About COBOL
CS399 – COBOL and Legacy Systems
Spring 2021

Q: What's a "COBOL?"

A: COBOL is a programming language that has been in constant use in large enterprise systems from the 1960s thru the present. It stands for "Common Business Oriented Langage."

Q: What is it used for?

A: COBOL programs tend to be used in government and financial systems. A study by Thomson-Reuters [http://fingfx.thomsonreuters.com/gfx/rngs/USA-BANKS-COBOL/010040KH18J/index.html] found 43% of banking systems are still based on COBOL and 95% of ATM swipes are processed by COBOL programs. This helps account for the over 200 billion lines of COBOL still in use. Mellon Bank in New York alone has 343 million lines of COBOL making up over 112,000 separate programs [https://www.computerworld.com/article/2504568/the-cobol-brain-drain.html] and the Social Security Administration has over 60 million lines of COBOL code [https://www.gao.gov/assets/670/663057.pdf].

Q: Why haven't I heard about COBOL before this?

A: Over the past 30 years, initiatives to replace COBOL applications with more "modern" languages have come and gone, and with each one we thought no one would ever need COBOL programmers again. So eventually everyone quit teaching it. But we were wrong. Many of these initiatives crashed and burned, and even the "successful" ones ended up spending millions, if not billions of dollars only to end up with a "new" application that did exactly the same thing as the old application – except with many of the same bugs that were worked out of the legacy application over its 50-year lifespan. One popular example of this was a 2012 rewrite of a COBOL payroll system by the Australian Health Agency in Queensland. The rewrite cost $6 Million dollars, but it failed upon rollout and ultimately cost the Queensland Government an estimated $1.2 Billion [https://www.tmaxsoft.com/why-rewriting-legacy-applications-can-be-a-costly-mistake-and-how-to-avoid-it/]. After which, they ended up with an application that did exactly the same thing the legacy system did.

Q: Doesn't COBOL only run on old, obsolete computer systems?

A: Not at all! While it is true that many of the mini/mainframe vendors of the past, like UNIVAC, NCR, DEC, Prime and Amdahl have gone out of business or refocused their business, IBM actively supports COBOL on their modern Z family of computers running z/OS, the most recent of which was released in 2019. In addition, there are a variety of implementations for other environments such as .NET, Macs and Linux. GnuCOBOL is an open source implementation that conforms to the 1985 COBOL standard and runs on Windows, Macs and Linux.

Q: Well, what about the language? It was designed in the 1960s. That must mean it's obsolete.

A: While the language was originally designed in the 1960s, the first standard was published in 1968, with major updates in 1974 and 1985. For comparison, BASIC was invented in 1964, and in 2018, it (Visual BASIC) was ranked as the fifth most popular programming language in the world. C, the language used in most *NIX systems programming? It was in invented in 1972 (the same year UNIX was invented) by Ken Thompson and Dennis Ritchie. And Bjane Stroustrup created C++ in 1982.

Q; What's this crisis involving COBOL systems keep hearing about?

A: Because universities and colleges haven't been teaching COBOL for the past twenty or thirty years, organizations haven't been able to find replacements as their programmers retire. Barely 10% of today's COBOL programmers are under 35. In fact, over 60% are 50 years of age or older, and we're starting to see waves of retirements with no one to take their places. One study estimates that the "average" COBOL programmer is 60 years old [https://www.dallasnews.com/business/technology/2020/04/13/if-youre-an-expert-in-cobol-youre-suddenly-in-high-demand/]. A recent search on Glassdoor for COBOL jobs returned over 900 opportunities across the country as well as fully remote. Indeed, if you've been following the news, you may recall hearing about New Jersey's COBOL programmer crunch [https://www.tomshardware.com/news/new-jersey-cobol-coders-mainframes-coronavirus].

Q: Can you tell me some interesting facts about COBOL?

A1: COBOL was designed to be very English-like. For example, 'Profit=Sales-Cost' in other languages would look like 'SUBTRACT Cost FROM Sales GIVING Profit' in COBOL.

A2: COBOL has a first class data type (COMPUTATIONAL-3) that can represent a number of up to 31 digits. That's 9,999,999,999,999,999,999,999,999,999,999. This is possible because COMPUTATIONAL-3 represents numeric values using 4 bit decimal digits. Modern IBM z

systems support COMPUTATIONAL-3 arithmetic in firmware, making computations very efficient.

A3. The COBOL Report Writer feature allows you to write a sophisticated report with headings, subtotals, totals, footers, counters, etc. by simply describing what he output is supposed to look like and using this code pattern:

```
OPEN INPUT STUDENT-FILE.
OPEN OUTPUT REPORT-FILE.
INITIATE STUDENT-REPORT.
READ STUDENT-FILE AT END MOVE 1 to FILE-STATUS.
PERFORM GENERATE-REPORT UNTIL EndOfFile.
TERMINATE STUDENT-REPORT.
CLOSE STUDENT-FILE.
CLOSE REPORT-FILE.
STOP RUN.
```

Q; I'm sold! How do I learn more about this Amazing Language?

A: Sign up for CS399 this Spring: CS 399 SPST: COBOL AND LEGACY SYSTEMS, CRN 64667. Monday and Wednesdays at 2PM.