Some history of Computers and Computing

Computers:

Although the electronic computer is of recent origin (some 50 years), the idea of automating the process of computation was born long back, probably, when book-keeping, accounting and astronomy became tedious. The earliest calculational aid used by man/woman was abacus. Traders and businessmen all over the world used it in the past and it is still being used in some parts of the middle-east and Japan. The word abacus came from the Greek word abakos (for a board or a tablet) which, in turn, was probably derived from the Hebrew abaq which means dust. The sand surface which was used earlier for writing had evolved into a board with lines. The modern abacus is a wooden frame fitted with rigid wires on which counters made of wood or plastic can slide. The credit for making the first calculating machine goes to Blaise Pascal (1642). He built a mechanical digital calculator to help his father with his business accounts. There was a toothed wheel for each digit and each wheel had 10 positions. By means of a dial the position of each wheel could be set to represent a digit. Addition was performed by a simple gear mechanism. The carry over that resulted from the addition of two digits was transferred to the next place by the intermittent motion of a mechanism located after the ninth position of each wheel. Multiplication could only be performed by repeated addition. In the light of the present day hardware, Pascal's invention was the next logical step after abacus. Abacus was a simple register. With a carry facility it became a counter. The modern(?) programming language PASCAL is named after Blaise Pascal, the great inventor.

The design of the first commercially available mechanical calculator is attributed to Leibnitz (1671) (the same bloke who invented calculus). It was completed in 1694 and Leibnitz demonstrated, for the first time, that binary arithmetic was superior to decimal arithmetic as it simplifies machine construction. Leibnitz's machine could add, subtract, multiply, divide and even find square root. Leibnitz's machine was a full adder (you'll learn about a full adder in your Digital Hardware Design course), in which the carry was saved in the form of the position of a lever. A working model of this machine was exhibited at the Royal Society of London in 1794, but it was somewhat unreliable. It was commercially available in 1820.

The first machine that could be programmed is attributed to Charles Babbage (1822). Charles Babbage was an Englishman, a man of science and a mathematician. In 1822, he thought of a fixed program calculator that could compute and tabulate polynomial functions by the method of differences. Unfortunately, the "Difference Engine", as he called it was never completed. But soon he designed and constructed a new machine called the "Analytic Engine" which was the first programmable computer. Babbage's machine could be programmed to follow a series of steps, where each step could be a combination of four basic operations (+,-,*,/). But more important was the fact that the machine had decision making capability. As a result it could change the order of calculation depending on the value of a certain quantity, which it had computed. For example, the machine could skip a number of steps or go back to a step it had already performed depending on the previously computed result. In absence of such a condition it would follow the steps in the order given. So, it had if-then-else and while-do. The machine had a logical structure that resembled the modern computer. It had a calculator section called the mill (CPU) and a storage unit which consisted of a group of counters. Each group had about fifty wheels. A thousand fifty digit numbers could be stored in the storage unit.

Lady Ada Augusta, the Countess of Lovelace, Lord Byron's daughter, was Charles Babbage's girl friend. She was completely fascinated by Babbage's machine and used to spend hours thinking up procedures to solve various mathematical problems (hacking?). She is recognized as being the first computer programmer. The US Department of Defense has named its modern programming language ADA after her. There is a poster on Babbage and Ada in the corridor of the computer center.

Konrad Zuse was a German inventor. Around 1936 he had read about Babbage's engine. During the height of Nazism he started working on a calculator which he called Z1. He used, for the first time, what has now come to be known as floating point numbers (a strong motivation for this was his own financial problems). With floating point arithmetic Zuse managed to get a wide range of numbers with a reasonably small representation. Another feature of Zuse's machine was that it was based on binary arithmetic. Zuse lacked the funds to buy as many decimal counter wheels as would be needed. Since on/off relays were much cheaper, he decided to use them instead. As for the mode of handling these binary numbers, he was aware of the work of George Bool, an English mathematician, who had already shown how arithmetic operations could be represented in terms of binary logic. In order to program his machine, he came up with a programming notation, which was the first programming language. This remained unknown to the rest of the world until the war came to an end. This language was named "Plankalkul".

Subsequently, Zuse designed a series of machines - Z2, Z3 and Z4. Z3 was made entirely out of discarded telephone relays. Unfortunately for Zuse, his first three machines got destroyed in a bomb raid over Berlin in 1944. Zuse had started his own computer manufacturing company. It is unknown whether he became rich.

The first electronic computer was built in the United States and was the result of a series of efforts.

The keyboard calculator was developed in the US around the middle of the 19th century. Though it was improved upon from time to time, none was fundamentally different from Leibnitz's machine.

While working on the US 1880 census, Herman Hollerith devised an electro-mechanical sensing system by placing a tray of mercury under a punched card and using pins to sense the holes.

In 1887, Leon Bolle succeeded in implementing multiplication directly instead of by repeated additions. Around this time printing devices were incorporated into adding machines and calculators.

The Harvard Mark I, constructed jointly by the Harvard University and IBM was the first electro-mechanical computer. The group was led by Howard Aiken. The Mark I was 15m long, 2.4m high (size of a dinosaur?) and weighed 2 tons. it could store 60 constants and had 72 storage counters for storing intermediate values. The control, unlike in

Babbage's analytic engine consisted mostly of electro-mechanical relays as did most of the computing hardware.

