Gupta, Puneet Bhalla. Comparative study of C, Objective C, C++ programming language. *Discovery*, 2013, 5(14), 11-14

ABSTRACT

“A programming language is an artificial language designed to express computations that can be performed by a machine, particularly a computer”. The initial development of C occurred at AT&T Bell Labs between 1969 and 1973. It was named "C" because its features were derived from an earlier language called "B" was a stripped-down version of the BCPL programming language. This language was created for a specific purpose to design the UNIX operating system. From the beginning, C was intended to be useful--to allow busy programmers to get things done easily and without remembering the register binary values which was a major problem in coding with C language. To overcome problems of Objective-C language new language is derived in 1980’s. With concept of object oriented and some predefined rules or functionalities a language is designed known as C with classes. Later this language gets its originals name i.e., C++. C++ is a language which is semi dependent on machine and which is easy to understand. C++ is different from C in terms of syntaxes, header files, some rules and many more things. The C and C++ programming languages are closely related. C++ grew out of C, as it was designed to be source-and-link compatible with C. However, due to minor semantic differences, most non-trivial C programs will not compile as C++ code without modification—C++ is not a strict superset of C. This research paper concludes withthe influence of one language to another and comparative study of the languages.

Keywords: C++, C, UNIX, OBJECTIVE C.

1. INTRODUCTION

This paper is about the comparison of the C, Objective C & C++ programming language, the influences of one language on another, and the conditions under which they were created. “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” (A Brief History of the C Language). 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” (A Brief History of the C Language). 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 (A Brief History of the C Language).

The Objective-C language is a simple computer language designed to enable sophisticated object-oriented programming. Objective-C is defined as a small but powerful set of extensions to the standard ANSI C language. Its additions to C are mostly based on Smalltalk, one of the first object-oriented programming languages. Objective-C is designed to give C full object-oriented programming capabilities, and to do so in a simple and straightforward way. It is the main programming language used by Apple for the OS X and iOS operating systems and their respective APIs, Cocoa and Cocoa Touch.

"C++ was written by BjarneStroustrup at Bell Labs during 1983-1985. C++ is an extension of C. Prior to 1983; BjarneStroustrup 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." A Brief History of C++ "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" (Online C++ tutorial).

2. HISTORY

2.1. C

The initial development of C occurred at AT&T Bell Labs between 1969 and 1973; (A Brief History of the C Language) according to Ritchie, the most creative period occurred in 1972. It was named "C" because its features were derived from an earlier language called "B".In 1978, Brian Kernighan and Dennis Ritchie published the first edition of The C Programming Language. This book, known to C programmers as "K&R", served for many years as an informal specification of the language. The version of C that it describes is commonly referred to as K&R C. The second edition of the book covers the later ANSI C standard. K&R introduced several language features:

Standard I/O library,
longint data type,
unsignedint data type,
In C Language, There are four basic Date type.

Basic data type in C language (Brian et al. 1988).

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 (Stephen, 2005)

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

There are lists of all the reserved keywords in C language. "These keywords cannot be abbreviated, used as variable names, or used as any other type of identifiers" (C reserved keywords).

Reserved keywords in C language (C reserved keywords)

```
auto  else  long  switch
break  enum  register  typedef
case  extern  return  union do  int
char  float  short  unsigned
const  for  signed  void
continue  gotosizeof  volatile
default  if  static  while 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 (Brian et al. 1988).

2.2. C++

Stroustrup found that Simula had features that were very helpful for large software development, but the language was too slow for practical use, while BCPL was fast but too low-level to be suitable for large software development. When Stroustrup started working inAT&T Bell Labs, he had the problem of analyzing the UNIX kernel with respect to distributed computing. Remembering his Ph.D. experience, Stroustrup set out to enhance the C language with Simula-like features. C was chosen because it was general-purpose, fast, portable and widely used. Besides C and Simula, some other languages that inspired him were ALGOL 68, Ada, CLU and ML. In 1983, the name of the language was changed from C with Classes to C++ (++ being the increment operator in C). New features were added including virtual, function name and operator overloading, references, constants, user-controlled free-store memory control, improved type checking, and BCPL style single-line comments with two forward slashes (//). 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. There are 32 keywords both in C and C++ as shown.

