An Analysis of the Usage of Various Programming Languages to Classify Whether they are Relevant, Extinct, or on the Verge of Extinction

Apoorv Mohit[1] *and Sunaina Rustagi*[2]

Abstract

The first modern high level general purpose programming languages appeared over seventy years ago. Since then, with the exponential increase in the usage of different types of computer software and developments in computer hardware, numerous high level programming languages have been created for various purposes, with each language having its own relative advantages and disadvantages. Usually, specific general purpose high level programming languages are appropriate for specific applications or problems because of the nature of capabilities of the languages such as efficiency, memory consumption, expressiveness, availability of compilers, and tools etc. Newer languages incorporate significant functionalities of older languages which may serve as an "inspiration" for it while introducing new features and functionalities. However, the creation of newer programming languages does not necessarily mean the obsolescence of older languages because of factors such as programming effort, familiarity, and popularity. However, their usage for newer and upcoming applications may eventually decrease and familiarity and popularity might consequently fade away with it. In this paper, we take a look at 38 different programming languages that have been invented and identify the least used programming languages to provide an overall estimate of the least used programming languages in today's time and the programming languages on the verge of death.

Keywords : **Programming languages, usage**

I. INTRODUCTION

A large number of high level programming languages have been developed since the very invention of modern computers and continue to be developed today. Some languages were developed by individuals trying to create tools for specific functions, for example, C++ was created by Bjarne Stroustrup who initially wanted a tool for designing and implementing a distributed version of the Unix kernel. No such tool existed at the time in 1979. He needed something that could express the structure of a program, interact with the hardware, and be sufficiently efficient and sufficiently portable for serious systems programming [1]. While some languages were created by individuals to overcome the limitations of older languages such as Python, which was developed by Guido Van Rossum who wanted to enhance programmer productivity. He observed that ABC programming language had a much higher productivity than C, but had failed to gain popularity for a variety of reasons and was not being maintained. So, he started to develop and implement a language improving upon ABC's deficiencies [2], but some languages were created by a group of developers at an institution or a corporation. Fortran is the best living example of this statement, which was created by a group of engineers at IBM led by John Bakus. Fortran, which stems from "Formula Translation", began the process of abstracting software

Paper Submission Date : February 2, 2023 ; Paper sent back for Revision : February 20, 2023 ; Paper Acceptance Date : February 25, 2023 ; Paper Published Online : April 5, 2023

[1] A. Mohit (*Corresponding Author*), *Student* ; Email : mohitapoorv12@gmail.com ;
ORCID iD : https://orcid.org/0009-0005-3360-7436
[2] S. Rustagi, *Student* ; Email : sunainarustagi1101@gmail.com ; ORCID iD : https://orcid.org/0009-0005-8661-6723
[1,2] Maharaja Surajmal Institute of Technology, C-4 Market, Fire Station Road, Janakpuri, New Delhi -110 058

DOI : https://doi.org/10.17010/ijcs/2023/v8/i2/172776

from the hardware it ran on. Previous machine language programs had to be written for a specific computer but a Fortran program can be run on any computer with a Fortran compiler on it [3].

As we observed, the creation of new languages stems from various concerns, causes, and mainly problems encountered by developers and computer scientists. Since problems and the inability of current technology to handle some tasks are the origins of new languages, it is safe to state that a lot of new languages keep developing and older ones evolve. However, as we saw in the three cases discussed earlier, the development of new languages is not to replace or capture the base of other languages, but to overcome some problems which cannot be solved using the existing languages. Thus, it is safe to say that the mere creation of new languages is not the cause of obsolescence of another or an older language. Every language has its own relative advantages and disadvantages over other languages and that makes a language a better candidate than others for a specific task, which in turn keeps different languages relevant in industry as well as academia.

The present work tries to establish the least used but widely known high level programming languages and point out if any of the programming languages are on the verge of death. In addition, we also provide an insight into the popularity and familiarity of different programming languages reviewed in this paper by conducting a survey among the current higher education students of Computer Science, related fields, and working professionals.

II. LANGUAGES

In the current text, we analyzed 38 programming languages in total. For an easier comparison, analysis and understanding of the languages, they can be classified according to the decade of their invention or initial release.

III. RESEARCH METHODOLOGY

The present work utilizes websites, a survey conducted by the authors, global surveys and research papers to carry out relevant research and analyze the parameters fit for estimation of the least used programming languages and the ones on the verge of death.

IV. PARAMETERS OF ESTIMATION

To identify the relevance, popularity and utilization of the programming languages, we extensively researched the following five parameters:

1) Stack Overflow Developer Survey, 2021

2) GitHut 2.0

3) Employment Availability

4) The TIOBE Index

5) Survey Conducted through Google Forms

V. CONTRIBUTION WITH STACKOVERFLOW

Stackoverflow[1] is a website that was launched in 2008. It serves as a platform for users to ask and answer questions through active participation and upvote or downvote answers similar to Reddit[2] and edit the questions and answers in a way similar to Wiki.

TABLE I.

