

## Chapter 1

We hear from different sources (newspapers, magazines, etc.) the uses of computers (such as high speed supercomputers performing trillions of math computations per second or networks that transmit data to any part of the world, etc. to minute computers that can be placed in our watches, books, clothing, etc.) but what is computer science?

First misconceptions about computer science:

1. **Computer Science is the study of Computers:** This is an incomplete definition. In fact computer science began much earlier than the advent of the first computer (between the 1920's to the 1940's and even before that). It used to be considered a branch of **logic and applied mathematics**. In fact, computer science should not be defined as the study of computers (even though it should not be ignored because it is integral) but should include more as pointed out by Fellows & Parberry:

“Computer science is no more about computers than astronomy is about telescope, biology is about microscope, or chemistry is about beakers and test tubes. Science is not about tools. It is about how we use them and what we find out when we do.”

2. **Computer science is the study of how to write computer programs:** Writing a program in Java, C++, etc. is only a means to an end. Scientists, business people, etc. who uses computer to achieve a result do so after long hours of developing a solution to the problem. To speed up processing, they convert their solution to a program which is executed by the computer to produce the result at a much faster rate. So computer programs (like computers) could also be viewed as a tool which is used to achieve a result that would lead to decision.
3. **Computer science is the study of the uses and applications of computers and software:** A computer scientist is responsible for *specifying, designing, building and testing* software packages – not just being able to use them (such as excel, word, etc.)

**These definitions are not wrong but as individual definitions, each is incomplete.**

### So what is Computer Science?

Note: the central concept in a proper definition of Computer Science must include **algorithm**.

So the Gibbs and Tucker definition seems to be very comprehensive:

**Definition: Computer Science is the study of algorithms, including:**

1. Their formal and mathematical properties.
2. Their hardware realizations.
3. Their linguistic realizations.
4. Their applications.

So it is the task of the computer scientist to design and develop algorithms to solve a range of problems. The process includes the following operations:

- Studying the behavior of algorithms to determine if they are correct and efficient (their formal and mathematical properties).
- Designing and building computer systems that are able to execute algorithms (Their hardware realizations).
- Designing programming languages and translating algorithms into these languages so that they can be executed by the hardware (their linguistic realizations).
- Identifying important problems and designing correct and efficient software packages to solve these problems (their applications).

To appreciate this definition of Computer Science, we need to look more closely at the meaning of **algorithm**.

Informally an algorithm is something similar to that of a recipe for cooking or assembling a shelf bought from the store. It consists of a step by step instruction to do something. These instructions can be either conditional and/or iterative, but they are always sequential.

However, in Computing, it does involve more than a step by step instruction.

**Definition:** *A well ordered collection of unambiguous and effectively computable operations that, when executed, produces a result and halts in a finite amount of time.*

Based upon this definition, we can categorize problems into two major groups – solvable and unsolvable. The latter group can further be broken into two sub-groups - those that could be solved, but will take thousands of years and those that really do not have a solution.

Let's look at the definition for algorithm more closely and see what it means.

...a well ordered collection...

If we look at the following algorithm found at the back of a shampoo bottle:

**Step 1: Wet hair**  
**Step 2: Lather**  
**Step 3: Rinse**  
**Step 4: Repeat**

Step 4 did not indicate what to repeat. It is not therefore well ordered. It should specifically state what to repeat.

...of unambiguous and effectively computable operations...

An ambiguous algorithm for baking a pie:

- Step 1: Make the crust
- Step 2: Make the cherry filling
- Step 3: Pour the filling into the crust
- Step 4: Bake at 350°F for 45 minutes.

To a novice cook, steps 1 & 2 may be very ambiguous indeed. It would need clarifications.

**An operation must not only be understandable but also doable.**

- Step 1: Generate a list  $L$  of all prime numbers  $L_1, L_2, L_3, \dots$
- Step 2: Sort the List in ascending order
- Step 3: Print the 100<sup>th</sup> element in the list,  $L_{100}$
- Step 5: Stop.

This problem is not doable because we cannot generate a list of all the prime numbers – infinity.

...that produces a result

It is possible to develop an algorithm that cannot produce the answer (i.e. the correct result) but instead will produce a result such as an error message, a red light warning, etc.

Use of the addition algorithm from page 7 will produce a result.

...and halts in a finite amount of time...