```
auto  else  long  switch
break  enum  register  typedef
case  extern  return  union
char  float  short  unsigned
const  for  signed  void
continue  gotosizeof  volatile
default  if  static  while
do  intstruct  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++.
New keyword in C++.

```
asmdynamic_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 typeidtypename
using
virtual wchar_t1
```

2.3. Objective-C

Objective-C was created primarily by Brad Cox and Tom Love in the early 1980s at their company Stepstone. Both had been introduced to Smalltalk while at ITT Corporation's Programming Technology Center in 1981. Cox was intrigued by problems of true reusability in software design and programming. He realized that a language like Smalltalk would be invaluable in building development environments for system developers at ITT. However, he and Tom Love also recognized that backward compatibility with C was critically important in ITT's telecom engineering milieu (Biancuzzi et al. 2009). Cox began writing a pre-processor for C to add some of the capabilities of Smalltalk. He soon had a working implementation of an object-oriented extension to the C language, which he called "OOPC" for Object-Oriented Pre-Compiler (Cox, 1983). Love and Cox eventually formed a new venture, Productivity Products International (PPI), to commercialize their product, which coupled an Objective-C compiler with class libraries. In 1986, Cox published the main description of Objective-C in its original form in the book Object-Oriented Programming, An Evolutionary Approach.

3. INFLUENCES

C is influenced by languages like Assembly, B (BCPL, CPL), ALGOL68, FORTRAN i.e., these languages helped in developing C. C is the parent language of many other languages or mostly of all the languages. Many later languages have borrowed directly or indirectly from C, including: C#, D, Go, Java, Java Script, Limbo, LPC, Perl, PHP, Python, and Unix's C Shell. The most pervasive influence on these languages has been syntactical, and they tend to combine the recognizable expression and statement syntax of C with underlying type systems and data models that can be radically different.

Objective-C is influenced by C, Smalltalk. It is a powerful set of extensions to the Standard ANSI C Language. It is designed & developed to give C full object-oriented programming capabilities, and to do so in a simple and straightforward way.JAVA, Objective-J & other languages have some features of it. C++ is influenced by C, Simula, Ada, etc. C adapts the features of classes, inheritance, etc from Objective-C. C++ is firstly known as C with classes, later it gets its name C++. JAVA, ADA, PHP, C99, C# languages are influenced by it.

4. COMPARATIVE STUDY OF LANGUAGES

In early times programming or coding is done with the help of binary digits. All the characters have their binaries. In those days coding is very difficult because all the coding is done on binary or in assembly language. In these languages memory register is directly used so remembering all the register names and commands and binary values is not an easy work. If any changes a programmer wants to do in that program then doing changes is much more difficult from making a new program. So programmer's requires a language which minimise their task and make coding easier for them. After some time C Language was derived in 1970's the purpose of discovering this language is to make programming easier for programmer. In C Language simple English is used for coding. No binaries as well as no need to remember the name of register. In C Language syntaxes are written in very simple language which is easy to understand as well as to remember. C Language is derived from or influenced form the Assembly Language so it is machine oriented or machine dependent i.e., the program can only run in that system on which it is design (Table 1). It cannot be port to any other system which is having different configuration from the system on which it is design. Other problem is that C Language is procedural language i.e., it has its set rule for defining and declaring all the variables, functions, etc... For example first line of void main () is always used for variable declaration. All the variables are only declared in this line only not else anywhere in the program. By developing C Language problem of writing language is solved but now some new problems are arises. To overcome these problems a new language is derived in 1980's that is Objective-C Language. Objective-C Language is object oriented language. In this language no rules are described. In this language there are no rules for name of variable, for declaring variables or for doing any task. In this language printing is done just by writing the sentence in square brackets. In this language all the work is done in classes that why it an object oriented language. But this language is not easy to learn or understand because in this there are no set rules for defining any variable or function. But the advantage of this language is that it gives the concept of objects. By using programming becomes very easy.

