

(Approx. 677 words)

Dan's Desk

Computer Fundamentals #4

By Dan Douglas, President, Space Coast PCUG, FL

June 2018 issue, The Space Coast PC Journal

www.scpcug.org

datadan (at) msn.com

Here is part 4 of my series on "Computer Fundamentals" - a series of articles to explain the fundamentals of computers, programming and their usage that will hopefully improve your understanding of how it all works and why things are the way they are.

This time we will examine just a few of the more common well-known programming languages out of the hundreds developed. Do you realize that a player piano is one of the early forms of programming? Programming started by being a way for a human to instruct a machine to do something.

In this case, what started out as a non-programmable device, the piano which had been around for centuries, became a programmed computer when modified to strike keys when a sensing mechanism detected holes in a roll of paper. The position of the hole told the machine which key, or keys, out of the possible 88, to strike to produce a note.

This kind of programming was tied to a certain type of automation machine and could only work on that specific machine that it was designed for. You couldn't take a roll from a player piano for example and use it to automate a drum playing the same tune. Wikipedia lists this as the first known example of programming - "the automatic flute player described in the 9th century by the brothers Musa in Baghdad."

Very early actual computers used no programming at all, but rather connected bits of circuitry to calculate and produce a result. This evolved into the 1st generation of programming languages, often referred to as machine language coding.

These programs were written in either decimal or binary format and read into the computer through toggle switches, punched cards or on magnetic tape. They were simply the expression of the old connected bits into logical operations doing the same task. An example would be to take the contents of storage area 1 and add it to the contents of storage area 2.

The 2nd generation of programming languages are called Assembly languages. They were tied to a specific architecture and instruction set of a brand or model of computer but did provide for human readable code that saved many hours of programming time and reduced the chance of errors.

Next came the first of the 'high-level' languages which formed the 3rd generation of programming languages. In 1954 FORTRAN was invented and is still used to today as

a high-performance language. In many cases it reduced the complexity of a program by a factor of 20. FORTRAN is derived from the words formula translator and is representative of the main focus of FORTRAN to aid with numeric computation and scientific task procedures.

The 3rd generation of languages used a compiler function which would take the relatively slow performing high level language and translate it into machine language for faster execution. During the late 50's there was a recognized need for non-scientific languages for use in the business community. Grace Hopper developed a program language called FlowMatic which addressed this concern and led directly to the development of Cobol (Common Business Orientated Language) in 1959 ,which became one of the most popular programming languages which is still in use today. The Department of Defense (DoD) standardized on COBOL in the 1960's and required it to be supported on every Computer purchased by the DoD. The goal was to have a portable language base that would allow for a program to be employed in multiple instances without having to rewrite it for each installation.

It is only during the late 1990's that programming languages such as C and C++ finally evolved to where languages such as Java became the primary language that truly allowed for 'write once, run anywhere' capabilities. This is why web pages can run on any computer using standard browsers independent of whether the hardware is from Apple, Dell, HP or any other company.

