CSC211 Assignment 0:

Milestones in the History of Computing

Group Members:
Masoud Faramarzi
Michael Eiger
Abigail Kaye
Gregory Bassett
Devin Croteau

Date: February 12, 2021

Table of Contents

Charles Babbage's and Ada Lovelace’s Contributions

Charles Babbage was an English mathematician and inventor born in 1791 (Editors of Encyclopaedia Britannica, 2020). Around the year 1812, he conceived the idea of possibly calculating arithmetic mechanically. This led him to create one of the first calculators during this same time period, which could compute arithmetic to eight decimal points. In 1823, he received support from the government to design what he would call the Difference Engine. It used a memory system similar to modern computers and he had hoped that it would be able to compute values up to twenty decimal points. However, after almost ten years of construction, one of his lead inventors refused to complete the project and it was never built. During the mid-1830’s, Babbage began to work on an entirely new project. He called this device the Analytical Engine and used punch cards, a form of memory used to store numbers and sequences, to give the machine its computing power. The memory was large enough to hold 1000 50-digit numbers, which was larger than any other computer could handle before 1960.

Babbage had many contributors to his computer projects, but one of the most notable among them was Ada Lovelace. Lovelace was another English mathematician and is known today as the world’s first computer programmer (Editors of Encyclopaedia Britannica, 2021). She was impressed by Babbage’s inventions when they met in 1833 by a mutual friend, but she did not help contribute to his inventions until 1843. During this year, Lovelace helped translate an article written by Italian mathematician and engineer Luigi Federico Menabrea. With this detailed account and her keen mind, she helped make many memorable contributions to the Analytical Engine. She even proposed a program that could possibly be used to compute Bernoulli numbers. Due to her accomplishments over the years in the world of computer science, an early programming language called Ada was even named after her. Babbage always saw computers as a method to compute arithmetic and solve problems, but Ada’s appreciation for computers took it a step further. She saw the potential that computers possess and envisioned that they could be used for much more. She always dreamed that they could be used to produce or play music (Essinger, 2021). She never knew just how much she really contributed to making that dream a reality.

Although the Difference Engine and the Analytical Engine were never built, the construction was a rigorous process that was built upon for years to perfect the computer. In fact, a team of researchers in 2002 built a replica of the Difference Engine. They only used components and tools that would have been available to Babbage, except for the cogwheels which were made using modern techniques. The computer was completed and proved that it could have been possible for Babbage to complete the project during his lifetime. It’s important to study and learn about these two brilliant minds because they set the foundations for what a computer is today. Babbage and Lovelace were the first visionaries in computer science and saw all that computers had to offer. Without their help, it could have taken decades before the first computer was built. This would have led to a massive decrease in the technological advancements that we use and appreciate every day.

Herman Hollerith's Punch Card System

Herman Hollerith was an American statistician, businessman, and inventor who is known for his development of the punch card system. This system is widely considered the precursor to modern automatic computation. In 1880, Hollerith was employed by the U.S. Census Office after graduating from Columbia University with an engineering degree (da Cruz, 2021). The census consisted of “not only a population count, but also include[d] the gathering and analysis of data on industry, agriculture, and other aspects of the economy” (Kistermaan, 1991, p. 2). Data collection and analysis proved to be quite cumbersome and laborious at the time because all of the work was done by hand. The human aspect of computation also resulted in a high risk of error.

Hollerith recognized the need for a more efficient and effective method of collecting and analyzing census data. Inspired by the Jacquard loom, “an automatic weaving machine that is controlled by specially coded punch cards,” and observations of conductors punching train tickets, he decided to use punch cards to automate the collection and analysis of data gathered during the census. Hollerith went on to design a tabulating machine “that used the location of holes on each card to tally not only overall numbers but also individual characteristics and even cross-tabulations” (Gauthier, n.d.).

Hollerith’s system was highly beneficial, greatly reducing the time needed to collect and analyze census data. The system was also capable of “providing more statistics at a lower cost for processing” (Gauthier, n.d.). The system’s success was recognized internationally, as evidenced by “the Austro-Hungarian Empire [being] the first to decide that the system was worth using for a census in Austria to be conducted on December 31, 1890” (Kistermaan, 1991, p. 8). Over time, Hollerith’s system was further improved and expanded upon. The punch card system, along with his other inventions, are seen as “the foundation of the modern information processing industry” (da Cruz, 2021). His work is significant because it helped lay the foundation for automatic computation we use today.