CLASSIFICATION OF LANGUAGES ON THE BASIS OF THEIR INVENTION

Decade	Languages
1950 – 1959	FLOW-MATIC, FORTRAN, LISP, COBOL, ALGOL, RPG
1960 – 1969	SIMULA, SNOBOL, BASIC, APL, PL/I
1970 – 1979	PASCAL, C, SMALLTALK, SQL, ML
1980 – 1989	C++, ADA, MATLAB, OBJECTIVE-C, PERL
1990 – 1999	HASKELL, PYTHON, VISUAL BASIC, RUBY, JAVA, JAVASCRIPT, PHP
2000 – 2011	ACTIONSCRIPT, C#, F#, GROOVY, SCALA, GO, DART, COFFEESCRIPT, SWIFT, KOTLIN

[1] https://stackoverflow.com
[2] https://www.reddit.com

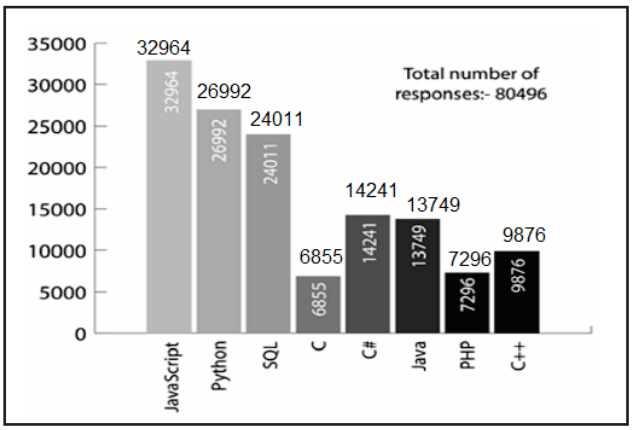

Fig. 1. StackOverflow Survey 2021
Usage of Different Programming Languages

StackOverflow Developer Survey 2021

There were several significant facts discovered in a survey conducted by StackOverflow [4] to learn which tools the current developers use, how they study and level up, and what they want from programming, as a career or hobby. In Fig. 1, there were 80,496 responses from developers (minimum 5,000 respondents per connection) for the languages they use for their work. Over 10,000 Javascript developers have expressed an interest in learning or continuing to develop in Go. JavaScript is currently used by the majority of developers who want to use Dart. We also see that SQL developers are the only ones who wish to work in PHP.

LISP, COBOL, APL, C, and SQL were among the few languages listed (Fig. 2) in the StackOverflow survey from 1950 to 1979. Only 1096, 437, and 536 people (0.01%, 0.005%, and 0.006%) utilized LISP, COBOL, and APL respectively out of 83,052 responses in 2021. This obviously demonstrates that these three languages are some of the least used programming languages in today's time. LISP has long been a popular language for Artificial Intelligence applications. Many of the essential notions of functional programming are still present in today's LISP dialects which include imperative style variables, assignment commands, and iteration [5]. In 2021, COBOL overtook all other languages as the most dreaded language, that is, that developers did not wish to work with it, with 84.21% of the respondents dreading it.

C and SQL on the other hand continue to be popular. C continues to empower the world despite the prevalence of higher-level languages. A kernel developed in C has been powering the world's most popular operating system for decades, with over 90% market share. The C language is also used in the kernels of LINUX, macOS, Android, iOS, Windows Phone, and other operating systems, which is one of the main reasons why C is still widely used. Oracle Database, MySQL, MS SQL Server, and PostgreSQL are just a few of the world's most popular databases. SQL is a declarative abstract language that allows us to work with data as logical sets independent of how it is physically stored, making it an excellent choice for users.

C++, MATLAB, Objective-C, Perl, Haskell, Python, Visual Basic, Ruby, Java, JavaScript, and PHP were among the languages in our decade band of 1980 – 1999. Visual Basic is the least popular in this category, and it is known that Microsoft planned to discontinue all versions of Visual Basic from 1.0 – 6.0 by 2008 (Fig. 3). This means that the Visual Basic development environment, as

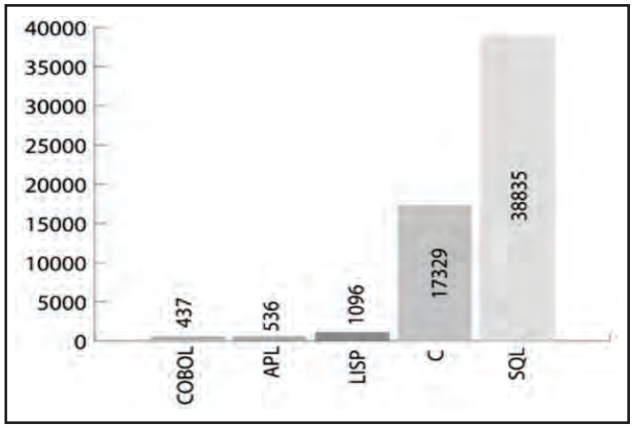

Fig. 2. StackOverflow Survey 2021
Number of users who worked with these languages

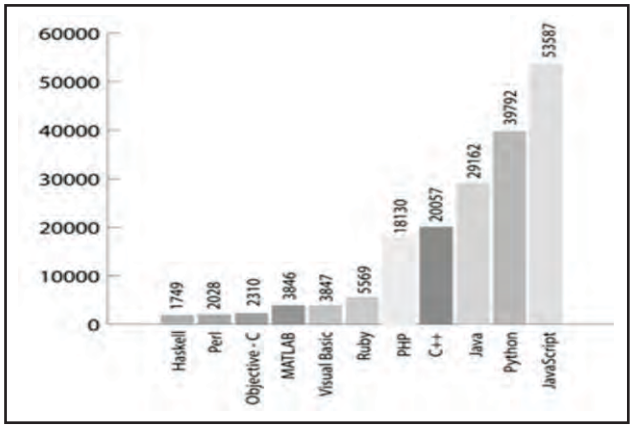

Fig. 3. StackOverflow Survey 2021
Number of users who worked with these languages

well as the runtime environments that go with it are no longer supported. As it is no longer developed by Microsoft, classic VIsual Basic is effectively defunct, but the latest release of Visual Basic, VB6 is still in use at some places.

According to the most recent scenario for 2021, Python was the most wanted language, that is, developers wanted to work with it for the sixth year in a row, followed by JavaScript which was the most popular language of the year. JavaScript completed its 9th year in a row as the most commonly used programming language. For most developers, programming is equivalent to web programming. Python traded places with SQL to become the third most popular language. The most popular languages after Python and JavaScript were C++ and Java. C++ is a programming language that is commonly used in competitive programming and game creation. Java, on the other hand, is a server-side language that is utilized in most back-end development projects.

