Programming Languages Overview, CSCE 314

Carsten Hood, UIN: 922009787

September 18, 2015

1 Introduction

Programming languages progressed rapidly over the latter half of the twentieth century, culminating in a series of powerful, high-level languages developed near the turn of the millennium. By tracing the origins and features of a dozen notable programming languages over this period, one may begin to understand at least a portion of the complex web of relationships connecting them.

Of the various programming paradigms that have emerged in the evolution of modern computer languages, this report will focus on two in particular—the object-oriented paradigm, pioneered by SIMULA 67 and cemented by many later programming languages, notably C++ and Java; and the functional paradigm, exemplified by the pure, lazy language Miranda and its popular successor Haskell. These two paradigms—object-oriented and functional—are not mutually exclusive, as demonstrated by Scala, the youngest of the languages surveyed; and many other methodologies and styles exist and coexist.

Also discussed are more specialized programming languages, including ALGOL and the later MATLAB, both designed for advanced numerical computations, along with SQL, a standard in database interaction. Most of the programming languages mentioned in this report, if not at least sustained by teachers and researchers, are commonly used in industry today.

2 Programming Languages

2.1 ALGOL

ALGOL is a programming language developed to perform scientific calculations. A committee of western scientists developed the first version in 1958, and subsequent standards were established in 1960 and 1968. Primary focuses of the language include portability and universality [21].

ALGOL was popular among researchers. It had a significant impact on the syntax of later programming languages, introducing the block structure and providing for if-statements and iteration. The language features reserved keywords, user-defined data types, and dynamic arrays [21].

2.2 SIMULA

SIMULA was developed in the 1960s by a pair of Norwegian computer scientists, Kristen Nygaard and Ole-Johan Dahl, as a means of performing computer simulations [24]. The language is largely a descendant of ALGOL 60 [9].

The first version of SIMULA gained popularity in Europe for performing simulations; however, it soon became apparent that an improved, more versatile version could be useful in more general programming applications. A second version of the language, SIMULA 67, launched the object-oriented programming paradigm. It introduced the terms “class” and “object”, as well as the notion of inheritance and subclassing [24].

SIMULA 67, due to its revolutionary nature, inspired a host of other object-oriented programming languages. Both C++ and Smalltalk were directly inspired by SIMULA 67 [23].

2.3 C

In 1972 Dennis Ritchie developed the programming language called C to replace assembly language in implementing the Unix operating system. The C language was designed to retain many of the low-level capabilities of assembly language, while providing for more versatility and efficiency of implementation [7].

C achieved widespread use as a replacement for assembly language in various applications, including the coding of operating systems and of compilers and interpreters of higher-level languages [8].

Now C is recognized as one of the most popular programming languages of all time [26].

2.4 SQL

Structured Query Language, better known as SQL, was released in 1974 [4]. The language was designed as a means of interacting with databases. It still the most widely used language in database systems [11].

SQL provides basic commands for communicating with a database, and is capable of most required database interactions. It is often extended for use in specific implementations, as in Microsft SQL Server, Access, and Oracle [11].

2.5 C++

Inspired by his experience with SIMULA 67, considered the first object-oriented programming language, Bjarne Stroustrup worked to augment the powerful and widespread C language with object-oriented features. In 1983 he released the first version of C++. Subsequent improvements to the C++ language, including a new standard in 2011, have solidified its place in the field of computing [2].

Like its predecessor C, C++ is an imperative language. Furthermore, it features classes and inheritance and is capable of generic programming. C++ also boasts a robust standard library and continued development, with another update expected in 2017 [2].

Since its creation C++ has been widely used across various technologies. Due to its efficient nature, it is especially popular in resource-constrained or performance-critical systems and applications [25].

2.6 MATLAB

MATLAB was developed by MathWorks in 1984. It was designed to perform advanced numerical computations and linear algebra, and soon spread into other domains [6].

The term MATLAB stands for matrix laboratory. It encompasses a programming language as well as an application and a special development environment for writing programs. The language is object-oriented; many consider its features to be similar to those of C++. It also allows for efficient matrix and vector calculus [10].

Today MATLAB is used by millions of scientists, researchers and engineers around the globe [17].

2.7 Miranda

Computer scientist David Turner developed Miranda in the mid-1980's. His objective was to produce a simple and flexible functional programming language for commercial use [22]. Miranda gained popularity among researchers. Today it is used in the teaching of functional programming languages [19].

