LAURA BARRET: Welcome back to Hindsight is 20/19, the podcast where we look at 250 years of Dartmouth’s history through 25 objects from the Library’s archival collections, one object per decade. I am Laura Barret and I will be your host. In this episode, I’ll be talking about the development of the computer programming language called BASIC. Before I do, take a moment and try to remember the first time you ever used a computer.

The first time I remember using a computer was in 1983. My parents bought a Commodore 64 and set it up on a giant drafting table my dad had built in our family room. The computer was beige and boxy. I was 10 years old, and I primarily used it and our dot matrix printer to create giant banners to hang around the house.

About a week ago, I wandered around Baker-Berry Library asking people to tell me about their first memories using computers.

LB: Can you please tell me your name?

All: James Adams, Yenny Dieguez, Amy Bucci, Erin DeSilva, Wendel Cox, Todd Minsk, Joe Montibello, Richard Nadler.

LB: “Can you tell me about the first time you remember using a computer.”

JAMES ADAMS: “My uncle had a computer in his basement and my cousin and I going on AOL when I was very young…going on to chat rooms and that was all very exciting.”

YENNY DIEGUEZ: “I think I was like 5 or 6 years old and my parents had just moved into a new home. There was a computer on the floor and I was playing solitaire and the other one that was a machine. I was playing a game and that was the first time I used a computer.”

AMY BUCCI: “I actually took a class when I was in the 8th grade and it was a little mini Macintosh. We used mac paint.”

ERIN DESILVA: “I was at my cousins who had an Apple 2E and there was a game involved.”

WENDEL COX: “I played a computer game.”

TODD MINSK: “Well I was in high school. Hanover High school in Hanover, New Hampshire. The high school had some arrangements with Dartmouth to have a terminal in the school that was connected the College’s computer apparatus. I remember being asked, because I was already the information specialist of the school, I remember being asked for Boston Bruins schedule which in those days was a printed pamphlet. Somehow, I found one and because people wanted multiple copies, photocopy machines were not really available in those days, so I got idea to type the contents into a computer listing which these days would be considered ordinary word processing and got castigated for using scarce computer time for such a mundane purpose.”

JOE MONTIBELLO: “I also remember having a book on how to program, actually using the BASIC language. Following sort of recipes out of that book to make tiny little games or other things that a computer could do and then saving them to an audio recording tape or a cassette tape for those people who remember that technology.”

RICHARD NADLER: “I bought the computer and wanted to see what I could do with it. In those days if you wanted to do anything with a computer, you needed to learn to program in BASIC.”

LB: “Do you mind telling us about what year that was?”

JA: “94 or 93 maybe.”

YD: “2002 or 2003.”

AB: “It was 1986.”

ED: “Probably about 1989.”

WC: “1979.”

TM: “1960s. Probably about 1968-69.”

JM: “Maybe 1983 or 1984.”

RN: “I’m going to guess it was around 1969.”

LB: Back in 1964—years, or in most cases even decades, before any of the people you just heard from first used a computer—two Dartmouth mathematics professors produced the first version of the programming language Dartmouth BASIC. The professors, John Kemeny and Thomas Kurtz, were committed to making their programming language accessible to everyone—even people with little or no familiarity with computers or computer programming. One of their aspirations was for all Dartmouth students, including all humanities majors, to learn to program in BASIC. While BASIC was based on earlier programming languages, Kemeny and Kurtz used familiar English words in all the language’s statements. Today, we’d say they were committed to creating a *user-friendly* programming language, but the phrase “user-friendly” wasn’t in use until about 20 years after BASIC was developed.

This user-friendly approach is the focus of today’s episode. I won’t be delving into the technical details of the creation of BASIC. I’m interested in how its creators strove to make BASIC accessible to a variety of potential programmers. This leads us right into our item, or rather *items*, for today: the BASIC user manual. The first user manual was created in 1964. In preparing for this episode, I read through the opening pages to this manual and five subsequent manuals published over the next 19 years. The way Kurtz and Kemeny try to connect with readers in the first page or two of each manual gives us insight to their goals and reminds us just how new and unfamiliar computers and computer programming were to most people at the time.

Let’s start with that first manual, published in 1964, the year BASIC was born. This manual provides a brief preface in which the authors lay out four requirements they had for the language. One is that the language be “very easy to learn,” two, that the communication between the programming language and the language of the computer is fast, three, that BASIC serve as a “stepping stone” for students who want to go on to learn one of the standard programming languages such as FORTRAN, and, lastly, that it be a “general purpose” language useful for every kind of machine computation.” Next, we learn that they stayed as close to “ordinary English” as possible; as proof, they present a few lines of programming in the form of a mathematical equation that, as I can attest, are easily understood by a novice.