Looking at the most recently released languages, that is, those released after 2000, C# has risen to great heights. C# is a popular language among desktop developers, but it is also the most widely used language among AR/VR and game developers, thanks to the Unity game engine's enormous popularity in these fields. Because of its simple syntax, F# will be the least used in 2021. This attitude toward the language by its creator is one of the reasons why F# has yet to get widespread recognition for its suitability for both front-end and back-end development beyond data science.

Kotlin was first released in 2011, and while it hasn't garnered much traction yet, it is steadily gaining popularity. The flawless compatibility of Kotlin with Java 6 was one of the key reasons for its success. This helped Kotlin establish itself as a viable Java alternative. Google launched Kotlin as a first-class programming language for Android app development in 2017. Go is an open-source programming language created at Google in 2007 and it is aimed at being simple, dependable, and efficient. However, Go has some very specific design flaws, and many users dislike how it handles errors.

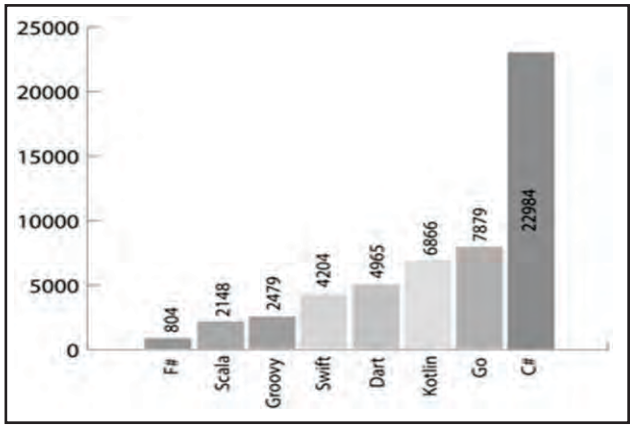

Fig. 4. StackOverflow Survey 2021
Number of users who worked with these languages

VI. CONTRIBUTION WITH GITHUB

GitHub [6] was launched in 2008 by Tom Preston-Werner, Chris Wanstrath, P. J. Hyett, and Scott Chacon who were the founders. It incorporates Git's distributed version control and source code management (SCM) features. Every project has access control as well as capabilities like bug tracking, task management, continuous integration, and Wikis. According to Wikipedia, GitHub has over 73 million developers and over 200 million repositories as of November 2021. (including at least 28 million public repositories).

GITHUB 2.0

GitHub is the largest code host in the world, with 40 million users and more than 190 million repositories as of January 2020. By analyzing how languages are used in GitHub it is possible to understand the usage of programming languages among developers. Fig. 5. depicts the number of pull requests made on GitHub throughout the course of the year. The percentage figures are the actual fractions of pull requests for each of the languages shown in the graphs.

Smalltalk appears to be approaching extinction when compared to early languages from 1950 – 1989. Smalltalk is a dynamic language that uses a virtual machine, making it extremely slow to use. Smalltalk was the first graphical programming language to offer live programming and advanced debugging. Fortran is a programming language that is used to do large-scale numerical computations. Because our understanding of what a good language design is and what qualities a language should have has grown so much, it can be said that it lacks many of the design aspects of modern languages. Despite its 66-year legacy, Fortran is still thought to be alive.

MATLAB, LISP, Perl, and Objective-C are also among the languages with the fewest pull requests

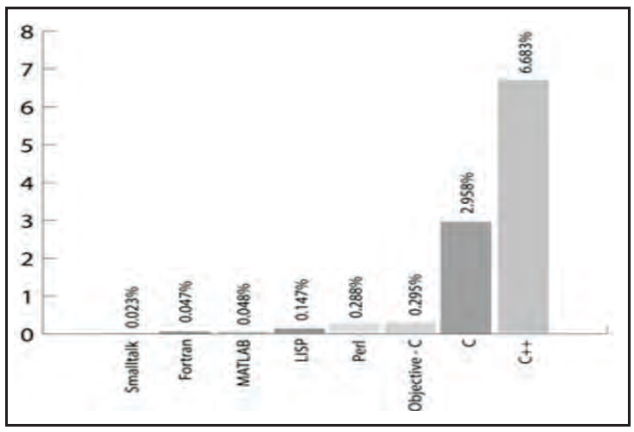

Fig. 5. GitHub Pull Request 2021 Percentage of pull requests made throughout the year in the particular languages

(Fig. 5). When it comes to C and C++, C++ was created as an extension of the C language. As it is one of the most widely used programming languages, C is often referred to as the "mother of all programming languages." C has been one of the most extensively used and favoured programming languages since its inception. Languages like Assembly, B (BCPL, CPL), ALGOL68, and FORTRAN impacted the development of C [7]. In the field of game development and competitive programming, C++ is extremely significant owing to its functionalities such as pointers, classes, objects, functions and efficient run time.

As Visual Basic continues to be one of the most dreaded languages of the year, the pull requests went up to only 0.042% of total pull requests (Fig. 6) followed by Haskell. Haskell is a useful language that isn't widely utilized. Haskell's steep learning curve may deter newcomers from adopting the language. Haskell is a "typeful" programming language; Types are pervasive, and the newcomer is best of becoming well-aware of the full power and complexity of Haskell's type system at the outset. For those whose only experience is with relatively "untypeful" languages such as Perl, Tcl, or Scheme, this may be a difficult adjustment. For those familiar with Java, C, Modula, or even ML, the adjustment should be easier but still not insignificant, since Haskell's type system is different and somewhat richer than most. In any case, "typeful programming" is part of the Haskell programming experience and cannot be avoided [8].