Miranda is pure, lazy, and non-strict. It also features polymorphic capabilities. Miranda heavily influenced the design of later functional languages, notably Haskell [19].

2.8 Haskell

Intent on creating a standard functional programming language, a group of researchers developed Haskell in 1990. The language has since been enhanced by subsequent standardizations and community support. It has seen widespread use in teaching and researching functional languages, as well as in industry for easier, safer, and more efficient use over imperative programming languages [13].

Haskell is pure, meaning its functions are similar to mathematical expressions; lazy, meaning functions do not evaluate their arguments; and statically typed, meaning the type associated with each expression is determined at compile-time. These and other features streamline the language and serve to limit errors and unnecessary code [14].

2.9 Python

Guido van Rossum first released Python in 1991. Python was initially intended to succeed the programming language ABC, and was designed with an emphasis on readability and efficiency of development. Today Python is widely used for various applications, such as scripting, artificial intelligence, and information security. It allows for efficient, high-level application development [27].

Python is object-oriented and features dynamic semantics. It also features high-level integrated data structures. Modules, packages, and a powerful standard library allow for easy reuse and portability [12].

2.10 Java

Computer scientists at Sun Microsystems developed Java with the goal of superseding C and C++ with a cleaner, higher-level, and more efficient programming language. The first version of Java was released in 1995. Frequent improvements have come every few years since [5].

Java adheres to and even furthers the object-oriented programming paradigm. Notably, the Java language is platform-independent and programs can generally run across operating systems, giving rise to the slogan "Write once, run anywhere" [18].

Today Java is often ranked as the most popular programming language, having surpassed both C and C++ in the TIOBE Index, which ranks programming languages by popularity. Java is especially widespread in web applications due to its portability. Google has also adopted the Java language for applications running on the Android mobile operating system [26].

2.11 C♯

C♯ was initially developed by Microsoft to be used with the .NET platform. Its design philosophy focused on simplicity, object-oriented capabilities, and type-safety. The language was first released in 2000. C♯ has seen widespread use among Microsoft technologies and has become a competitive alternative to Java in many applications, including operating systems, compilers, and video games [3].

C♯ is a member of the C family of programming languages, and thus is closely related to C, C++, and Java. Furthermore it supports many varied programming methodologies, including functional and generic styles. Its chief designer, Anders Hejlsberg, considers C♯ to most closely resemble an improvement of C++ [15].

2.12 Scala

Having contributed to the improvement of Java but feeling limited by its constraints, Martin Odersky began work on a new programming language. In 2004 a public version of Scala was released. Being nearly a decade newer than Java, the Scala language incorporated modern features either lacking in or poorly assimilated into Java [28].

From the start Scala integrated functional and object-oriented capabilities. While all values in Scala are considered objects and operations are executed through calling methods, functions with a preference towards immutability can also be used. Scala features type-inference and a simple syntax. Scala was also designed to utilize Java libraries and to coexist with Java code [20].

Recently Scala has gained popularity as a modern programming language for general software development. Many large technology companies, including Twitter and Intel, use Scala for critical backend systems [20].

3 Programming Language Comparison

3.1 Programming Paradigm

3.1.1 Object-Oriented Languages

Most languages described herein are object-oriented, being based around instances of objects associated with defined class types. SIMULA 67 is typically regarded as having introduced the object-oriented paradigm, directly inspiring the development of C++. Python, Java, C♯, and Scala are all primarily object-oriented.

3.1.2 Functional Languages

Functional languages are often contrasted with imperative languages. In functional programming, programs comprise mathematical functions and emphasize higher-level operations and immutability. In so-called pure functional languages expressions produce constant values and once created data cannot be changed. Miranda was an early pure functional language; it heavily influenced Haskell, another pure functional language widely in use today [1].

While Scala also allows for functional programming, it is considered an impure functional language because it is primarily object-oriented and does not mandate immutability. Java, which has incorporated functional capabilities, may also be regarded as an impure functional language.

3.2 Popularity

Popularity is one of the best metrics for determining the real-world value of a programming language. Every day engineers and researchers determine what systems and technologies to use to optimize program development and user experience, thereby casting their votes in the marketplace of programming languages.

The popularity of a programming language can be gauged by various means. The number of projects, programs, or even lines of code estimated to be written in a certain language may be considered, as well as the number of engineers guessed to use or know the language. The TIOBE Index amasses data from twenty-five online search engines, among other factors, to determine the popularity of different programming language [26].

