**Language Comparison**

**650046219**

**Abstract**

The purpose of this essay is to outline, compare and contrast the C-family of computer programming languages: C, C++, C# and Objective C. This is done by a brief history of each language, the current applications of each language and the reason why each language is used and best suited for the given situation, an explanation of pros and cons with comparisons for clarity and a chart comparing features of these languages against other popular languages.

**Word count**

1563 (excluding chart)

I certify that all material in this dissertation which is not my own work has been identified.

**Origin of c family**

C

The C programming language has grown to be one of the most influential computer languages. A previous computer language, B, became the main inspiration for C. The reason for the development of B was originally due to problems with accessing data types on newer hardware, namely the PDP-11. At the very start of the 1970’s Brian Kernighan and Dennis Ritchie began working on B which had evolved into New B (NB) and then C. This name was chosen simply as it is the next letter after B. C was originally bundled with a new version of UNIX in 1972. Many C standards have been developed and improved in this until today.

C++

One of the directions the improvements to C were to include aspects of the Simula languages. Simula and B left a niche in terms of speed and practicality for larger software engineering projects and a middle ground was sought by Bjarne Stroustrup at the end of the 1970’s which lead to the development of C with classes which became known as C++ in 1983 as ++ is the increment operator in C. C++ has had less iterations and versions than C although the next major update is due at the end of 2017.

C#

The start of 2000’s saw the development of the Microsoft backed .NET Framework which was initially based on a version of C named Simple Managed C or SMC. Anders Hejlsberg formed a team and designated the language they were working on as COOL which stood for C-like Object Orientated language. The name C sharp refers to the sharp symbol in musical notation which is used when a standard note is to be played a half tone higher. Since its first official release in 2002, C# has progressed to version 7 which is part of `Visual Studio 2017. C# has notably been described as an imitation of Java.

Objective-C

In the early 1980’s Brand Cox and Tom Love began the development of enhancing C to solve issues such as code re-use with solutions gained from the Smalltalk Language. Originally Named OOPC for Object-Orientated Pre-Compiler. In the mid 1980’s Objective-C was described in a book released by Cox and over the next couple of years the GCC compiler was extended to support Objective-C based on the NeXTSTEP language. This language was instrumental in some of Apple’s core libraries.

**Applications and rationale**

C programming is used primarily in Operating systems as its use of pointers and memory management makes it a powerful tool for programming kernels and a variant is used in almost all operating systems. Its small size and portable nature makes it ideal for embedded systems such as car on-board systems, home electronics and even internet-of-things enabled devices. Another reason why C is suited to this type of use is it has a very small runtime and relatively small memory footprint. Compared to C++ the size of a compiled C program can be half the size.

C and C++ are also widely used in the programming of popular database languages. As Databases can be found in practically every industry, so can the use of C and C++. C and C++ are also used in the development of 3D movies as they are efficient at handling vast calculations per seconds. This saves movie produces time and money as the films can be animated or rendered much faster. C++ is also used in game development as it is adept at handling graphics as well as desktop applications for consumers or research.

Given C# and Objective-C’s development history, it should be of little surprise that its main use is in developing Microsoft and Mac applications respectively. C#’s versatility has seen it be used in programming servers and even the Raspberry Pi. C# is also predominately used in the development of web apps and can be used for cross-platform development of mobile apps. Objective-C has been primarily used for Apple programs, both desktop and mobile. It is currently being phased out in favour of a new language Swift which is rapidly replacing its use and has of this month entered the top 10 programming languages for the first time. As Apple owned the precursor language to the Objective-C and Apple’s corporate strategy of developing both hardware and software, Objective-C sees little use outside of apple products although with the change over to Swift, its future could take on a new direction.

**Analysis and comparison**

C advantages

One of the biggest advantages of C is that it strikes a good balance between low level control of a computer, while maintaining readability. C has become so popular because of its portability; its compiled or executable code will return the same behaviour on different architectures. One of the unique features of the C language is its use of pointers to refer to memory addresses which are useful in having data persist while the program is not running and can be more reliable when working with changing variables.

C disadvantages

Out of the C family of languages, C alone does not include Classes which are increasingly seen as a fundamental part of a modern programming language. Having classes allows for more complex data structures to be modelled and reduces the need to rewrite code. Having such low level access to computers memory can cause errors in regard to memory management. The C language also lacks runtime checking of code and simple exception handling.