PHP was one of the first server-side languages to be embedded in HTML, making it easy to add functionality to websites without having to access data from external files. Java, Python, and JavaScript, on the other hand, continue to rise in all aspects. PHP is used by 81.7% of all websites whose server-side programming languages are known [9].

The most popular scripting languages, such as JavaScript and Python were used by the majority of users, according to StackOverflow. As Node.JS supports a "JavaScript everywhere" paradigm, integrating all web application development around a single programming language rather than having a separate language for server-side and client-side scripts, JavaScript's popularity skyrocketed. Python is a popular programming language among developers because it is simple to learn and use. Its syntax is also simple. Being the most popular does not imply that it is ideal, and there are some drawbacks, the

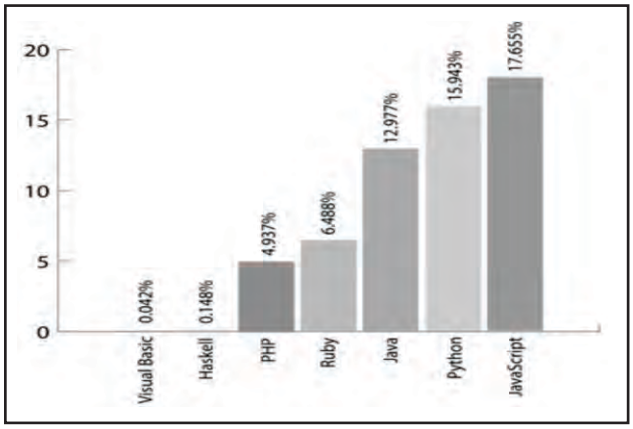

Fig. 6. GitHub Pull Request 2021 Percentage of pull requests made throughout the year in the particular languages

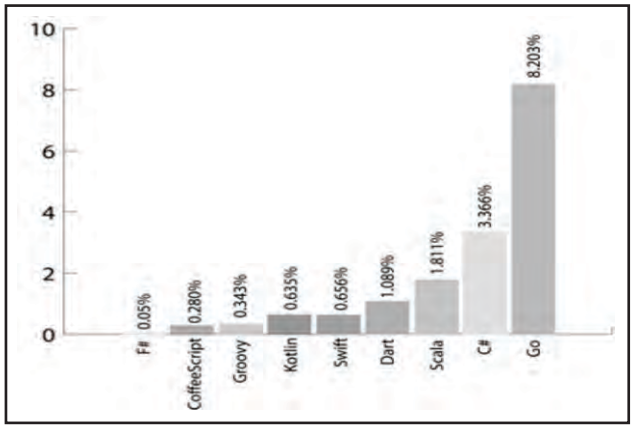

Fig. 7. GitHub Pull Request 2021 Percentage of pull requests made throughout the year in the particular languages

most notable of which is the slow execution speed and significant memory usage.

F# is most commonly used as a cross-platform Common Language Infrastructure (CLI) language on .NET, although it can also create JavaScript and GPU code. This attitude towards the language by its creator is one of the reasons why F# has yet to get widespread recognition for its suitability for both front-end and back-end development, beyond data science. In 2021, Kotlin, which was first introduced in 2011, was the best language for Android development. Although both Java and Kotlin can be used to create high-performance apps, Google's libraries, documentation, and learning resources continue to favor Kotlin, making it the preferred language for Android app development today. It is steadily increasing in the technological world and is expected to have a bright future.

Googlers introduced Go, which is also known as Golang. Go is thought to be productive and effective, as evidenced by its highest pull requests. C# is a fairly active programming language. It has been built to evolve and adapt to new technologies, and C# isn't going anywhere anytime soon. In Tiobe's rankings, it is presently the 6th most popular language. CoffeeScript is a little programming language that converts to JavaScript. CoffeeScript is no longer available on the market as of January 2020, though the GitHub repository is still active.

Scala fuses object-oriented and functional programming in a statically typed programming language. It is aimed at the construction of components and component systems. Components in this sense are simply software parts which are used in some way by larger parts or whole applications. Components can take many forms; they can be modules, classes, libraries, frameworks, processes, or web services. Their size might range from a couple of lines to hundreds of thousands of lines [10]. Scala is designed to be compatible with C# and Java.

However, the majority of the projects on GitHub were written in C, C++, Java, JavaScript, Ruby, PHP, and Python, with Haskell, Scala, and Go coming in last [11].

VII. EMPLOYMENT AVAILABILITY

Method

We reviewed Naukri.com [12], which is an Indian employment website that connects employers to job seekers for various companies within different industries.

We used "*Language Name Developer*" as our parameter for the desired search and then took the total number of results as our observation.

The rough estimate of the number of jobs currently in the industry is a direct indicator of the usage of a language. A language being currently used means more lines of code are being written, either for new projects or for maintaining or supporting the older lines of code for a particular language.

Analysis

The following tables show the number of job postings for various languages on Naukri.com as of 4 January, 2022.

Tables II to VII list the number of jobs posted online on Naukri.com by language. It is clear from these observations that languages which were developed during 1990–1999 and 2000–2011 are the most used programming languages today in industries, while the languages developed in 1950–1959 and 1960–1969 are rarely in use today.

However, the languages developed earlier might not

TABLE II.

NUMBER OF JOB POSTINGS FOR THE LANGUAGES IN 1950–1959 DURATION AS OF 4 JANUARY, 2022

Language	Number of Job Postings
FLOW-MATIC	9
FORTRAN	30
LISP	11
COBOL	24,768
ALGOL	6
RPG	94

TABLE III.

NUMBER OF JOB POSTINGS FOR THE LANGUAGES IN 1960–1969 DURATION AS OF 4 JANUARY, 2022

Language	Number of Job Postings
SIMULA	0
SNOBOL	0
BASIC	0
APL	0
PL/I	5

TABLE IV.