To overcome problems of Objective-C language new language is derived in 1980's. With concept of object oriented and some predefined rules or functionalities a language is designed known as C with classes. Later this language gets its originals name i.e., C++. C++ is a language which is semi dependent on machine and which is easy to understand. C++ is different from C in terms of syntaxes, header files, some rules and many more things. For example: - In C if return type of main ( ) is declared integer (int) then return keyword must be used to return a value but in C++ no need to use return keyword.C++ uses the objected oriented concepts that why it is also known as Object Oriented Programming Language. In this language
all the work is done in classes so making changes are much easier from C language.

5. CONCLUSION

This paper studies about the languages, their history, and influence of one language to another. The purpose of developing C Language is to give programmer a coding friendly language. But this language has some drawbacks also. To overcome these drawbacks new language Objective-C is derived. It is influenced by C. Objective-C also has some issues to overcome these issues C++ is derived. It has features of C and Objective-C both. Since all these languages are derived from one another but they have some dissimilarity also.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 44 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++.

Table 1
Comparative study of the languages

Basis	C	Objective-C	C++
Paradigm(s)	Imperative (Procedural), Structured	Reflective, class-based object oriented	Multi-paradigm, procedural, object-oriented, functional, generic
Appeared in	1972	1983	1985
Designed by	Dennis Ritchie	Brad Cox &Tom Love	BjarneStroustrup
Developer	Dennis Ritchie & Bell Labs(creators); ANSI X3J11 (ANSI C);ISO/IEC JTC1/SC22/WG14(ISO C)	Apple	BjarneStroustrup Bell Labs ISO/IEC JTC1/SC22/WG21
File name extension	.h, .c	.h, .m, .mm	.h, .hh, .cpp, .hpp, .cxx, .c++
Syntax	Very complex	Easiest	Moderate
Inheritance feature	No such feature	Supports all types of inheritance excluding multiple inheritance	Supports all types of inheritance
Emphasis on	Functions	Both(function & data)	Data
Function Declaration	Functions can be declared at the point of use	Totally class-based so function are declared in the classes	Functions must be prototyped before main
Concept of overloading	Doesn't support overloading	Supports overloading	Supports overloading
Header files	Need to include them	No need	Need to include them

Concept of garbage collection	For de-allocating memory free( ) is used	Automatic garbage collection	For de-allocating memory delete ( ) is used
Keywords	32	12	98
Use of return keyword	Main( ) doesn’t automatically returns zero(0)	No concept of main( ) & return	Main( ) automatically return zero(0)

SUMMARY OF RESEARCH

The paper is regarding the cumulative studies of three well known languages i.e.C,C++and Objective C.It comprises of all the characteristics ,advantages and disadvantages, and the developments of all the three languages . Also the paper helps to compare the languages on various aspects and helps us to decide where to use which language so as to obtain more potential output.

FUTURE ISSUES

1. More developments in the field of languages and development work may add or deduce the comparisions.
2. Support for handling memory security vulnerabilities inlow-level programming. Can the pain in programming bereduced, and can ad hoc techniques be eliminated?

DISCLOSURE STATEMENT

There is no financial support for this research work from the funding agency.

ACKNOWLEDGMENT

Much thanks to our guide for his constructive criticism, and assistance towards the successful completion of this research work.

REFERENCE

1. A Brief History of the C Language. Available athttp://hubpages.com /hub/A-Brief-History-of-the-C-Language (Accessed May 25th, 2010)
2. A Brief History of C++. Available at http://hubpages.com/hub/A-Brief-History-of-the-C-Language (Accessed May 25th, 2010)
3. Online C++ tutorial, A Brief History of C++, available at http://www.intap.net/~drw/cpp/cpp01_03.htm (Accessed May 25th, 2010)
4. 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, 23-39
5. Brian W Kernighan, Dennis M Ritchie. Chapter 2; Types, Operators and Expressions in the C programming Language, Prentice-Hall Publishing, 1988
6. 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).
7. Biancuzzi Federico, Warden Shane. Masterminds of Programming. O'Reilly Media, Inc. 2009, 242–246, ISBN 0-596-51517-0
8. Cox Brad. The object oriented pre-compiler: programming Smalltalk 80 methods in C language. ACM SIGPLAN Notices (New York, NY: ACM), 1983, 18(1).doi: 10.1145/948093.948095. Retrieved 2011-02-17