If we look at the shampoo algorithm, it never stops. We can modify it to stop:

- Step 1: Wet your hair**
- Step 2: Set the value of *WashCount* to 0**
- Step 3: Repeat steps 4 through 6 until the value of *WashCount* equals 2.**
- Step 4: Lather your hair**
- Step 5: Rinse your hair**
- Step 6: Add 1 to the value of *WashCount***

## Step 7: Stop.

**NOTE:** All operations used to construct algorithms belong to one of only three categories:

- a. Sequential operations: A single well defined task (or instruction)
- b. Conditional operations: The “question-asking” operations whereas the next operation is based on the answer to the question.
- c. Iterative operations: The looping operations of the algorithm. Doing the same thing over and over with maybe minor changes or no changes. Only an update condition so that we can stop at some point in time.

### **A brief history of computing:**

As we have noted, computer science deals with the development of some sort of a solution to a problem. In the earlier days, most of these were in the realm of science math, astronomy, etc. and so most were quantifiable. In fact, even today, most processing done by the computer are quantifiable.

The earliest of such implements was the “**Pascaline**”, a mechanical device developed by Blaise Pascal in 1672 that could do simple addition and subtraction. This was followed by the “**Leibnitz’s Wheel**” (Leibnitz studied Pascal’s and others work) in 1674, which could do addition, subtraction, multiplication and division. Both of these were based on interlocking mechanical cogs (wheels) and gears.

Note: These were mechanical machines which indeed helped to speed up calculations, but they were not true computers because they were not programmable (they had to be operated manually) nor did they have a memory.

The first device that was programmable and had memory (albeit on cards) was the “**Jacquard loom**” a machine developed by Jacquard in 1801 to manufacture rugs and clothing. The most important contribution of the Jacquard loom is that it shows that at that time, a skilled craftsman had the potential to automate his/her work. He/she could devise and develop a machine to do his/her work removing the need for the skilled human to do the same work. Now, a very unskilled person could do the same work – all they need to learn is some basic skill as to how to use the machine. With such development, production increases and productivity increases as well.

These break through had enormous impact on later scientist including Charles Babbage who actually developed the “**Difference Engine**” that was capable of addition, subtraction, division, multiplication up to 6 significant digits as well as solving polynomial problems among others. It could be regarded as the father of computer and the real beginning of computers, (even though it could be argued that Pascal was the father of the computers). He contemplated a more sophisticated machine, the “**Analytic Engine**” (1820’s – 1830’s) which he was unable to complete during his life – it was too advanced for his time. However, this machine was built in 1991 according to his specification and worked perfectly. This machine was very much like the modern

computer – consisting of a **mill [ALU]** (performs the math operations), **store [MEMORY]** (to hold data), **operator [PROCESSOR]** (to process operations contained on punch cards) and **output unit [INPUT/OUTPUT]** (to store result on punch cards)

Another person influenced by the above was Herman Hollerith (1890). He devised the **Keypunch** machine to put US Census data onto cards. These cards were then taken to either a **Tabulator** or a **Sorter**, both of which were programmable. These machines show how data could be processed rapidly and he left the US Census to eventually start up his own company (in 1902 – the Computer Tabulating Recording Company) which was so successful that by 1924 it became international and so he renamed his company to International Business Machine (IBM).

### **The birth of Computers: 1940 – 1950**

The advent of war gave a new impetus for the development of the computers to be used in the military. In 1931, the US Navy and IBM jointly funded Prof. Howard Aiken from Harvard University who build the Mark 1. This was the first computer to use the binary numbering system – it used vacuum tubes and relays to represent 0 and 1. This machine was completed in 1944 and is generally considered as one of the first computers. At about the same time in conjunction with the Army, Eckert and Mauchly from the University of Pennsylvania developed the ENIAC (Electronic Numerical Integrator and Calculator) – completed in 1946 and was the first electronic and programmable machine. It was a thousand times faster than the Mark 1. It contained 18000 vacuum tubes and was some 30 tons in weight.

Of course others were building computers at the same time (Atanasoff and his student Berry) but they were used for more specialized purpose and did not receive the recognition as the Mark 1 and the ENIAC. Furthermore, in England, Alan Turing directed the construction of the Colossus (1943) – used to crack the German U-boat code. The Colossus did not receive the recognition as the ENIAC because of its secret purpose. Similarly, in Germany, Konrad Zuse developed the Z1 (similar to the ENIAC).

Even all of these all of these computers were computers in the real sense – they were electronic devices, were programmable and had memory, they were not like the modern computer. In 1946, Von Neumann proposed that the computer should be built based on the “**Stored program concept**”. Prior to this, all computers used external wires and plugs and even though there was memory, it was used to store data only – not **instructions**. He proposed that the instructions be made “**general purpose**” and be and be stored in memory as well so rather than rewiring the machine to run a program, we write new programs. This also gave birth to programs as it is known today. His model has come to be known as the **Von Neumann architecture** and it is the basis on which all modern computers are built. The first of these computers was the EDVAC (Electronic Discrete Variable Automatic Computer - in the US) and independently built in England by Maurice Wilkes, was the EDSAC (Electronic Delay Storage Automatic Calculator). A commercial version of the EDVAC – the UNIVAC 1 was the first computer built and sold – to the US census Bureau in 1951.

