**Donald Knuth Biography**

**Background/Early Days**

Donald Knuth was born January 10, 1938, in Milwaukee, Wisconsin to German-Americans Ervin Henry Knuth and Louise Marie Bohning. His father was a teacher in a Lutheran high school and a church organist. Donald attended Case Institute of Technology where he first majored in Physics however, he switched his major to Mathematics in his sophomore year.

Donald’s love for programming started when he came across a IMB 650 computer in the statistics department of his college. He wrote many interesting programs during his undergraduate years, including one to rate the performance of players of the basketball team he managed. Donald performed so well at mathematical studies at college that the faculty awarded him a Masters in mathematics when he finished his Bachelors work. He then went on to receive a PHD in mathematics from the California Institute of Technology.

**Most Influential Work**

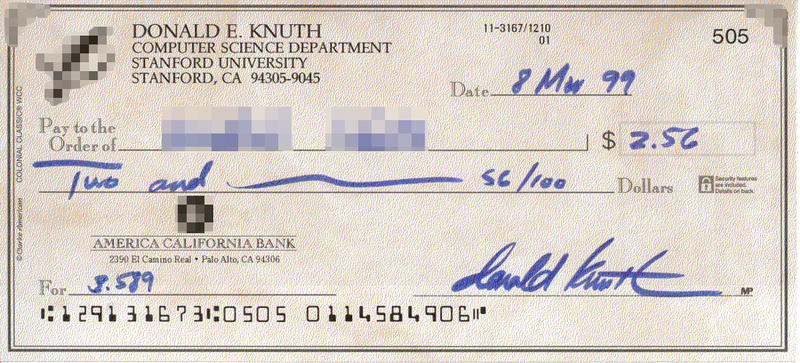
After receiving his PHD from CalTech he went on to become an associate professor at the college. Donald then started to do private consulting, and wrote compilers for various computers. The word got around that he knew a lot about compilers, and in his second year at CalTech, Addison-Wesley asked him to write a book on compilers. He originally planned to publish this as a single book. As Donald developed his outline for the book, he decided that he required seven volumes to thoroughly cover the subject. He published the first volume of “The Art of Computer Programming” in 1968. That same year Donald moved from CalTech to Stanford University, a move motivated by the fact Stanford also agreed to hire his colleague Robert Floyd, the 1978 Turing Award winner.



“The Art of Computer Programming” – Volumes

1. Basic Concepts and Information Structures
2. Seminumerical Algorithms
3. Sorting and Searching
4. Combinatorial Algorithms
5. Lexical Scanning and Parsing
6. Context Free Languages
7. Compiler Techniques

Donald is also widely known for the humorous side of his personality. He used to pay a finder's fee of $2.56 for any errors or mistakes discovered in his books. $2.56 because "256 pennies is one hexadecimal dollar". He also used to give $0.32 for valuable suggestions made for his book. According to an article in the Massachusetts Institute of Technology's *Technology Review*, these Knuth reward checks are "among computerdom's most prized trophies". Most people who receive a check from Knuth do not actually cash them in, instead they frame them on their wall. Knuth had to stop sending real checks in 2008 due to bank fraud. He instead now gives each error finder a "certificate of deposit" from a publicly listed balance in his fictitious "Bank of San Serriffe".



(One of Knuth’s reward checks)

**Awards**

In 1974 Donald Knuth was awarded the prestigious Turing award. The Turing award is an annual prize for an individual whose contributions "should be of lasting and major technical importance to the computer field".

“The 1974 A.M. Turing Award was presented to Professor Donald E. Knuth of Stanford University for a number of major contributions to analysis of algorithms and the design of programming languages, and in particular for his most significant contributions to the "art of computer programming" through his series of well-known books. The collections of technique, algorithms, and relevant theory in these books have served as a focal point for developing curricula and as an organizing influence on computer science.”

**Extra Work**

It is important to point out that Donald Knuth had some influential unexpected side work in areas that lay outside his primary specialization. While working on “The Art of Computer Programming” in 1971 he published his groundbreaking paper "An empirical study of FORTRAN programs**.**" In this paper, he laid a foundation of empirical analysis of computer languages by providing empirical evidence about the critical influence of the level of optimization of "inner loops" on performance. Essentially it was Knuth who justified the legitimacy and importance for high level languages of C language shortcuts i++, a+=1. The constructs that for a long time caused dislike of most orthodox language designers and academic language researchers raised on strict Algol tradition.

This paper might have inspired the introduction to C increment statements. The language C came into existence in 1969-1973, in parallel with the early development of the Unix operating system. The most creative period occurred during 1972, one year after Knuth paper was published.