C++ advantages

C++ extends C so that many C programs can compile and run as C++. C++ is a higher level language than C: machine-level code is abstracted away in favour of readability. C++ takes advantage of many Object-Orientated benefits such as objects, classes inheritance and overloading. C++ and C# are the only languages in the C family which natively support Namespaces. This is used to have keywords overridden or to use the same variable name in different circumstances.

C++ disadvantages

One major omission in terms of OOP in C++ is that multitasking is not included. With multiple processors the modern standard, even in mobile phones, performing tasks iteratively does not take advantage of modern hardware. Another Disadvantage of C++ is the lack of security features and support for garbage collection.

C# advantages

One of the biggest advantages of using C# is its similarity with Java which is currently the only programming language more widely used than any other in the C-family. C# also addresses the issues of memory leak and includes functionality for garbage collection which allows developers to spend less time concerned with allocating and de-allocating memory and also keeps objects from accidently using the contents of another object.

C# disadvantages

Some of the disadvantages from using C# come not just from the language, but from licensing issues. C# requires the .NET framework to be installed on the same machine to run and some parts of the language are not open-source. By needing the framework in order to run this limits the portability and performance of C#. In terms of the language, a possible drawback could be seen in that multiple inheritance is not permitted without a workaround.

Objective-C advantages

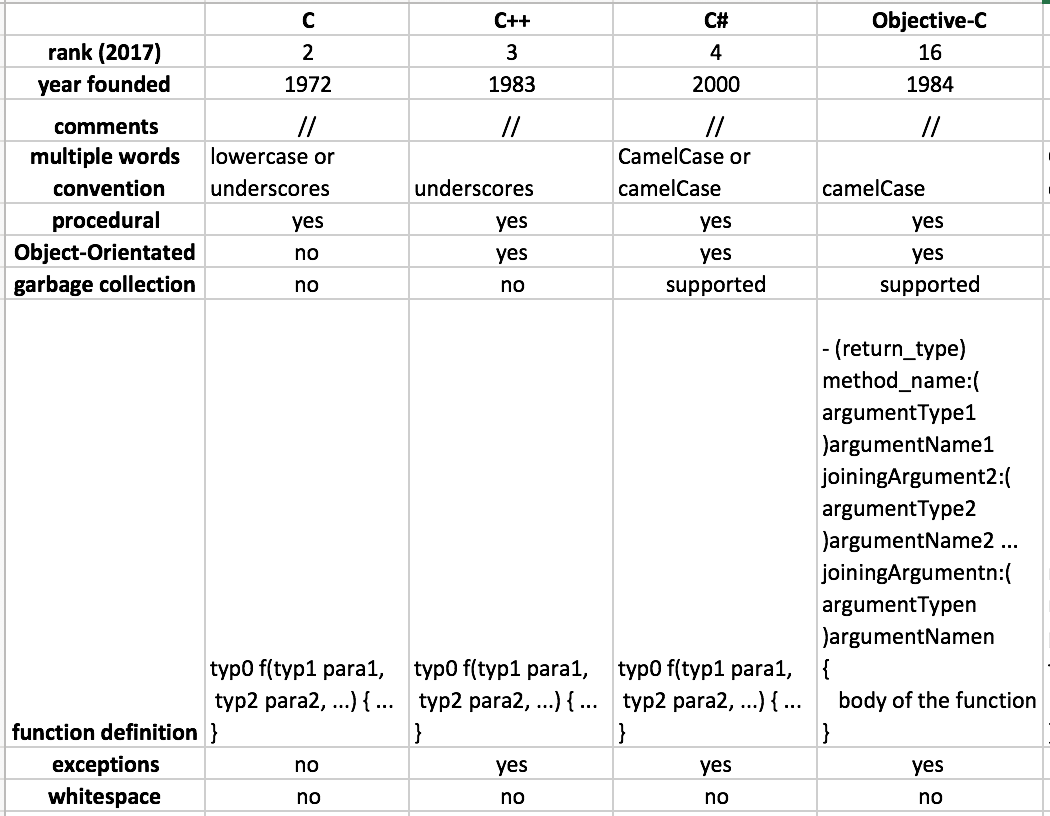
Until the last few years, Objective-C has been the *de facto* programming language of Apple products such as Mac and iPhone. The popularity of these platforms ensure lots of documentation is published and library support for developers. Similar to Microsoft backed C#, Objective-C takes advantage of the Cocoa framework which was previously NextStep and also can bind to other languages such as Python and Ruby. Objective-C is credited for being a dynamic language in that languages such as C and C++ are compiled languages which requires that code is translated into machine code to become executable. OOP is implemented in Objective-C can take advantage of C code and even wrapped to give the behaviour of objects.