NUMBER OF JOB POSTINGS FOR THE LANGUAGES IN 1970–1979 DURATION AS OF 4 JANUARY, 2022

Language	Number of Job Postings
PASCAL	6
C	37,895
SMALLTALK	3
SQL	32,307
ML(OCaml)	1

TABLE V.

NUMBER OF JOB POSTINGS FOR THE LANGUAGES IN 1980–1989 DURATION AS OF 4 JANUARY, 2022

Language	Number of Job Postings
C++	11,744
ADA	0
MATLAB	13,775
OBJECTIVE-C	25,735
PERL	94,321

TABLE VI.

NUMBER OF JOB POSTINGS FOR THE LANGUAGES IN 1990–1999 DURATION AS OF 4 JANUARY, 2022

Language	Number of Job Postings
HASKELL	61
PYTHON	95,237
VISUAL BASIC	4,763
RUBY	1,629
JAVA	90,260
JAVASCRIPT	96,435
PHP	15,322

TABLE VII.

NUMBER OF JOB POSTINGS FOR THE LANGUAGES IN 2000–2011 DURATION AS OF 4 JANUARY, 2022

Language	Number of Job Postings
ACTIONSCRIPT	38
C#	17,554
F#	1,79,270
GROOVY	110
SCALA	9,291
GO	1,07,000
DART	139
COFFEESCRIPT	9
SWIFT	5,364
KOTLIN	818

be used explicitly in industry today, but are an integral part of the newer languages that are developed. FLOW-MATIC first appeared in 1955 but was later extended into COBOL, which still has around 150–200 billion lines of existing code [13]. ML was a programming language that first appeared in 1973 and though it is not in wide use today, many of the newer programming languages such as OCaml and F# are considered to be dialects of ML. ML Meta Language is the root of a tradition of "strongly typed functional programming languages" that include Edinburgh ML, Miranda, Haskell, Standard ML, OCaml, Elm, ReasonML, and PureScript. F# is part of this family [14]. Similarly, Objective-C, which is primarily used for writing iOS and MacOS, was made on top of C language with added features of SmallTalk, which first appeared in 1972 and is now rarely used in industry.

In contrast, some of the languages that have been developed quite recently, namely CoffeeScript and ActionScript, are seen to be used very rarely. CoffeeScript was a language that aimed to increase the readability of JavaScript code. Drawing inspiration from Python and Ruby, it simply made the syntax compact. However, shorter syntax with no or poor documentation led to dissatisfaction in larger projects and developers found it more time consuming to write code in CoffeeScript and once compiled, debug it in JavaScript. As of today, CoffeeScript is considered a dead language. ActionScript, on the other hand, was developed at Adobe which is a large corporation. Actionscript's interactive projects and graphical animations provide a rich learning environment. The program's results are immediately visible which makes debugging easier and the final results more tangible and rewarding [15], but it failed to gain traction as a viable language. Adobe has announced its plans to end support for the language. However, it is still is used for prototyping, but will soon be about to extinct as other languages take its place.

VIII. THE TIOBE INDEX

The TIOBE programming community index [16] is a measure of the popularity of programming languages created and maintained by TIOBE Software BV based in Eindhoven, Netherlands.

The index is calculated from the number of search engine result queries containing the name of the language. It covers searches in the following search engines:

1) Google

2) Google Blogs

3) MSN

4) Yahoo!

5) Baidu

6) Wikipedia

7) YouTube

By inspecting the popularity of programming languages using a vast data set and the process such as the one utilized by the TIOBE index, we can find the relevance of different programming languages and use it as a parameter to estimate which languages are no longer widely used.

Table VIII shows the TIOBE index for January 2022 of the programming languages being discussed in the present work.

TABLE VIII.

TIOBE INDEX FOR JANUARY 2022 FOR LANGUAGES REVIEWED IN THIS PAPER

TIOBE Index January 2022 Ranking	Programming Language	Rating
1	PYTHON	13.58%
2	C	12.44%
3	JAVA	10.66%
4	C++	8.29%
5	C#	5.68%
7	JAVASCRIPT	2.09%
9	SQL	1.80%
10	SWIFT	1.41%
11	PHP	1.40%
13	GO	1.04%
15	CLASSIC VISUAL BASIC	0.98%
16	MATLAB	0.96%
17	GROOVY	0.94%
19	FORTRAN	0.7%
20	PERL	0.71%
24	OBJECTIVE-C	0.58%
25	COBOL	0.51%
29	KOTLIN	0.37%
34	ADA	0.34%
35	LISP	0.33%
37	DART	0.27%
45	RPG	0.15%
47	HASKELL	0.15%
48	ML	0.15%

IX. RESEARCH SURVEY

A. Method

After reviewing programming languages on websites like GitHub, StackOverflow, TIOBE Index, and Naukri.com, both at the national and international levels on the internet, we conducted a survey on the popularity of programming languages on a local level as well.

The survey had a sample size of 200, and the age range of respondents was from 17 – 52 years.

The survey was conducted using Google forms and contained the following responses for all the 38 languages discussed in this text:

1) Never heard of it;

2) Heard of it, but never worked with it;

3) Worked with it.

The information gathered reveals how many people are familiar with the languages discussed in this paper.

80.1% of the respondents had a Computer Science background while the other respondents belonged to, including but not limited to Civil Engineering, Chemical Engineering, Industrial Engineering, and Management.

The responses were those who are either currently active in the industry or are about to enter it in the coming years.

B. Analysis

Fig. 8 reflects the percentage of persons who work with a language and their familiarity with it.

Let us start with the first categorized band of languages, which spans the years 1950–1959 (Fig. 8). On having a look at the FLOW-MATIC chart, it can be noticed that there are no developers or students working with it. As a result, it is possible to say that the language is no longer in use. Fortran, on the other hand, has a fairly small number of users. The majority of people have never heard of the language, and many of those who have don't work with it. However, Fortran is still used in technical business.