The Turing Machine

In 1936 Alan Turing established the idea of Turing machines. A Turing machine has the ability to simulate any computer algorithm. While this machine is only hypothetical, the idea of one works as follows: The memory is represented by a tape that is infinitely long. The spaces on this tape are either blank by default or have symbols on them (“Turing Machine”, n.d.). The machine can do one of three things with said tape: read the symbol on the space it is focusing on, change the symbol or erase it entirely, or shift the tape in either direction (“What is a…”, n.d.). A Turing machine can be constructed to simulate any finite logic based algorithm, called finite control. The creation of these impossible machines proved that there are limitations to the power of mechanical calculations. Real life computers work on a much different design called random-access memory.

Turing completeness describes a system that can perfectly simulate a Turing machine. One such programming language like this is said to be capable of showing all tasks that can be completed by computers. In order for something to be proven Turing complete it has to simulate another Turing complete system. This is typically done by simulating a universal Turing machine (“Turing Machine”, n.d.). Most programming

languages are considered Turing complete, but only if the requirement of infinite memory is ignored.

A universal Turing machine is another kind of hypothetical machine, this one capable of mimicking the behavior of any other Turing machine (Vitanyi, 2009). Experimenting with the idea of Turing machines led Alan to the Church-Turing thesis, which concludes that any computable problem can be run on a Turing machine (Vitanyi, 2009). A perfect Turing machine would show that more powerful computers are not needed to solve these problems.

The significance of Turing machines and their creation is in their nature. Because any machine can have its algorithms computed on a Turing machine, any limitation on theoretical Turing machines apply to real computers as well. This thought experiment led to a much wider and stronger understanding of the idea of computability. While our machines are fundamentally different from Turing machines, they are limited in the same ways.

Electronic Numerical Integrator and Calculator (ENIAC)

The Electronic Numerical Integrator and Calculator, otherwise known as the ENIAC, was the one of the most successful programmable computers built by the United States (Freiberger, 2008). Its construction began in early 1943, during World War II. The army sponsored this research because it would have been extremely beneficial for calculating a target position for artillery. It was created by John Mauchly, J. Presper Eckhert Jr., and other colleagues at the Moore School of Electrical Engineering at the University of Pennsylvania. This computer was finally completed in February 1946. Since the war ended by the time the ENIAC was built, it was not used for its original purpose of artillery targeting during World War II.

The ENIAC was one of the fastest computers of its time and was considered a huge success (Freiberger, 2008). However, it took days to rewire the machine due to the technological limitations of this time period. It also fully occupied a room 30 by 50 feet, weighed 30 tons, and consumed 150 kilowatts of electricity. Despite this, it still had computing speed and capabilities that no computer during this time period could compare with. Its increase in speed was attributed to the use of vacuum tubes instead of switches and relays (Bellis, 2020). It was also one of the first computers to utilize conditional branching. This means that it could carry out different instructions or change the order of these instructions based on the values that were received.

Although the war was over by the time this computer was built, it still had many useful functions. The ENIAC was used to help design a hydrogen bomb, create weather predictions, design wind-tunnels, study cosmic rays, and much more. Furthermore, it created inspiration for other inventors. The ENIAC was extremely efficient and successful, proving that computers could be built and possibly expanded upon to be even more efficient if research was continued. Over the years, computers became more efficient, smaller, and more affordable. The ENIAC helped inspire the construction of the Electronic Discrete Variable Automatic Computer, otherwise known as the EDVAC. Creating the ENIAC also helped expand research into making vacuum tubes more efficient, which eventually led to the creation of transistors and microprocessors. Soon enough, IBM began making computers and they became affordable by the average consumer by the mid-1970's (University of Pennsylvania, 2017).

Von Neumann's Architecture

Von Neumann's Architecture (VNA) was published by John Von Neumann in 1945 ("Von Neumann…", 2019). In his work, Von Neumann outlined the concept of a stored program and how computers could process program information (Cabrera, n.d.). The VNA design contains a central processing unit (CPU), arithmetic and logic unit (ALU), memory unit, control unit (CU), and inputs / outputs. This idea is focused on instruction and program data being stored in the same memory. Each component of the VNA design yields a different function. For instance, the CPU is responsible for executing commands from programs within which said commands originate. The ALU performs simple mathematical operations (e.g., addition, subtraction) and logic gates (e.g., and not, or) ("Computer Organization…", 2018). The registers store information derived from operations performed by the ALU before it is processed by the CPU. In addition to handling timing, the CU controls the ALU, memory, and input and output devices. Buses send and receive information and connect to CPU and memory. The memory unit consists of random access memory (RAM), which is a partition-based memory storage unit, and a hard drive, both of which work in tandem to allow the CPU to run more efficiently.

