The Origins of Computing: Studying the Pioneers of Computer Science

By Jeff Kraus

With the ever-present push toward advancement in computing technologies, we find ourselves always looking forward to imagine the major development that will come next, the innovation that will shape the business, education, and everyday living of tomorrow. With this forward-looking mindset, we often fail to recognize the technological revolutions of the past that laid a foundation for every computing and processing device in existence. This examination of computers’ history is precisely the topic of our study in Dr. Perry Alexander’s Honors freshman seminar: The Origins of Computing.

When we first began our investigation into the beginnings of the very principles of computer science, Dr. Alexander raised awareness to the fact that, while we could all name the inventor of the cotton gin and the light bulb, nobody could answer the question, “Who invented the computer?” And so, we delved into the individuals who laid the groundwork for the machines we rely on every day. These people, who in their time were known as mathematicians, logicians, and even philosophers, never actually saw anything that resembled our modern computer during their lifetimes, yet their contributions were nevertheless crucial to the development of the very machine on which I type this article.

Beginning with the mathematical and logical concepts underlying computer science, we must look back to the 19th century and to German mathematician David Hilbert. Hilbert was a major proponent of the formalism movement, and he sought to provide a complete and consistent logical base for all mathematics. First, he wanted a complete, formalized system for the axiomatization of propositional logic and first-order predicate calculus. He also sought to axiomatize geometry by offering his own formal set of axioms to replace Euclid’s. In his famed twenty-three problems from the year 1900, Hilbert included one challenging the math world to prove the consistency of arithmetic’s axioms. All these formalized approaches to logic and math opened the door for the automated manipulation of the same mathematical systems, which is what we now call computing.

Through the early 20th century, an Austrian man named Kurt Gödel followed in Hilbert’s path by adding his own theorems to the formalization of mathematics. In response to Hilbert’s second problem, mentioned above, Gödel published what is now known as his “incompleteness theorems”. Basically, he proved that for any axiomatized system with enough power to describe arithmetic of natural numbers, the system cannot be both consistent and complete, and the axioms cannot be proven consistent within the system. For his proof, Gödel used contradiction to show that there is always a statement that is true but not provable within the theory. Later, Gödel traveled to the United States to share his work with the American Mathematical Society as well as in lectures at the Institute for Advanced Study, Princeton, and Notre Dame. While in the US he also befriended famed physicist Albert Einstein, and the two fed off each other’s ideas for many years. In fact, Einstein once said that the brightest part of his day was when he could walk home with Gödel, and the two would discuss their work with each other.

Gödel’s developments of the Incompleteness Theorem and universal formal languages paved the way for our next subject, Alan Turing, to actually develop a computational engine based on these mathematical and logical premises. Perhaps Turing’s most influential contribution was his description of the Turing machine as a hypothetical computing device that can manipulate symbols printed on a strip of tape and move left or right on the tape. He was able to show that his machine could perform any computation that was represented as an algorithm. Thus, we have the first conceptual computer processor and the program that it runs. Additionally, Turing used his machine to prove that the “halting problem” was unsolvable; no single algorithm can decide whether a Turing machine will ever halt or run indefinitely. Much of Turing’s other work was during World War II while acting as a British codebreaker of German messages. His biggest accomplishment here was in developing a machine to find the settings for the German Enigma machine, allowing the British to crack some of Germany’s most confidential correspondence. In fact, some of his papers about probability and cryptography were so highly valued by the British intelligence that they were not released to the public until the year 2012. Later in his career, Turing proposed what is now called the “Turing test” to determine the intelligence of a computer. This test comprised of carrying on a conversation with a computer to see if one could distinguish between human dialogue and the computer’s responses. All these accomplishments led to Turing’s title as the “father of computer science” and his name on the Turing award, the world’s highest honor in computer science.

These three men made giant strides on the path toward the computers we have now in laptops, in phones, in tablets, in cars, and in all kinds of systems that we may begin to take for granted. Our seminar is not finished examining the progress of computers, as many more people and factors have influenced the subject. Still, we know that the best way to envision and innovate the future of technology is to keep a firm understanding of the computer’s origin by perceiving and appreciating the necessary steps of mathematical and logical geniuses like David Hilbert, Kurt Gödel, and Alan Turing.