According to the TIOBE Index as determined in September of 2015, Java is the most popular programming language. Its popularity is rated at 19.6% of all languages. Java is widely used in computer education and web and application development, all markets that expanded rapidly after its inception. Java is notable for its portability [16].

The C programming language is ranked next in popularity, at 15.6% [26]. C is still commonly used for low-level applications, such as programming operating systems and compilers, and in embedded systems. While many newer languages have focused on distancing developers from hardware, C has retained its place as a powerful low-level language [16].

Ranked behind C and Java are C++ (6.78%), C♯ (4.9%), and Python (3.7%), all general-use, object-oriented programming languages common in industry. Notably, C++ has steadily declined in popularity over the past decade, presumably due to the advent of similar but more modern languages like Java. Scala, a much more recent development, has gained popularity as developers and companies adopt it for various uses, though it is still ranked only twenty-seventh.

MATLAB, ranked seventeenth, maintains its role in advanced mathematical calculations. Haskell, ranked thirty-eighth, is common in research and education, and also enjoys some use in industry, while its predecessor Miranda has been largely confined to the study of functional language in academic circles. Similarly, both ALGOL and SIMULA have given way to their successors in real-world applications, and are now enshrined in the fast-moving history of computer science.

References

[1] Keith Adams. The best languages for getting into functional programming, August 2014.

[2] Albatross. History of C++. http://www.cplusplus.com/info/history/, 2015.

[3] David Bolton. All about the C# programming language. http://cplus.about.com/od/introductiontoprogramming/p/profileofcsh.htm.

[4] Chris Collins. History of SQL. https://ccollins.wordpress.com/2007/05/20/history-of-sql/, May 2007.

[5] Wikibooks contributors. History of the Java programming language. https://en.wikibooks.org/wiki/Java_Programming/History, July 2015.

[6] Wikibooks contributors. MATLAB programming. https://en.wikibooks.org/wiki/MATLAB_Programming, September 2015.

[7] Wikipedia contributors. The C programming language. https://en.wikipedia.org/wiki/The_C_Programming_Language, September 2015.

[8] Wikipedia contributors. C (programming language). https://en.wikipedia.org/wiki/C_(programming_language), September 2015.

[9] Ole-Johan Dahl. The birth of object orientation: the Simula languages. http://staff.um.edu.mt/jskl1/talk.html, June 2001.

[10] Altius Directory. MATLAB programming language. http://www.altiusdirectory.com/Computers/matlab-programming-language.php, 2015.

[11] Quinstreet Enterprise. What is SQL? http://www.sqlcourse.com/intro.html, 2015.

[12] Python Software Foundation. What is Python? Executive summary. https://www.python.org/doc/essays/blurb/, 2015.

[13] fpcomplete.com. The benefits of Haskell. https://www.fpcomplete.com/business/about/about-haskell/, 2015.

[14] Haskell.org. Haskell homepage. https://www.haskell.org, 2015.

[15] Anders Jehlsberg. *The C# Programming Language*. Microsoft Corporation, 2011.

[16] Larry Kim. 10 most popular programming languages today. http://www.inc.com/larry-kim/10-most-popular-programming-languages-today.html, June 2015.

[17] MathWorks. MATLAB. http://www.mathworks.com/products/matlab/, 2015.

[18] Norman Matloff. A quick, painless introduction to the Java programming language. http://heather.cs.ucdavis.edu/~matloff/Java/JavaIntro.html, March 2003.

[19] Miranda.org. Miranda. http://miranda.org.uk, December 2010.

[20] Martin Odersky. What is Scala? http://www.scala-lang.org/what-is-scala.html.

[21] University of Michigan. The ALGOL programming language. http://groups.engin.umd.umich.edu/CIS/course.des/cis400/algol/algol.html, November 1996.

[22] University of Michigan. Miranda programming language. http://groups.engin.umd.umich.edu/CIS/course.des/cis400/miranda/miranda.html, December 1997.

[23] Progopedia. Simula 67. http://progopedia.com/language/simula-67/, 2015.

[24] J. Sklenar. Introduction to OOP in SIMULA. http://staff.um.edu.mt/jskl1/talk.html, 1997.

[25] Bjarne Stroustrup. *The C++ Programming Language*. Addison-Wesley, July 2013.

[26] TIOBE Software. TIOBE Index for September 2015. http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html, September 2015.

[27] Bill Venners. The making of Python. http://www.artima.com/intv/pythonP.html, January 2003.

[28] Bill Venners. The origins of Scala. https://www.artima.com/scalazine/articles/origins_of_scala.html, May 2009.