The second edition of the BASIC user manual was published the following year, in June 1965. It’s clear that during that year, someone realized that the user-*friendliness* of BASIC needed to be applied to its user-*manual*, too. This time around, not only is BASIC described as “precise, simple, and easy to understand,” but the start of the manual itself is a little easier to understand, too. On page one, readers are greeted with the question, “What is a program?” I’ll read to you directly from the manual to answer that question:

“A program is a set of directions, or a recipe, that is used to tell a computer how to provide an answer to some problem. It usually starts with the given data as the ingredients, contains a set of instructions to be performed or carried out in a certain order, and ends up with a set of answers as the cake. And, as with ordinary cakes, if you make a mistake in your program, you will end up with something else—perhaps hash!”

So, writing a computer program is kind of like baking a cake? All right, I’m with you. Honestly, they lost me a bit at the hash part--I’ve baked a lot of cakes and made my share of mistakes, but I’ve never ended up with hash. Still, the analogy is a good one.

Farther down on the page, the authors explain the precision needed in a programming language, again, in a fairly understandable way: they write “It is easy for a programmer to present a program in the English language, but such a program poses great difficulties for the computer because English is rich with ambiguities and redundancies, those qualities which make poetry possible but computing impossible.”

In 1968, an updated user manual was published. In the introduction the authors stuck with the cake analogy--it must have been working!

In another edition, published in 1971, the introduction is lengthened for the edification of novice users. While the section titled, “What is a Program?” remains, it now appears sandwiched in between two other sections. It’s preceded by, “What is a computer?” and followed by, “What is BASIC?”

A computer, we are told, “ is a very simple and at the same time a very complex machine,” that, on the one hand, merely follows explicit directions while, on the other hand, performs its work so rapidly that an array of other devices are needed to correlate all the tasks the computer is performing.

The section addressing the question, “What is a program?” still contains the cake analogy, with some modifications, and also includes a new piece that seems to be trying to quell users’ intellectual anxieties. First, the authors explain that, “Any mistakes in a program render it just about useless,” and that there are two forms of errors: errors of grammar and substantive errors. They explain, “In the case of recipes for baking cakes, misspelling and typographical errors are examples of errors of form; some of these may make the recipe unreadable. An example of a substantive error would be a direction to use baking soda instead of baking powder.”

And now we get to their kind reassurance that programmers, even novice programmers who are nervous about making mistakes, are still more intelligent than mere machines. They write

“Since a computer has much less intelligence or common sense than a human being, programs for it must adhere strictly to rules of form or grammar. ….This language is called machine language, and its difficult nature has led computer specialists to invent other more easily used language that can be converted or translated to machine language.”

In the following section, the authors analogize computing with keeping an account, such as a checkbook. Again, everyday examples to make everyday people feel at ease trying their hand at computer programming.

The last two manuals I looked at were published in the following decade, in 1981 and 1983. While these manuals assert that BASIC is geared to “meet the needs of beginners and casual users,” and give an easy-to-understand description of computer programs, there’s not as much of the hand-holding as we saw in most of the previous editions. Analogies to baking cakes and balancing your checkbook are nowhere to be seen, and the authors no longer felt the need to tell us exactly what a computer is.

Remember back those interviews at the start of this episode? The early 1980s is when a number of people, myself included, first remember using a computer, and most of those memories took place in someone’s house. Computers were no longer accessible only in government facilities, in specialized labs, or on some college campuses. Computers were entering homes. They weren’t only being used to process complex mathematical equations, they also were used for gaming, word processing, and even a bit of graphic design. Today, computers are almost everywhere, and we often “use” them without even realizing it. From smartphones and watches that track our steps and map our locations, to baby monitors and children’s toys, to smoke detectors, light switches, and even toasters. I opened by asking you to remember the first time you used a computer which, at least for those of us over 25 years old, is a distinct memory rooted in a specific time and place. If we now try to think about the last time we did *not* use a computer, it’s a lot harder than it sounds. In fact, Kurtz and Kemeny’s analogies might seem a bit quaint. We use computers way more often than we bake cakes or balance checkbooks….in fact, these days, it would be a challenge to do either of those things without the help of a computer.

Hindsight is 20/19 is a production of the Dartmouth College Library and is produced as part of the celebration of Dartmouth’s 250th anniversary, Highlighting selected objects from Rauner Special Collections Library. This episode was written and directed by Laura Barret, produced by Julia Logan. Our sound engineer was Peter Carini. I want to give special thanks to everyone I interviewed for this episode. Thank you for listening and we hope you will continue to enjoy *Hindsight* *is 20/19*.

PETER CARINI: Hi. This is Peter Carini. The sound engineer for this episode. The following songs were used in the episodes and sourced from the free music archive. “Surfing Day” by Marcos H Belanos. ‘Living in a Dream” by Twin Guns. And “Deluge” by Cellophane Sam.