**PROGRAMMING LANGUAGES AT THE VERGE OF DYING**

* Tania Aggarwal

It’s generally and very truly said that Nothing lasts forever, and programming languages are no exception. No matter how ‘cool’ or ‘hyped up’ a programming language is, eventually the better ones take over and today’s ‘favorite’ fades away.

Since the beginning, there have been hundreds of programming languages, of which some have faced the test of time pretty well and are still being widely used while the others didn’t fare very well. They just came, ruled, and faded away. Among all the various possible reasons for this inevitable decline, one can be attributed to the fact that with time, new generations of developers embrace other languages and frameworks they find easier to work with.

Below given are the few programming languages, in no particular order, that are mostly dead or are likely to die in the coming few years.

**Objective C**

Developed by Brad Cox in 1983, Objective C is a superset of the C programming language and provides object-oriented capabilities and a dynamic runtime. The language follows the Smalltalk-derived syntax and it’s notable for being the primary language used by Apple for both iOS and macOS. The language single-handedly ruled the particular domain until the arrival of its alternative language Swift in 2014. Since then, the preference of swift over Objective C by the developers due to its various remarkable features like robustness, better memory management, et cetera has led Objective C to find a place in the list of dying programming languages.

Having said that, it’s been 36 years since Objective C came around, and the language still has some occupancy in the market. Hence of which, it’s probably gonna take a few more years for Swift to completely take over.

**Perl**

Perl is a family of two [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language), [interpreted](https://en.wikipedia.org/wiki/Interpreter_(computing)), [dynamic programming languages](https://en.wikipedia.org/wiki/Dynamic_programming_language). The language is used for a wide range of tasks such as web development, text manipulation, GUI development, network programming, and many more. Originally developed by Larry Wall in 1987, Perl has the features that ease the task of the programmer, but it comes at the expense of greater CPU and memory requirements. This resulted in its increasing disuse.

The latest sister version of Perl - Perl 6 (also known as Raku) was released in December 2015. This release did revive the language somewhat, but by then, the advent of several other prominent programming languages especially Python made it difficult for Perl to regain control over the market.

**Pascal**

Pascal is a general-purpose, procedural programming language that was specifically developed for teaching programming practices in a structured manner. Descended from ALGOL 60, Pascal is the creation of Niklaus Wirth. It’s named after the French mathematician [Blaise Pascal](https://www.parkersoftware.com/blog/little-acorns-humble-history-computing/), who invented the first mechanical calculator.

Pascal led to the development of Delphi (Object Pascal), which in general can be considered as an object-oriented derivation of Pascal language.

The language enjoyed its popularity and demand in the market due to various features such as easy to learn, extensive error checking, strongly-typed, etc. But Delphi soon took the place of Pascal, condemning it to the list of dead programming languages. However, considering its growth and demand in the current scenario, Delphi itself is on the verge of vanishing now.

**Haskell**

Haskell is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language), [statically typed](https://en.wikipedia.org/wiki/Static_typing), [purely functional](https://en.wikipedia.org/wiki/Purely_functional_programming) [programming language](https://en.wikipedia.org/wiki/Programming_language) with [type inference](https://en.wikipedia.org/wiki/Type_inference) and [lazy evaluation](https://en.wikipedia.org/wiki/Lazy_evaluation). The language was particularly designed for handling symbolic computations along with list processing applications and it is considered to be much relevant to the research and industrial application domain.

Also, it supports various features such as better reliability, shorter lead times, etc. which has earned it a preference in various big-tech companies like Facebook, GitHub, IBM, et cetera in the past.

However, a look at the TIOBE popularity index shows a consistent decline in its demand and usage in the past 8-10 years. This decline could be attributed to various factors like difficulty to learn, static typing, etc. This has lead to expect it to die in coming years.

**VB.NET**

Visual Basic .NET (VB.NET) is a [multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [programming language](https://en.wikipedia.org/wiki/Programming_language), implemented on the [.NET Framework](https://en.wikipedia.org/wiki/.NET_Framework). Visual Basic was developed by Microsoft as a variant version of the BASIC and then VB.NET was launched in 2002 as the successor to Visual Basic.

VB.NET has lived its fair share of popularity for a longer period of time but the latest TIOBE Index reports show that VB.NET is now losing its charm against its competitor language C# and that C# is taking over in terms of demand and popularity.

However, due to its strong command over the programming world in the past, VB.NET still has some adequate occupancy in the market and it’s probably gonna take years to die out.

It’s nothing wrong with the programming languages which make them disappear. It’s just that with evolving technology, developers find alternative better languages to catch up with the trend. The truth is that programming languages seldom really die, they just become outdated. As a developer, it’s our responsibility to analyze the trends, to understand the demand of the tech market, to know the current and future scope of the languages in the IT world, and adapt accordingly!