LISP appears to be on par with Flow-Matic in terms of popularity. RPG is having similar outcomes because the vast majority of people are unaware of it. The RPG language evolved with computing in the late 1960s and early 1970s. It is worth noting that the evolution of the RPG programming language was inextricably linked to the advancement of IBM corporate computers. The RPG language evolved in tandem with the advancements in Operating Systems and computers [17]. Unlike Flow-Matic and LISP, many people are familiar with Cobol and Algol, but only a small percentage of those who have worked with it have done so recently.

Simula and Snobol, which were used in our next band from 1960–1969 (Fig. 9), can be considered dead languages because very few people know about them and there are no professions available in them, as shown in this paper. BASIC, however, elicited some reactions and

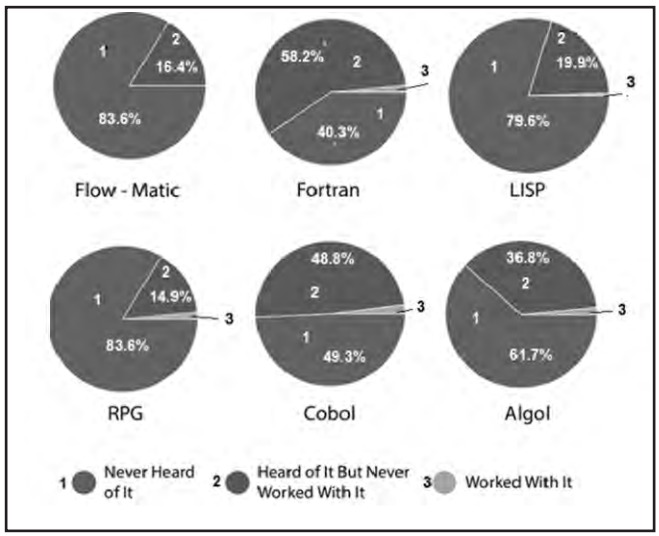

Fig. 8. Survey results for languages between 1950–1959

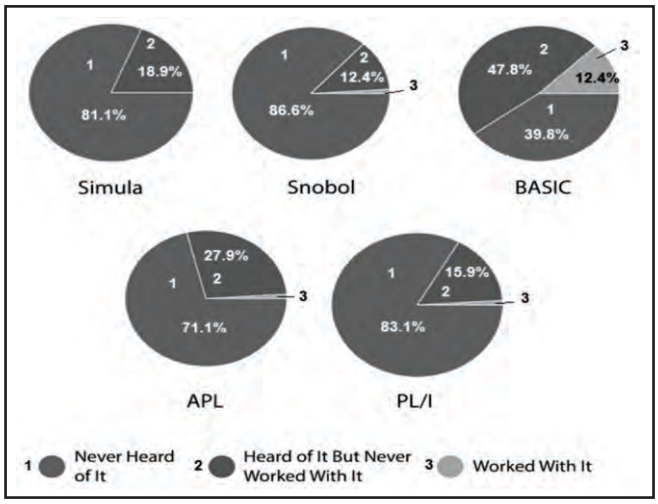

Fig. 9. Survey results for languages between 1960–1969

can be classified as a still-in-use language. APL and PL/I are two other languages that are on the decline in the computing industry.

C and SQL achieved new heights, which were amongst our third band of languages (1970–1979), and the majority of responses (Fig. 10) came from those who worked with them. PASCAL's usage has been narrowed, but it is not yet dead. Smalltalk perished as a result of haste, wrong development, and hype, as seen by the graph, which shows that the language is no longer in use.

C++, being a widely used language (Fig. 11), yielded predicted results because it is a language that everyone is familiar with and with which the vast majority of people work or have previously worked. MATLAB and Objective - C weren't as popular but are still used for a few purposes. As it may be observed, a sizable fraction of the

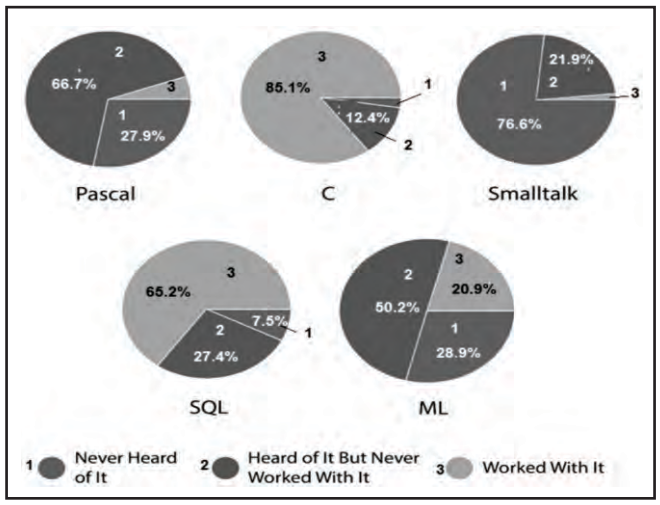

Fig. 10. Survey results for languages between 1970–1979

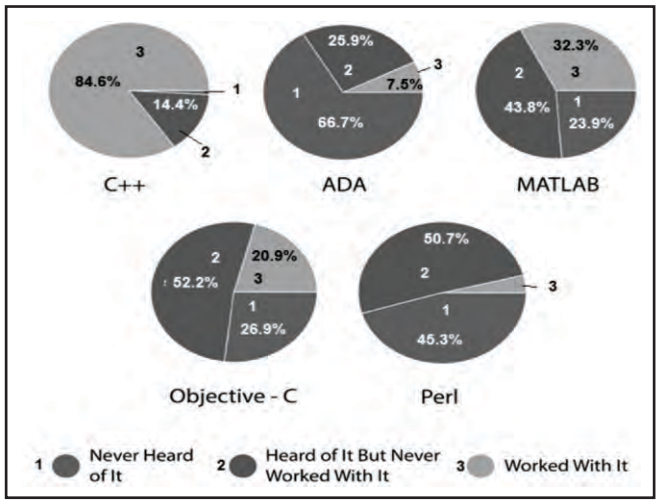

Fig. 11. Survey results for languages between 1980–1989

population is conversant in these languages. When it comes to ADA, few people are familiar with the language, but some people are still working with it. Perl came up with identical results, with the majority of people not knowing what it is.