The design Von Neumann came up with is still widely used in computers today. Specifically, the VNA structure dictates the means by which most stored-program computers have processed program information since its inception (Cabrera, n.d.). Von Neumann's work is thought to be both accessible and enduring given his paper from 1945 in which he detailed his architecture, "First Draft of a Report on the EDVAC", presents a logical description of the structure. This approach is opposed to his contemporaries describing alternate computer architectures with an emphasis on engineering jargon, rendering such structures less universally understood and thus less frequently adopted. Von Neumann's work is further groundbreaking because it consequently established the concept of a Von Neumann bottleneck (VNB) in 1977 (Belstaff, 2006). Relative to the VNA, VNB relates to the reality that data cannot be retrieved/stored and processed at the same time. This is because VNA structure requires said operations to demand resources from a common bus. Modern-day computers are tasked with finding ways to optimally work around this structural limitation accordingly.

UNIVAC

The Universal Automatic Computer, also known as UNIVAC, was the first commercial computer developed for business and government applications. The computer was developed by John Mauchly and Presper Eckert, who also created the Electronic Numeric Integrator and Computer (also known as ENIAC). While working on ENIAC, Mauchly met with Census Bureau officials to discuss the potential uses for computers outside of military contexts. Similar to Hollerith's punch card system, UNIVAC was initially developed between 1948 and 1951 as a statistical tabulator for the Census Bureau (Gauthier, n.d.). The computer's Uniservo magnetic tape drive allowed it to spend more time processing data at faster speeds than other business computers could (Computer History Museum, n.d.). Faster, more efficient data processing was beneficial to companies because it could save money and produce results faster.

UNIVAC was able to process about 1,000 calculations per second (“UNIVAC, the first…”, 2010).

Over time, UNIVAC's use expanded from government into business. General Electric was the first commercial customer to use UNIVAC. Using UNIVAC, General Electric was able to automate various processes, such as payroll, budget reports, and inventory. In these ways, UNIVAC demonstrated its utility as a tool for business by successfully saving companies both time and money by developing reports and performing calculations (Computer History Museum, n.d.). It also had uses beyond government and business. One example is UNIVAC famously predicting Dwight Eisenhower’s victory in the 1952 presidential election (“UNIVAC, the first…”, 2010). In many ways, UNIVAC proved to be an asset and its use expanded. However, “[b]y the mid 1950s, UNIVAC’s lead over IBM had evaporated, thanks to poor marketing, delayed products and new models from IBM” (Computer History Museum, n.d.). Ultimately, UNIVAC was significant because it expanded the use of computers from the military into commercial and government sectors. Thanks to this expansion, computers would become increasingly accessible to ordinary people, fostering the age of computers.

Invention of the Transistor

A transistor is a device that conducts electricity and controls its strength. A transistor allows a weak signal to have control over a higher flow. In early stages of its development it was simply a vacuum tube. Although it got the job done, it was not efficient in doing so. Much electronic power would be expended in the form of heat that would greatly reduce its lifespan (Watkins, n.d.). This temporary solution is what paved the way for the transistor to be used widely in electronics. It had both efficiency and size over the vacuum tube, as well as an increased lifespan for it doesn’t release nearly as much heat (Watkins, n.d.).

The transistor was developed by William Shockley, John Bardeen, and Walter Brattain and was first demonstrated on December 23, 1947 at Bell Laboratories. Shockley developed a theory about such a contraption ten years before it was actually made, but failed to implement his ideas (Watkins, n.d.). Bardeen and Brattain were the ones who actually developed the point-contact transistor, experimenting with and modifying the transistor to increase the amplification of electrical signals (Ganapati, 2017). Their success was subsequently followed up by Shockley further honing the transistor in order to create the bipolar transistor. The bipolar transistor is what is widely used in modern electronics. As a result, Shockley is mainly credited as the inventor of the transistor (Watkins, n.d.).