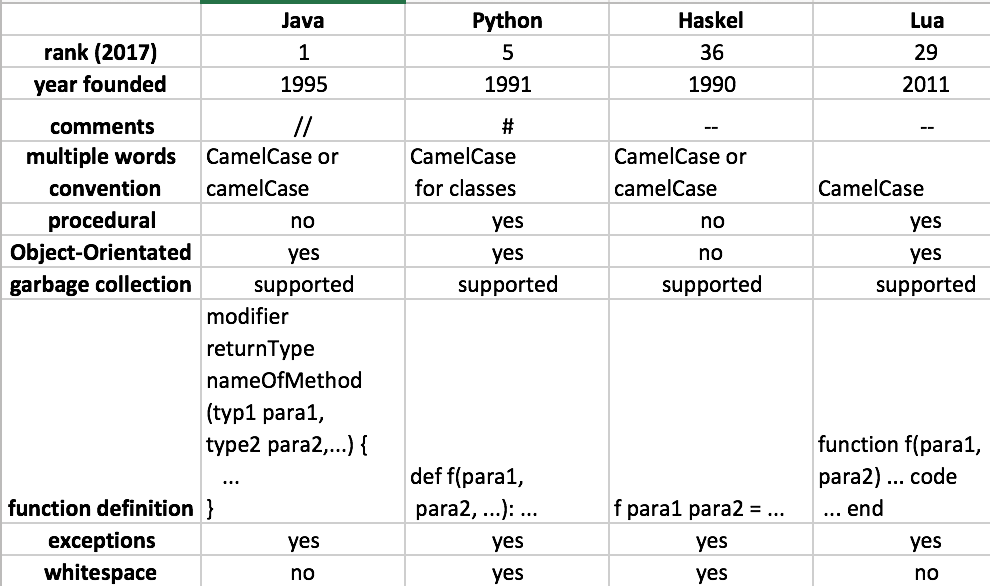
Objective-C disadvantages

Unlike C++ and C#, Objective-C does not support the concept of namespaces natively. Namespaces are useful when importing multiple libraries as functions within a namespace exist only in that scope which allows identically named functions or variables to call without interfering with the behaviour of each other. Similar to C#, multiple inheritance is not supported and also has its portability hampered by its reliance on the Apple backed Cocoa framework.

**Language comparisons**

This chart shows some of the similarities and differences of features between the C-family languages and other modern or popular languages.



****

**Conclusion**

C is informally known amongst Computer Scientists as ‘the father of all programming languages’ due to the fact that it the programming language that has influenced the most other languages and its ties to the UNIX operating system which has also formed the basis of modern operating systems. Most modern languages have been inspired in some form by C in terms of syntax, control structures or data models. As its closest variants have been described and detailed earlier, the limitations of the language have inspired different approaches in the attempt to improve C and restrict its limitations.

The inclusion of classes in the other C-family languages allows for better encapsulation and representations of real world entities, increases efficiency and allows more powerful concepts such as inheritance and generic programming to be utilised while increasing readability. This trade-off comes at the cost of limiting low level control to such an extent that C is still hugely relevant to many computing applications for the foreseeable future.

**References**

1. D. M. Ritchie, The Development of the C Language, Research Paper, 03 1993.

2. B. Stroustrup, A History of C++: 1979-1991," Research Paper, 20 04 1993.

3. B. Stroustrup, Programming: Principles and Practice Using C++, Developer’s Library, Addison-Wesley, 2009

4. J. Skeet, C# in Depth, Manning Publications, 2010

5. S. G. Kochan, Programming in Objective-C, Addison-Wesley Professional, 2011

6. L.Hunt, Coding Standards for .NET, Standards Reference, 2007

7. ISO/IEC 9899:TC2, Programming Languages - C, ISO Document, 2005

8. ISO/IEC 14882:2014 Programming Languages - C++, ISO Document, 2011

9. Object-Oriented Programming with Objective-C, Apple Inc., Developer Reference, 2010

10. .Net Framework 4.6 and 4.5, Microsoft, Developer Reference, 2017

11. M Pinkerton, G Miller , D MacLachlan, Google Objective-C style guide, Developer reference,