John Backus (1924 - 2007)
The Father of High-Level Programming

Widely considered the first true high-level programming language, Fortran was revolutionary. It allowed computer programmers to only enter short instructions for operations instead of the much longer underlying machine code that a computer is able to read. This was achieved through the success of the Fortran compiler that would translate the new, more comprehensible instructions to industry efficient machine code, which previous attempts had so far failed to accomplish. Suddenly professionals in fields outside of computing could take advantage of the emerging power of computers even with little knowledge of how to program them. In 1956, an IBM team led by John Backus introduced Fortran to the world and laid the groundwork for what we all see as modern computer programming.

With a restless and intelligent mind, John Backus was certain to achieve great things; but had to endure relative obscurity before stumbling on his path during an innocuous trip past IBM’s headquarters in New York. Born into a wealthy family in Philadelphia, Pennsylvania, as the son of a stockbroker, Backus was luckier than most with the resources available to him to realise his potential. Despite his expensive education however, his tenure at high school was unremarkable, noting that “the delight of the place was all the rules you could break”. Hindered by an appetite for practical work and distaste for rote learning, Backus’s bad habits extended into university; culminating in an expulsion from the University of Virginia after less than a year due to poor attendance. Shortly after, however, with his country in the grips of a world war, he was whisked away on military service where glimpses of his brilliance emerged.

After scoring excellent marks on his military aptitude test, Backus was sent to a number of universities where he dipped his toes into various courses from medicine to engineering. Seeking stability after being honourably discharged from the army, Backus found his feet at a radio technician school where he dabbled in electronics. There he found a love for mathematics and undertook a postgraduate degree at Columbia University. In 1949, with graduation looming, Backus decided on a trip to IBM's Headquarters on Madison Avenue, where the company's behemoth SSEC (or Selective Sequence Electronic Calculator) was on show. After mentioning his pending maths degree and interest in the machine to a nearby tour guide he was brought swiftly to the manager and was subjected to an on-the-spot test of mathematical brain teasers. This was Backus in his element and he was instantly hired as a programmer to work on the beast, where his efforts instructed the machine to execute complex calculations that were utilised in the proceeding Apollo missions.

Backus then began working on IBM's 700 series Electronic Data Processing Machines or their first commercial computers. The code at the time was at machine-level and therefore very slow to programme, with Backus likening his task to "hand-to-hand combat". This frustration motivated him to find a way to make programming easier and, following a request to his superior, he was granted permission to form a team to tackle this issue. Backus proceeded to assemble a group, 10 strong, populated by adept problem solvers, from chess whizzes to cryptographers, that could work together to accomplish his goal. After two years the motley crew issued the first release of Fortran, which, due to its impressively efficient compiler, soon became the darling of technical types around the world.

Fortran was built on the principle of speedcoding that describes common mathematical operations into a condensed instruction to be compiled into the lethargic machine code later. The language looked like a mix of algebra and colloquial English making it quite easy to learn and use which led to its popularity. Another significant factor was the prerequisite of only the installed compiler rather than a dedicated machine to run it, allowing any system to have access to it. While Fortran, true to its name (FORmula TRANslating System), was more accommodating to mathematicians and scientists, it inspired others to create similar programming languages that could be more accessible to the public and therefore used in other industries.

One such product was COBOL or COmmon Business Oriented Language, that evolved from a language called FLOW-MATIC for UNIVAC created by Grace Mary Hopper. Hopper had previously tried to introduce something similar to Fortran before Backus (comprehensible language to a wider audience) but her compiler proved unsuccessful, the machine-code translations often running much slower to what a skilled programmer could achieve. Hopper did, however, notice the discontent a lot of people had for the mathematical symbols used in Fortran, after its inception, and understood that what they wanted was words. This provided the idea for her verbose FLOW-MATIC and later COBOL where you could enter commands such as: MULTIPLY HOURS BY WAGE-RATE.

A couple of years after Fortran's first release, Fortran II was introduced. The new edition contained a handy tool that continues to endear itself to programmers today: the subroutine. The monotony of repeating nearly identical code was quelled and the hassle of developing one's own functions abated. Programmers only needed to know what to input and what would be outputted to use them and they could be slotted quite seamlessly into their programs. Backus's later work included a notation (The Backus-Naur Form BHF) of a context-free grammar that described the syntax of a language he helped designed called ALGOL 58 and later ALGOL 60. This was an internationally backed project sought as progress on Fortran and the new languages emerging. Later efforts saw his foray into functional programming at IBM, seeking to develop a way for programmers to code their problem in a more abstract manner.

Backus strived to find better ways of programming throughout his career and Fortran is his lasting legacy, a pivotal moment in the development of computer software. The system, remarkably, is still used for certain scientific calculations today, such as in weather forecasting, but his mark in history is the paradigm shift he created in the way people communicated with computers. He was a trailblazer during programming's infancy, as Ken Thompson (developer of the Unix OS) notes, "95% of the people who programmed in the early years would never have done it without Fortran.".

Patrick Lee
15325692

References:
[http://www.nytimes.com/2007/03/19/obituaries/20cnd-backus.html]
[https://www.theguardian.com/science/2007/apr/05/obituaries.news]
[http://amturing.acm.org/award_winners/backus_0703524.cfm]
[*Making The World Work Better: The Ideas That Shaped A Century And A Company* - Kevin Maney - Steve Hamm - Jeffrey M. O'Brien - *Printed 2011*]