The transistor has been dubbed as the most important invention of the 20th century, and for good reason. This device allows for the amplification of electrical signals. It controls the flow to other parts of the circuit or device. Without the transistor electronics would not be nearly as developed as they have become, nor would they be as versatile (Ganapati, 2017). Ultimately, the transistor is significant because it is the foundation of modern computer technology.

Grace Hopper and the First Computer Language

Grace Hopper was an American computer scientist and United States Navy rear admiral. Hopper worked on Mark 1, a computer prototype, wrote a Manual of Operations for the Automatic Sequence-Controlled Calculator, and is credited with coining the term “bug” (Norwood, 2017). Among these achievements, Hopper is largely known for developing the first computer language. Between 1955 and 1959, Hopper worked for Remington Rand and developed FLOW-MATIC. Prior to the introduction of this language, data processing was performed using mathematical notation and symbols. FORTRAN is one example of a computer language that used mathematical symbols (“Grace Murray...”, 2017).

The use of mathematics in processing could make it more difficult for people who weren’t comfortable using such notation and symbols. Hopper recognized the need to make data processing more accessible for people. She believed that those who worked as data processors, “who were not typically mathematicians or engineers, would be more comfortable using word-based languages.” As she explained in one interview, “What I was after in beginning English language [programming] was to bring another whole group of people able to use the computer easily …. I kept calling for more user-friendly languages. Most of the stuff we get from academicians, computer science people, is in no way adapted to people” (“Grace Murray...”, 2017). She subsequently developed FLOW-MATIC, which converted English terms into machine code that could be understood by computers. Thanks to this language, data processing could be performed by someone who didn’t have a background in mathematics, engineering or computer science (“Grace Murray…”, 2017). As the first processing language to incorporate English-like statements, FLOW-MATIC is significant because it made programming more accessible and easier for people to comprehend, while also paving the way for natural-language programming and processing.

Invention of Integrated Circuits

Integrated circuits can be defined as a collection of electronic circuits on one chip containing semiconductors (Horowitz & Hill, 1989, p. 61). Integrated circuits are usually made of silicon. The invention of integrated circuits occurred in 1958 to address deficiencies related to transistors. Transistors had become common in electrical appliances such as radios. However, there were drawbacks to transistors that prevented them from being used in more complex electronic devices. One such limitation of transistors was their size, as they had to be large enough to be connected to wires. The main idea of making an integrated circuit was to create a circuit that could synthesize transistors, wires, and other electrical conduction components. Coupled with their size, the cost of transistors also made working with many of them at once complicated (Saint, n.d.)

In July 1958, Jack Kibly figured out that all components could be made out of silicon, not only transistors (The Computer History Museum, 2021). At that time, capacitors and resistors were generally not made out of silicon. Another inventor, Robert Noyce, was working on the same idea in California. Kilby’s idea was better

regarding the individual components, while Noyces's idea was more advanced regarding connections between different components. Finally, in 1961, a patent for the integrated circuit was registered by Mr. Robert Noyce. Despite Kilby filing his patent earlier, it was still in the analysis step and finally was approved as late as June 1964. Even though both inventors are acknowledged for their important roles relative to the integrated circuit's invention, only Kilby was recognized as a Nobel Prize winner for being the "inventor" of the integrated circuit. In contrast, Noyce passed away in 1990 without receiving any accolades for his work (Public Broadcasting Service, 1999). Overall, integrated circuits were notable because they addressed the limitations posed by transistors by improving factors related to their size, speed, and cost.

ARPANET

The Advanced Research Projects Agency Network (ARPANET) was established by the United States Department of Defense In 1969 (Bidgoli, 2004). ARPANET is considered an experimental computer network that was the Internet's forerunner (Schneider et al., 2009). Its users initially regarded this network as confusing to utilize and thus was not very popular around its inception. Originally, the most dominant application of this network was limited to cases in which researchers were traveling or permanently relocating and hoped to enable their work's portability. Those researchers were familiar with the procedures used in the other location and thus knew how to talk to those host computers in their previous department. A group of graduate students called Network Working Group (NWG) designed software to make a host-to-host network. Those graduate students were from the existing or potential ARPA network sites. By 1970, ARPANET had its first host-to-host connection, which expanded by several more host-to-host connections by the year's end. Gradually, the number of hosts and nodes continued to increase in ARPANET (Featherly, 2021). Over the next few years, ARPANET also began to establish nodes internationally.

