Donald Knuth

Born:	January 1938
Nationality:	American
Institution:	Stanford University
Fields:	Mathematics, Computer Science
Achievements:	Grace Murray Hopper Award (1971) Turing Award (1974) Turing Lecture (2011)

Early Days

Donald Knuth attended a Lutheran School and it was here that his love for education, music and mathematics developed. During his early teens, it was thought that Donald's strengths would lie in literature rather than programming. It was not until after school when he was first introduced to computers, that Donald's passion changed from sentence structure to code structure. Knuth is a graduate of Case Institute of Technology where he studied mathematics as an undergraduate. His post graduate studies brought him to California Institute of Technology where he received a Ph.D. in Mathematics in 1963. Throughout this time Knuth was involved in software development, acting as a consultant to Burroughs Corporation and as an editor of Programming Languages for AMC publications. His interests in software development began while helping in programming centres as an undergraduate. Knuth's programming knowledge developed alongside Mathematics until he was appointed to Stanford in 1968 as the first endowed chair in Computer Science.

Key Work

The Art of Computer Programming

Knuth began to write his first academic book on 'compiler design' in 1962 before realising that the scope was just too great to be refined to the compiler design topic. The book title was altered to 'The Art of Computer Programming' in which Knuth tackles the issues involved in the analyst of algorithms. Donald Knuth's work on the analysis of algorithms is the work he has received most recognition for. Donald Knuth specialised in this area due to the 'need for a book, rather than the demand for a book.' Knuth went about much of his research with this quote in mind. He looked to change how software was developed whilst simultaneously improving and enhancing techniques used, rather than solving human tasks. Knuth once said "let us change our traditional attitude to the construction of programs. Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather than explaining to human beings what we want a computer to do." It is this philosophy which made his work so progressive at the time and his software engineering techniques so interesting.

Analysis of Algorithms

Knuth is regarded as the father of analysing algorithms, it is Knuth who took the idea of using mathematical methods to understand programs and made a life time of work from it. It is not the study of solving problems mathematically, rather, monitoring and formally assessing algorithms to determine how fast and efficiently they return the solution that

Knuth took an interest in. Knuth has papers on analysing ‘Length of strings in Merge sort’ to ‘Linear Probing and Graphs.’ His work shows the evolution of current terminology and fundamental ideas currently accepted such as ‘Big Omicron, Big Omega and Big Theta.’

TeX

During the publication of the second volume of the ‘The Art of Computer Programming’ in 1968, Knuth came across the problem of typesetting. The gallery proofs of this new volume, printed using phototypesetting did not meet his high standards. Knuth took it upon himself to invent the concept of literate programming, a way of producing compilable source code and cross linked documentation typeset in TeX from the same original file. While working on his book that was published in 1976, Knuth and his student Peter B Bendix attempted to solve the world problem in algebraic systems. Their solution totally changed the way mathematics and scientific articles are written today.

It is not just the intelligence and significant skill in his field that impressed me about Knuth but also the way he went about his work. As aforementioned, Knuth’s work was driven, not by the desire to create valuable code structure, rather, he looked to develop areas of computer science in which he thought progress was needed. TeX is still widely used today and as *E F Robertson says* “I [EFR] am certain that the added ease of production and communication of mathematics using TeX has had a major impact on the subject over the last ten years.”

Personal Analysis

While Knuth was responsible for significant progress in computer science, it was his method of working rather than the product of his work, that caught my attention. It was his meticulous attention to detail and work ethic that made him such a well know and highly regarded computer scientist. Knuth was renowned for offering a cheque of $2.56 to anyone who found an error in his published books whether it be technical, typographical or historical. The payment of $2.56 was chosen as it corresponded to the hexagonal dollar. Knuth’s attention to detail and his desire to not just make a name for himself or make money, but to improve the field of computer science made him stand out to me.

Impact of the individual

Donald Knuth was the first to look beyond advancing what computer programming could do in the world, and rather looked at how the programming itself could be improved. He took programming as an art and produced algorithms and measures to monitor and compare programs. Knuth’s work is thorough, meaningful and is as progressive now as it was when fist published. Knuth spent days working on the wording of just a single sentence in his book The Art of Computer Programming and as a result it is still the book computer scientists have at the forefront of their bookshelves. Knuth is well known as a perfectionist and his book, the labour of his love for the subject, is a prime example of this.

When asked where his career may have lead him if it wasn’t for computer science, Knuth stated that if it hadn’t been for computer science it would have been “algorithms even if fate had turned out differently than it did, and if there wasn’t a penny in programming.” Many computer scientists are thankful that it was their field he took interest in.