The first digital electronic computer was called ENIAC (Electronic Numeric Integrator and Automatic Calculator). It used vacuum tubes for switching. It was designed primarily by John Mauchly and Presper Eckert of the Moore school, Pennsylvania and was constructed with finances provided by the US army in 1946.

John von Neumann, a Hungarian, is probably one of the greatest mathematician, physicist and engineer ever born. He had made significant contributions to logic, mathematics, quantum mechanics, thermodynamics, fluid mechanics and originated `Game Theory' which he applied to economics. He was also a consultant to the atomic bomb project at Los Alamos. In this connection he got interested in the design of a powerful computing machine on the lines of ENIAC. Collaborating with the Radio Corporation of America (RCA) he supervised the building of a computer incorporating many of his own ideas. He developed the concept of "core memory" which is used even today. The core refers to an electro-magnetic core whose polarity was used to represent 0 or 1. Von Neumann, motivated by the fundamental mathematical results of Alan Turing and Alonzo Church (which forms the basis for the modern theory of computing) developed the concept of a "stored program". Essentially this means that the program or the sequence of instructions could also be stored in the same form as data. No longer did the central processor need two separate sets of instructions for reading the program and data. As a consequence, a computer could modify its own set of instructions, treating instructions too as data (quite different from Church's lambda calculus where data also is treated as function or program; though, surprisingly, they are equivalent). It now became possible to write programs to process other programs. Most modern computers use this concept and the corresponding architecture is known as the "von Neumann architecture".

The work of von Neumann really opened the flood gates, and in less than 50 years we have had microcomputers, minicomputers and main frames, IBM PC's and Mac's, Power PC's and Pentium's, HP's, VAX's, SUN's, SGI's, Cyber's, Cray XMP's and YMP's, the Indian supercomputers - PARAM and ANUPAM, and, of course, himalaya. Megaflops, Gigaflops etc. have become concepts known to even housewives.

Computing:

The notion of computing is, of course, much more fundamental than the notion of computers and the history of computing is older by several centuries than history of computers. The history of computing is as old as the history of numbers (after all numbers are for computing). We have already seen some of the very early algorithms like - Euclid's GCD, Sieve of Eratosthenes, testing primes, perfect nos., factorials etc. Then there are a whole set of procedures for ruler and compass constructions which you must have studied in school. Root finding etc., the complex calculations in astronomy are all algorithms which were all formed much before the advent of computers. In fact, computing has been a primary activity for all civilizations.

The idea of having an algorithm, or recipe for performing some task, has existed for thousands of years. For many years, people also held the belief that if any problem could be precisely stated, then with enough effort a solution could eventually be found (or else a proof that no solution exists could be provided). In other words, it was believed that there was no problem which was intrinsically so difficult that, in principle, it could never be solved. One of the great supporters of the above belief was one of the greatest mathematicians of all times, David Hilbert (Germany, 1862-1943) who once said:

"Every definite mathematical problem must necessarily be susceptible of an exact settlement either in the form of an actual answer to the question asked, or by the proof of the impossibility of its solution and therewith the necessary failure of all attempts...one of the things that attracts us most when we apply ourselves to a mathematical problem is precisely that within us we always hear the call: here is the problem, search for the solution; you can find it by pure thought, for in mathematics there is nothing that cannot be known".

Hilbert's aim was to develop a formal mathematical system in which all problems can be precisely formulated in terms of statements that are either true or false. His idea was to find an algorithm, which given any statement in the formal system, would determine whether or not that statement was true (in such a case the statement is a theorem). If Hilbert had achieved his objective, then any problem which was well defined could have been solved by simply executing the algorithm. Deciding the truth of a statement in a formal system was called the "Entscheidungsproblem" by Hilbert, who considered it to be a fundamental open problem in mathematics.

Unfortunately for Hilbert's objective, the 1930's brought a wave of research which showed that the Entscheidungs problem is not computable. That is, no algorithm of the type for which Hilbert longed, exists. The mathematicians, instead of heaving a sigh of relief (they would have been rendered jobless if Hilbert's program came through), were stunned by this discovery. The first result was due to Kurt Godel and it is the now famous incompleteness theorem. Among other things, Godel showed that there is no algorithm whose input can be any statement about integers and whose output tells whether or not the statement is true.

Closely following on Godel's heel, further Mathematicians such as Alonzo Church, Stephen Kleene, Emil Post, Alan Turing and many others, found more problems which had no algorithmic solution. Interestingly, most of these problems were found in the 1930's even before the first electronic computer was built. Perhaps the most important of these was the result due to the British Mathematician Alan Turning, which appeared in a paper titled "On Computable Numbers with an Application to the Entscheidungsproblem", which defined the notion of an algorithm and showed that the Entscheidungsproblem of David Hilbert was not computable.

Each of the above mathematician had to precisely define the notion of an algorithm, and each defined it in a different way. Godel defined algorithm as a sequence of rules for forming complicated mathematical functions out of simple mathematical functions, Church used a formalism called the lambda-calculus, while Turing used a mathematical object called the Turing machine and defined an algorithm to be any set of instructions for his simple machine. All these seemingly different and independently contrived definitions turned out to be equivalent and they form the basics of the modern theory of computing. No modern programming language can achieve more, in principle, than the Turing machine or the lambda-calculus. You'll learn about these definitions in the Theory of Computation course.