In the following years, other packet-switching networks that were previously operating independently of ARPANET were added to the network (Roberts, 2001). Combining different networks with different structures was a challenge for the developers. To solve this problem, they figured out a concept called "internetworking," or open-architecture networking, in 1972 (Leiner et al., 2021). Each of those individual networks was designed to make host-to-host connections within their network. Such a system was finally developed by 1978 and was called Transfer Control Protocol (TCP/IP) (Navarri, 2020). There were many more networks by the late 1970s, including Computer Science Research Network (CSRN), Canadian Network (CDnet), Because It's Time Network (BITNET), and National Science Foundation Network (NSFNET). Out of ARPANET's contemporaries, NSFNET eventually replaced it as the backbone of the Internet, although NSFET too was replaced with commercial networks, which is the internet we know today (Featherly, 2021). Despite being dwarfed relative to both utility and popularity by commercial networks, the internet as it is presently known may not exist without ARPANET providing its foundation.

UNIX Operating System

UNIX was an operating system (OS) that predates both OSs such as Windows. UNIX was developed through collaboration between the Massachusetts Institute of Technology and AT&T Bell Labs in 1969 (Raymond, 2003). AT&T Bell Labs developed several versions of UNIX, which were collectively regarded as a "research UNIX." The UNIX OS includes many libraries and utilities along with the kernel, which is the OS's core computer program (Stonebank, 2000). UNIX includes the components that make it a "self-contained" OS, including: the OS kernel and shell, background services, graphics server, desktop environment, and applications ("Unix vs…", 2021).

UNIX OSs gradually added new features and refinements as new versions were released. However, the most prominent refinement entailed the eventual rewriting of UNIX in the C programming language in 1973 instead of the assembly language it was initially programmed with (Ritchie, 1993, p. 9). This rewrite established portability among UNIX OSs, which was a novel feature at the time. As a result, UNIX's popularity significantly increased over the next couple of decades, to the point that over 90% of the top 500 supercomputers during the 1990ss were using UNIX as their operating system (The Open Group, 2021). Many OSs have since been developed based on the UNIX idea, including Linux, which have dwarfed UNIX's popularity amongst modern-day users. However, UNIX OSs introduced a concept that remains relevant to all computer users: the universal link locator (URL) (Raymond, 2003). UNIX conceptualized this feature through its use of one uniform file namespace to access specified locations. Therefore, the modern-day perception of internet accessibility could have been morphed considerably without UNIX's conceptual contributions.

The C Language

C is a systems-programming language that Dennis Ritchie developed at AT&T Bell Laboratories beginning in 1971 (Ritchie, 1993, p. 6). C is based on programming foundations established by both Basic Combined Programming Language (BCPL) and B, programming languages released in 1967 and 1969, respectively (Raymond, 2003, p. 438). However, C aimed to resolve aspects of B's function considered by Ritchie to be insufficient for utilization on PDP-11, the current version of the UNIX operating system at the time (Ritchie, 1993, p. 6). B's inadequacies as perceived by Ritchie included its use of library procedures to handle characters, its inability to perform floating-point arithmetic natively, and its means of defining pointers. Ritchie addressed said inadequacies by adding both character (i.e., char) and integer (i.e., int) type declarations. Ritchie also modified B's compiler implementation for C so it could produce programs competitive with those produced by assembly languages in terms of both program speed and size. At this time, the initial version of C was not yet officially named "C," instead referred to by Richie as "New B."

Between 1971 and the beginning of 1973, Ritchie modified "New B" to include several further innovations (Ritchie, 1993, pp. 7-8). The primary innovation was that array-type values in expressions are converted from pointers to the first object that comprises a given array (p. 7). This modification to array-type objects morphed the semantics of arrays and rendered them a more comprehensive data structure. An additional modernization was that syntactical adjustments were made for type structure

declarations. As a result, users could gather objects of several types into an array, derive objects from a function, or use an object as a pointer to any previously created object in a program. After establishing these conventions towards the beginning of 1972, the programming language was renamed “C” (p. 8). The name change was succeeded by introducing the && (i.e., and) and || (i.e., or) style of operators. The operators & (i.e., and) and | (i.e., or) were present in B but had scenarios where their boolean value was evaluated incorrectly because said operators had also had contexts for use beyond boolean logic. The last significant development in C during this period was the inclusion of a preprocessor in 1973. This innovation initially only enabled the #include and #define parameterless macros called at the beginning of C programs but was extended to include macros that took parameter arguments by the end of 1973.

