**Tyler Carrier**

**Web Applications**

**Evolution and Prediction of Computer Languages**

**9/15/18**

**Box #88**

Since the beginning of time, humans have been unable to effectively communicate with technology. That was until 1957 when the first programming language, FORTRAN, was developed at IBM by a very small project team led by John Backus. It was designed to communicate with the programming tool known as IBM 704 (Harper and Stockman). Computer languages give us the opportunity to give instructions to a computer in a way which the computer will understand. FORTRAN was the first to accomplish this task. The feat was one that would spark a technological revolution.

Computer languages are still growing since the development of FORTRAN in 1957. One language the came soon after FORTRAN is known as COBOL or Common Business-Oriented Language. This language was developed in 1960 by teams at the Pentagon and other outside resources. This language heavily stemmed from the FLOW-MATIC language developed just a few years prior (Computer History). This language was designed for businesses to use. It was designed so that it could efficiently run on any computer, making it one of the first languages to be easily accessible to public corporations.

When 1964 came around, two men by the names of Thomas Kurtz and John Kemeny created a language known as BASIC (Beginner’s All-purpose Symbolic Instruction Code). This was a simple language designed for Dartmouth students who were unexperienced in coding. Although it started in just one school, BASIC became widely recognized and practiced at schools across the world. In time, BASIC was installed on mostly every computer being shipped to the common user (Computer History). This would encourage people of all types to learn programming.

Unix came in the year 1969, two other men, Kenneth Thompson and Dennis Ritchie developed a system known as Unix. They based their creation largely off the previously created Multics, which was developed in the mid-1960s. This is according to an article by Computer History, “UNIX combined many of the timesharing and file management features offered by Multics, from which it took its name.” Unix attracted many people which attributes to its long term success. Unix is still widely used today within the world of computer science.

Just three years later, one of the most popular computing languages of the 20th century came to light known as the C programming language. This language was created by a team led by Dennis Ritchie (co-founder of Unix). He and his team would go on to recreate the UNIX system in their new C based programming language (Computer History). This popular language would later lead to the development of other C based languages known as C++, C#, Objective-C. These languages are widely used today.

In 1983, one of MIT’s Artificial Intelligence Lab programmers, Richard Stallman, developed an operating system known as GNU or “GNU’s Not Unix.” Stallman wanted to provide people with an alternative to Unix. GNU had a slow start, but aided in the later development of Linux, a truly efficient alternative to Unix. Richard Stallman would later found a company named the Free Software Foundation based on his desire to create an alternative which would be free to users (Computer History).

When 1995 came around, two major advances in the world of computer science came to light. The first is known as Java. This is an object-oriented programming language developed by Sun Microsystems. Java instituted the “write once, run anywhere” idea which allowed programmers to create programs which could run on any system (Computer History). The second is JavaScript. This would aid largely in the development of HTML or Hypertext Markup Language. It was developed by Brendan Eich from Netscape Communications.

The development of the languages described were each contingent upon one or the other in separate ways. This is what makes the continual development of programming languages so important- it allows the next language to be that much better. Languages that have developed over time have translated into the creation of newer languages such as Elm, Rust, Kotlin, Elixir and Crystal (Pronschinske). These languages seek new ways to accomplish different tasks at the most efficient way possible.

The future of computer programming languages is very bright. I believe at some point in the future, languages will be developed to the point where even children could understand how to operate them. Languages will have a user friendly environment in which the level of comprehension will be minimal. In essence, I believe we will be able to communicate with computers through languages of the common tongue. For example instead of using if statements we will simply be able to say, “computer, do this, and if that doesn’t work than just do this.” This will allow the user to communicate with the computer in the most raw way possible. The history of computer languages prove its gradual progression throughout time, but it is just the beginning.

**Work Cited:**

Pronschinske, Mitch. “5 Emerging Programming Languages with a Bright Future.”

TechBeacon, TechBeacon, 22 Aug. 2018,

techbeacon.com/5-emerging-programming-languages-bright-future.

Software & Languages | Timeline of Computer History | Computer History Museum.”

Edited by Computer History, What Was The First PC?,

[www.computerhistory.org/timeline/software-languages/](http://www.computerhistory.org/timeline/software-languages/).

“Guide to Programming Languages.” ComputerScience.org,

[www.computerscience.org/resources/computer-programming-languages/](http://www.computerscience.org/resources/computer-programming-languages/).

Harper, David, and L.M. Stockman. The History of FORTRAN,

www.obliquity.com/computer/fortran/history.html.