## The modern era: 1950 to the present

For the last 50 years or so, computers have evolved. The Von Neumann architecture have been improved to modern day computers – that is, it is faster, with more memory, dual processors, bigger buses, etc. but the basic architecture remains basically the same.

**First Generation (1950 – 1957):** UNIVAC 1 and IBM 701 – used enormous amounts of vacuum tubes. They were bulky, expensive, slow and unreliable. Even to turn on the machines, large quantities on vacuum tubes may blow with the surge of power. Used only in large corporations, government and university research labs and military installations.

**Second Generation (1957 – 1965):** The vacuum tube replaced by the **transistor** (very small) and so the size of the computer started to get smaller and smaller. Now more small and medium businesses and colleges, etc. can afford a computer. They were more reliable and faster. This lead to the development of programming languages such as **FORTRAN** (Formula Translated Language) and **Cobol** (Common Business Oriented Language) – both of which are English-like and are considered as high level programming languages giving birth to **Programmers**.

**Third Generation (1965 – 1975):** The era of the **integrated circuit**. The discrete components as used in Transistors were no longer used. Instead, Integrated circuits were built using transistors, resistors and capacitors all of which were used to create a circuit and that circuit etched into silicon which further lead to the reduction in the size of the computers and the increase in the speed. This period saw the birth of the **minicomputer** (*supporting between 4 and 200 users simultaneously and lies between a workstation and a mainframe*) – the **PDP 1** and the birth of the software industry.

### **Fourth Generation (1975 – 1985):**

The first microcomputer (*“Although there is no rigid definition, a **microcomputer** (sometimes shortened to **micro**) is most often taken to mean a computer with a microprocessor ( $\mu P$ ) as its CPU. Another general characteristic of these computers is that they occupy physically small amounts of space. Although the terms are not synonymous, many microcomputers are also personal computers (in the generic sense) and vice versa.*

*The microcomputer came after the minicomputer, most notably replacing the many distinct components that made up the minicomputer's CPU with a single integrated microprocessor chip. The early microcomputers were primitive, the earliest models shipping with as little as 256 bytes of RAM, and no input / output other than lights and switches. However, as microprocessor design advanced rapidly and memory became less expensive from the early 1970s onwards, microcomputers in turn grew faster and cheaper. This resulted in an explosion in their popularity during the late 1970s and early 1980s. The world's first commercial microprocessor was the Intel 4004, released on November 15, 1971. The 4004 processed 4 binary digits (bits) of data in parallel; in other*

words, it was a 4-bit processor. At the turn of the century 30 years later, microcomputers in embedded systems (built into home appliances, vehicles, and all sorts of equipment) most often are 8-bit, 16-bit, 32-bit, or 64-bit. Desktop/consumer microcomputers, like Apple Macintosh and PCs, are predominantly 32-bit but increasingly 64-bit, while most science and engineering workstations and supercomputers as well as database and financial transaction servers are 64-bit (with one or more CPUs)” taken from <http://en.wikipedia.org/wiki/Microcomputer>.), was made up of a box consisting of switches – the Altair 8800 after which they rapidly advanced to the PCs of today. This rapid development came about due to the appearance of networking, wide spread use of spread sheets, databases, drawing programs, etc. E-mail was introduced along with the concept of GUIs to make the computer more user friendly.

### **Fifth Generation (1985 – ?):**

Owing to the pace of development in computers and computer accessories, it is difficult to continue to categorize computers into a generation scale. As pointed out by the authors (on page 27 – 28), the computer systems now include:

- Massively parallel processors capable of trillions of computations per second.
- Handheld devices and other types of Personal Digital Assistants (PDAs).
- High-resolution graphics for imaging, making movies and virtual reality.
- Powerful multimedia user interfaces incorporating sounds, voice recognition, touch, photography, video and television.
- Integrated global telecommunications incorporating data, television, telephone, fax, the Internet, and the WWW.
- Wireless data communications.
- Massive storage devices capable of holding hundreds of trillions of pieces of data
- Ubiquitous computing in which miniature computers are embedded in cars, kitchen appliances, home heating systems and even clothing.

In only 50 – 60 years, computers have progressed from the UNIVAC 1 (very expensive, very slow, and can store very little data) to the PCs today (less than \$2000.00, very fast, and has immense memories). In no other areas of human invention and development has there been a greater achievement and progress.