Previously coded in DEC, the innovations facilitated by C’s development as described above enabled Ritchie and his colleagues to rewrite PDP-11’s UNIX kernel using C during the summer of 1973 (Ritchie, 1993, p. 9). A fellow computer scientist at ABL, Steve Johnson, concurrently developed the Portable C Compiler (PCC) to compile programs coded using C after foreseeing portability issues associated with the C language and Ritchie’s work. PCC enabled the transfer of C programs and portions of UNIX coded in C across UNIX systems to more modern machines. PDP-11’s successful rewrite in C and the creation of the PCC led Ritchie’s team to rewrite many system utilities and tools in C between 1973 and 1980. The portability of UNIX operating systems and its programs as enabled by C resulted in a significant growth in popularity over the rest of the decade (p. 13).

Although C underwent further changes during the period from 1973 to 1980, news of its portability resulted in the language’s widespread use amongst computer scientists both within and outside of AT&T Bell Laboratories (pp. 9-10). Furthermore, additional C compilers were constructed around this time based on PCC to ensure C programs’ compatibility across operating systems and architectures (p. 10). The use of C continued to inflate in popularity due to the well-established portability of UNIX and C programs compiled using PCC and its contemporaries during the 1980s. As a result, C became one of the most widely adopted programming tools for both professional and commercial use. C’s relevance to modern-day programming is exemplified by how the TIOBE Index currently ranks it as the most popular programming language among computer scientists (“TIOBE Index…”, 2021).

Altair 8800

The Altair 8800 Microcomputer was created by a company named Micro Instrumentation and Telemetry Systems (MITS) (“Altair 8800…”, n.d.). MITS’s product was developed in 1974 and released in December of the same year. The Altair 8800 was a response to MITS’s financial troubles that they were experiencing due to unremarkable sales of their previous primary line of products: electronic calculator kits (Alfred, 2008; “Altair 8800…”, n.d.). The machine sold as a kit for 395 dollars and as a pre-assembled unit for 498 dollars (“Altair 8800…”, n.d.). The Altair 8800 base models were initially sold with only 256 bytes of internal memory; enough space to store one sentence of text (Alfred, 2008). The base machine also did not come with any external memory, input devices, display, or other accessories, although MITS sold these components and others separately for an additional fee. Despite its limitations, the Altair

8800 successfully sold roughly 5,000 units by August 1975, which significantly exceeded the MITS's internal expectations and proved the existence of a financially viable consumer market for personal computers ("Altair 8800…", n.d.).

Altair 8800's sales did more than enable a technology company to experience financial success in a new type of consumer market. Shortly after the Altair 8800 was released, it was featured in the January 1975 issue of *Popular Electronics*, a technology-oriented magazine that existed at the time (Alfred, 2008). This January 1975 issue of the magazine attracted various computer hobbyists including two enthusiasts of note: Bill Gates and Paul Allen. Upon viewing the issue's cover, which contained a cover of the Altair 8800, Gates and Allen immediately contacted MITS and offered them a deal to integrate a programming language called Beginners' All-Purpose Symbolic Instruction Code (BASIC) (Freiberger & Swaine, 2000, p. 53). BASIC was still vaporware at the time of their correspondence with MITS. However, MITS accepted Gates's and Allen's offer after being shown a demonstration of a program constructed using BASIC running on the Altair 8800 several weeks later (p. 54). MITS founder Ed Roberts was impressed with the demonstration and purchased the BASIC interpreter. This sale was the first of Gate's and Allen's company called Traf-O-Data, which was then renamed to Micro-Soft. The Altair 8800 microcomputer was responsible for ensuring that the company, today known as Microsoft, remained focused on personal computing products; a consumer sector in which the company would eventually experience remarkable financial success.

'C with Classes' and C++

"C with Classes" was an extension of the C programming language that Bjarne Stroustrup began developing at AT&T Bell Laboratories (ABL) in April 1979 (Stroustrup, 1996, p. 3). Stroustrup's intended to add functionality to C that was originally present in another programming language, Simula, such as support for classes and catching type errors during program compilation (p. 2). Stroustrup hoped to enhance both C's general capabilities and overall readability of C code by introducing such functionality while also leveraging C's inherent speed and portability during software development. His efforts first resulted in a pre-processor for C with Classes termed Cpre, first released in October 1979 with updates over the following few months (p. 4). By March 1980, Cpre enabled desired Simula features in C such as classes and derived classes, call and return functions, type checking, and default arguments (pp. 5-6). Compatible with machines already supporting C, C with Classes was ported to various minicomputers available at the time, such as the respective DEC VAX and Motorola 68000 machines (p. 6). Despite its moderate success, Stroustrup strived to create a language with greater functionality than the most recent version of C with Classes.