Python is the most popular programming language among many, according to technical indices. Python's popularity has grown as a backend web development language, thanks to the introduction of jQuery and more recently, Node.js, especially given it has a fragmented MVC ecosystem [18]. However, as big data becomes more popular, Python has become a more in-demand expertise than it has ever been, especially because it can be integrated into practically any online app.

Other than Python, Java and JavaScript are popular on this survey also, as they are on other websites and articles. The majority of responses (Fig.12) came from people who said they work with these languages or have previously worked with them. When it comes to Visual BASIC, over half of the people don't know what it is, but nearly a third of those who do are still using it. Because it is still unfamiliar to many, PHP produced some surprising results. The least used language in this band is Ruby.

Here, too, ActionScript, F#, and Scala followed the same unfavourable tendency. Moreover, half of the population is unaware of these languages. C# is steadily gaining traction, and as a result, it has received some positive feedback from those who deal with it. Groovy, on the other hand, is still mostly unknown. The Groovy language can be used as a Java platform scripting language.

Go was designed from the ground up for networking and infrastructure related applications. It is an attempt to mix the ease of an interpreted language with dynamic types combined with the efficiency of a compiled language with static types [19]. In this study, it can be shown that most people are familiar with Go but have never worked with it previously, with the exception of a handful. Dart was created for client development such as web and mobile apps, and is widely used by programmers. Dart has previously been used by a number of individuals that can be seen in this survey.

CoffeeScript, on the other hand, appears to be extinct (Fig. 13). CoffeeScript, which was launched in 2009, is a small language that compiles to JavaScript. In the market, the language is regarded dead, but its repository is still active. Swift and Kotlin are gaining traction in the tech world. However, they did not get the desired results in this

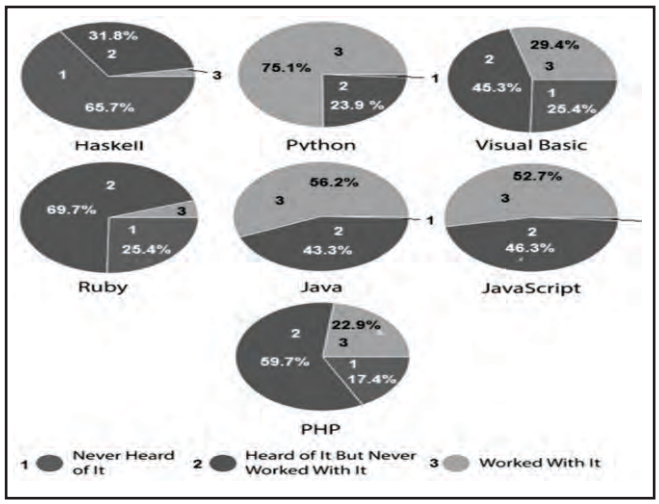

Fig. 12. Survey results for languages between 1990–1999

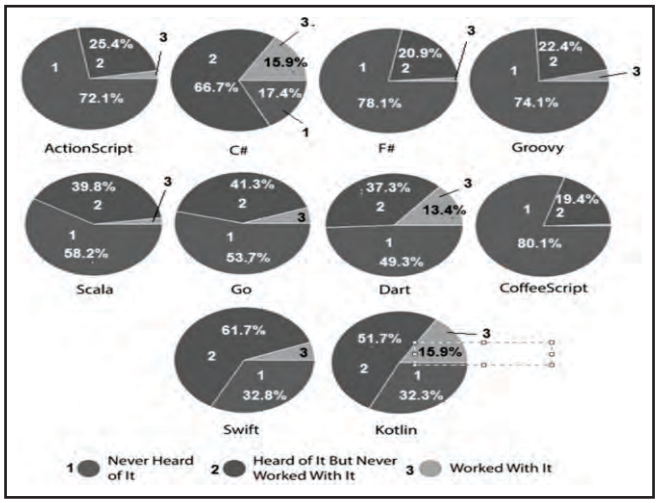

Fig. 13. Survey results for languages between 2000–2011

TABLE IX.

RESULT

Category	Languages
Dead or Extinct	FLOW-MATIC, Simula, Snobol, APL, PL/I, ML, Smalltalk, Basic, ADA, CoffeeScript
On The Verge of Death or Extinction	LISP, ALGOL, RPG, Visual Basic, Pascal, ActionScript
Relevant	FORTRAN, COBOL, C, SQL, C++, MATLAB, Objective - C, Perl, Haskell, Python, Ruby, Java, JavaScript, PHP, C#, F#, Groovy, Scala, Go, Dart, Swift, Kotlin

case. Swift is a programming language for iOS, iPadOS, macOS, tvOS, and watchOS and Kotlin is both for Android and iOS. People are becoming aware of them steadily. Swift incorporates the best of C and Objective-C, without the constraints of C compatibility [20].

X. RESULTS

After extensively reviewing all the languages discussed in this paper, we can categorize the languages in three groups:

1) Dead or extinct

2) On the verge of death or extinction

3) Relevant

1) Dead or extinct

These languages may serve as a basis for newer languages, or have been extended into other languages, but are no longer explicitly or directly utilized for further development.

2) On the verge of death or extinction

These languages are still being used for further development, but due to the shrinking developer base, decommissioning by large corporations or projects and lesser popularity, these languages will soon be dead or go extinct.

3) Relevant

These languages are currently extensively used for development and in academia, and are widely known.

We have tabulated the languages in these three categories as show in Table IX.

XI. CONCLUSION

Various programming languages have been developed since the inception of modern computers. In this paper we tried to identify the programming languages that are dead or are on the verge of death. This paper has challenged our preconception that older languages, that is, the languages developed earliest are most likely to be dead as the years go by. In this text, we were able to identify that languages, regardless of the time of their first appearance can be on the verge of extinction. Out of the 38 languages that were reviewed in this text, 10 were categorized as dead or extinct, 6 were on the verge of extinction, and 22 languages are still relevant.

AUTHORS' CONTRIBUTION

Apoorv Mohit conceptualized the methodology of the research and documented the findings. Sunaina Rustagi collected information on programming languages and organized it using explanations and infographics. Collectively, the two authors carried out the survey on programming languages usage and jointly finalized this original work with expertise.

CONFLICT OF INTEREST

The authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in the manuscript.

FUNDING ACKNOWLEDGEMENT

The authors have not received any financial support for the research, authorship, and/or for the publication of the paper.

REFERENCES

[1] N. Hamilton, “The A-Z of Programming Languages: C++.” Computerworld. com. [Online]. Available: https://www2.computerworld.com.au/article/250514/a-z_programming_languages_c_/?pp=5

[2] N. Hamilton, “The A-Z of Programming Languages: Python.” Computerworld. .com. [Online]. Available:https://www2.computerworld.com.au/article/255835/a-z_programming_languages_python/

[3] “Fortran The Pioneering Programming Language.” IBM.com. [Online]. Available: https://www.ibm.com/ibm/history/ibm100/us/en/icons/fortran/

[4]“Developer Survey 2021.” StackOverflow.com. [Online]. Available: https://insights.stackoverflow.com/survey/2021#overview

[5] C. Shipley and S. Jodis, *“Programming Languages Classification,”* in *Encyclopedia Inf. Syst.*, pp. 545–552, 2003, doi: 10.1016/B0-12-227240-4/00138-6.

[6] “A small place to discover languages in Githut.” Githut.com [Online]. Available: https://madnight.github.io/githut/#/pull_requests/2021/3

[7] A. Jindal, P. Khurana, and T. Goel, “Comparative study of C, Objective C, C++ programming language,” *Int. J. Eng. Comp. Sci.,* vol. 2, no. 1, pp. 202–206, 2013. [Online]. Available: http://www.ijecs.in/index.php/ijecs/article/view/103

[8] P. Hudak, J. Peterson, and J. H. Fasel, "A gentle introduction to Haskell 98," 1999. [Online]. Available: http://ais.informatik.uni-freiburg.de/teaching/ws06/info1/material/haskell-tutorial.pdf

[9] N. Prokofyeva and V. Boltunova, "Analysis and practical application of PHP frameworks in development of web information systems," *Procedia Comput. Sci.*, vol.104, pp. 51–56, 2017.

[10] M. Odersky, P. Altherr, V. Cremet, B. Emir, S. McDirmid, S. Micheloud, N. L. Mihaylov, M. Schinz, E. Stenman, and M. Zenger, *An Overview of the Scala Programming Language,* 2nd Ed., 2006.

[11] B. Ray, D. Posnett, V. Filov, and P. Devanbu, “A large scale study of programming languages and code quality in Github.” web.cs.ucdavis.edu. [Online]. Available: https://web.cs.ucdavis.edu/~filkov/papers/lang_github.pdf

[12] Naukri.com. [Online]. Available: https://www.naukri.com/

[13] R. J. Kizior, D. Carr, and P. Halpern, "Does COBOL have a future?," in *Proc. Inf. Syst. Educ. Conf.*, vol. 17, no. 126, 2000. [Online]. Available: http://proc.edsig.org/2000/126/ISECON.2000.Kizior.pdf

[14] D. Syme, "The early history of F#," in *Proc. ACM Program. Lang,* vol. 4, HOPL, Article 75, Jun. 2020, doi: 10.1145/3386325.

[15] S. Crawford, and E. Boese, "ActionScript: A gentle introduction to programming," *J. Comput. Sci. Colleges,* vol. 21, no. 3, pp. 156–168, 2006, doi: 10.5555/1089182.1089203.

[16] “TIOBE Index for May 2023.” TIOBE.com [Online]. Available: https://www.tiobe.com/tiobe-index/

[17] A. Sharma, F. Khan, D. Sharma, and S. Gupta, “Python: The Programming Language of Future,” *Int. J. Innovative Res. Technol.,* vol. 6, no. 2, pp.115–118, May 2020. [Online]. Available: https://www.academia.edu/84080385/Python_The_Programming_Language_of_Future

[18] A. Järleberg and K. Nilsson, "Go, F# and Erlang." csc.kth.se. [Online]. Available: https://www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2012/rapport/jarleberg_anders_OCH_nilsson_kim_K12040.pdf

[19] T. Ngo-Ye, J. J. Choi, and D. Gittens, "Preparing teaching Swift Programming and actual classroom teaching," *Issues Inf. Sys.*, vol. 21, no. 4, pp. 295–305, 2020.

[20] D. C. Pence, “RPG (Report Program Generator Language)” in *Encyclopedia Inf. Sys.*, 2003, pp.723–734, doi: 10.1016/B0-12-227240-4/00151-9.

About the Authors

Apoorv Mohit is an accomplished Information Technology Engineering student with specialization in Machine Learning (ML) and Data Science. He possesses a strong academic record, experience in programming, and data analysis, and has worked on several ML projects.

Sunaina Rustagi has a focus on Machine Learning and Data Science and is an excellent IT Engineering student. She served as an ML intern at NHPC in India and has a good academic background. She is an excellent programmer and a brilliant designer.