In 1982, Stroustrup's ambitions resulted in him developing a new programming language called C84, the language that would later become known as C++ (Stroustrup, 1996, p. 17). Stroustrup regarded C84 as a new programming language that utilized C with Classes as a foundation and mandated the creation of various user libraries and tools. The first compiler for C84, Cfront, began development in 1982 and was released in 1983 (p. 18). Over the same period, Stroustrup added several significant features to C84 that were not present in C with Classes, such as function names, operator

overloading, references, and user-controlled free-store memory control, among others (p. 21). In December 1973, C84's name was formally changed to C++ to prevent it from being regarded as an extension of C (p. 17). In 1984, Stroustrup implemented an additional modernization within C++: the stream input/output library (p. 38). The stream input/output library was the first of its kind by enabling data, stored as bytes, to be passed between a C++ program and the input/output sources. C++'s innovations were made available for public use upon its version 1.0 release in October 1985 (pp. 1, 31).

As enabled by the innovations described above, C++ experienced a progressive increase in popularity over the remainder of the decade as interest in its commercial and scientific applications grew (Stroustrup, 1996, p. 42). As a result, the language was standardized by the American National Standards Institute (ANSI) in March 1990, which demystified some of the functional and syntactical differences between C and C++. C++ has since been further refined and re-standardized to include features such as general templates (1991), exception-based error handling (1991), namespaces (1997), and hash tables (2003) (Stroustrup, 2018, p. 154). Over time, it has been used for software development and scientific computing purposes across various industries encompassing fields such as finance, technology, and medicine (Stroustrup, 2020). C++ continues to be updated in 2021 and is presently the fourth most popular programming language as determined by TIOBE Index ("TIOBE Index…", 2021).

Works Cited

Alfred, R. (2008, September 07). Dec. 19, 1974: Altair 8800 kits go on sale. https://www.wired.com/2008/12/dec-19-1974-altair-8800-kits-go-on-sale/

Altair 8800 Microcomputer. (n.d.). Retrieved February 08, 2021, from https://americanhistory.si.edu/collections/search/object/nmah_334396

Bellis, M. (2020, January 13). The History of the ENIAC Computer. ThoughtCo. https://www.thoughtco.com/history-of-the-eniac-computer-1991601

Belstaff, T. (2006, July 23). *A Review of the 1977 Turing Award Lecture by John Backus*. E.W.Dijkstra Archive. https://www.cs.utexas.edu/~EWD/transcriptions/EWD06xx/EWD692.html.

Bidgoli, H. (2004). In *The Internet Encyclopedia* (Vol. 2, pp. 39–39). John Wiley & Sons.

Cabrera, B. J. (n.d.) *John von Neumann and von Neumann Architecture for Computers*. http://w3.salemstate.edu/~tevans/VonNeuma.htm.

Computer Organization: Von Neumann Architecture. (2018, May 9). https://www.geeksforgeeks.org/computer-organization-von-neumann-architecture/

Editors of Encyclopaedia Britannica. (2021, January 14). Ada Lovelace | Biography, Computer, & Facts. Encyclopedia Britannica. https://www.britannica.com/biography/Ada-Lovelace

Editors of Encyclopaedia Britannica. (2020, December 28). Charles Babbage | Biography, Computers, Inventions, & Facts. Encyclopedia Britannica. https://www.britannica.com/biography/Charles-Babbage

Essinger, J. (2021). The History Press | Charles Babbage and Ada Lovelace: The computer's most passionate partnership. The History Press. https://www.thehistorypress.co.uk/articles/charles-babbage-and-ada-lovelace-the-computer-s-most-passionate-partnership/

Featherly, K. (2021). *ARPANET*. Britannica. https://www.britannica.com/topic/arpanet/a-packet-of-data.

Freiberger, P. (2008, October 7). ENIAC | History, Size, & Facts. Encyclopedia Britannica. https://www.britannica.com/technology/ENIAC

Freiberger, P., & Swaine, M. (2000). *Fire in the Valley: The Making of the Personal Computer* (2nd ed.). New York City, NY: McGraw-Hill.

Ganapati, P. (2017, June 04). Dec. 23, 1947: Transistor opens door to digital future. Retrieved February 13, 2021, from https://www.wired.com/2009/12/1223shockley-bardeen-brattain-transistor/#:~:text=Hill%2C%20New%20Jersey.-,It's%20been%20called%20the%20most%20important%20invention%20of%20the%2020th,and%20consumed%20too%20much%20power

Gauthier, J. (n.d.). Herman Hollerith - History - U.S. Census Bureau. Retrieved February 12, 2021, from https://www.census.gov/history/www/census_then_now/notable_alumni/herman_hollerith.html

Gauthier, J. (n.d.). Univac I - History - U.S. Census Bureau. Retrieved February 12, 2021, from https://www.census.gov/history/www/innovations/technology/univac_i.html

Horowitz, P., & Hill, W. (1989). *The art of electronics*. Cambridge Univ. Press.

Unix vs Linux. (2021). Javatpoint. https://www.javatpoint.com/unix-vs-linux.

Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., … Wolff, S. (2021). *Brief History of the Internet*. Internet Society. https://www.internetsociety.org/internet/history-internet/brief-history-internet/#f5.

Navarri, G. (2020, August 20). How the Internet was Born: From the ARPANET to the Internet. The Conversation. https://theconversation.com/how-the-internet-was-born-from-the-arpanet-to-the-internet-68072.

Public Broadcasting Service. (1999). *Integrated Circuit*. PBS. https://www.pbs.org/transistor/background1/events/icinv.html.

Raymond, E. S. (2003). *The Art of Unix programming*. Addison-Wesley Professional.

Raymond, E. S. (2003). What Unix Gets Right. https://homepage.cs.uri.edu/~thenry/resources/unix_art/ch01s05.html.

Ritchie, D. M. (1993). The Development of the C Language. *ACM Sigplan Notices*, *28*(3), 1-16.

Roberts, L. G. (2001). The Evolution of Packet Switching. https://web.archive.org/web/20160324033133/http://www.packet.cc/files/ev-packet-sw.html.

Saint, C. (n.d.). *Integrated circuit*. *Encyclopedia Britannica*. https://www.britannica.com/technology/integrated-circuit

Schneider, G. P., Evans, J., & Pinard, K. (2009). *The Internet: Illustrated*. Cengage Learning.

Stonebank, M. (2000, October 9). *UNIX Introduction*. UNIX Tutorial - Introduction. http://cs.boisestate.edu/~mhthomas/cs221/references/UnixTutorial/UNIX_0.htm.

Stroustrup, B. (1996). A history of C++ 1979--1991. In *History of programming languages---II*, 699-769.

Stroustrup, B. (2018). *A Tour of C++* (2nd ed.). Addison-Wesley.

Stroustrup, B. (2020, October 27). C++ Applications. https://www.stroustrup.com/applications.html

The Computer History Museum. (2021). *1959: Practical Monolithic Integrated Circuit Concept Patented*. https://www.computerhistory.org/siliconengine/practical-monolithic-integrated-circuit-concept-patented/.

The Open Group. *(2021). The UNIX® Standard.* The UNIX® Standard | The Open Group Website. https://www.opengroup.org/membership/forums/platform/unix.

TIOBE Index for February 2021. (2020). https://www.tiobe.com/tiobe-index/.

Turing machines. (n.d.). https://brilliant.org/wiki/turing-machines/#:~:text=Turing%20machines%20provide%20a%20powerful,the%20advantage%20of%20unlimited%20memory

University of Pennsylvania. (2017). ENIAC at Penn Engineering. Penn Engineering. https://www.seas.upenn.edu/about/history-heritage/e

Vitanyi, P. M. (2009). Turing machine. http://www.scholarpedia.org/article/Turing_machine

Von Neumann Architecture. (2019, April 16). https://www.computerscience.gcse.guru/theory/von-neumann-architecture.

Watkins, T. (n.d.). The History of the Transistor. Retrieved February 13, 2021, from https://www.sjsu.edu/faculty/watkins/transist.htm#:~:text=The%20transistor%20was%20successfully%20demonstrated,John%20Bardeen%20and%20Walter%20Brattain

What is a Turing machine? (n.d.). https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/turing-machine/one.html