= COBOL =

COBOL (/ ?ko?b?l / , an acronym for common business @-@ oriented language) is a compiled English @-@ like computer programming language designed for business use . It is imperative , procedural and , since 2002 , object @-@ oriented . COBOL is primarily used in business , finance , and administrative systems for companies and governments . COBOL is still widely used in legacy applications deployed on mainframe computers , such as large @-@ scale batch and transaction processing jobs . But due to its declining popularity and the retirement of experienced COBOL programmers , programs are being migrated to new platforms , rewritten in modern languages or replaced with software packages . Most programming in COBOL is now purely to maintain existing applications .

COBOL was designed in 1959 , by CODASYL and was partly based on previous programming language design work by Grace Hopper , commonly referred to as " the (grand) mother of COBOL " . It was created as part of a US Department of Defense effort to create a portable programming language for data processing . Intended as a stopgap , the Department of Defense promptly forced computer manufacturers to provide it , resulting in its widespread adoption . It was standardized in 1968 and has since been revised four times . Expansions include support for structured and object @-@ oriented programming . The current standard is ISO / IEC 1989 : 2014 .

COBOL has an English @-@ like syntax , which was designed to be self @-@ documenting and highly readable . However , it is verbose and uses over 300 reserved words . In contrast with modern , succinct syntax like y = x ; , COBOL has a more English @-@ like syntax (in this case , MOVE x TO y) . COBOL code is split into four divisions (identification , environment , data and procedure) containing a rigid hierarchy of sections , paragraphs and sentences . Lacking a large standard library , the standard specifies 43 statements , 87 functions and just one class .

Academic computer scientists were generally uninterested in business applications when COBOL was created and were not involved in its design ; it was (effectively) designed from the ground up as a computer language for businessmen , with an emphasis on inputs and outputs , whose only data types were numbers and strings of text . COBOL has been criticized throughout its life , however , for its verbosity , design process and poor support for structured programming , which resulted in monolithic and incomprehensible programs .

= = History and specification = =

= = = Background = = =

In the late 1950s , computer users and manufacturers were becoming concerned about the rising cost of programming . A 1959 survey had found that in any data processing installation , the programming cost US $ 800 @,@ 000 on average and that translating programs to run on new hardware would cost $ 600 @,@ 000 . At a time when new programming languages were proliferating at an ever increasing rate , the same survey suggested that if a common business @-@ oriented language were used , conversion would be far cheaper and faster .

In April 1959 , representatives from academia , computer users and manufacturers met at the University of Pennsylvania to organize a formal meeting on common business languages . Representatives among others , included Grace Hopper , inventor of the English @-@ like data processing language FLOW @-@ MATIC , Jean Sammet and Saul Gorn .

The group asked the Department of Defense (DoD) to sponsor an effort to create a common business language . The delegation impressed Charles A. Phillips , director of the Data System Research Staff at the DoD , who thought that they " thoroughly understood " the DoD 's problems . The DoD operated 225 computers , had a further 175 on order and had spent over $ 200 million on implementing programs to run on them . Portable programs would save time , reduce costs and ease modernization .

Phillips agreed to sponsor the meeting and tasked the delegation with drafting the agenda .

= = = COBOL 60 = = =

On May 28 and 29 of 1959 (exactly one year after the Zürich ALGOL 58 meeting) , a meeting was held at the Pentagon to discuss the creation of a common programming language for business . It was attended by 41 people and was chaired by Phillips . The Department of Defense was concerned about whether it could run the same data processing programs on different computers . FORTRAN , the only mainstream language at the time , lacked the features needed to write such programs .

Representatives enthusiastically described a language that could work in a wide variety of environments , from banking and insurance to utilities and inventory control . They agreed unanimously that more people should be able to program and that the new language should not be restricted by the limitations of contemporary technology . A majority agreed that the language should make maximal use of English , be capable of change , be machine @-@ independent and be easy to use , even at the expense of power .

The meeting resulted in the creation of a steering committee and short- , intermediate- and long @-@ range committees . The short @-@ range committee was given to September (three months) to produce specifications for an interim language , which would then be improved upon by the other committees . Their official mission , however , was to identify the strengths and weaknesses of existing programming languages and did not explicitly direct them to create a new language . The deadline was met with disbelief by the short @-@ range committee . One member , Betty Holberton , described the three @-@ month deadline as " gross optimism " and doubted that the language really would be a stopgap .

The steering committee met on June 4 and agreed to name the entire activity as the Committee on Data Systems Languages , or CODASYL , and to form an executive committee .

The short @-@ range committee was made up of members representing six computer manufacturers and three government agencies . The six computer manufacturers were Burroughs Corporation , IBM , Minneapolis @-@ Honeywell (Honeywell Labs) , RCA , Sperry Rand , and Sylvania Electric Products . The three government agencies were the US Air Force , the Navy 's David Taylor Model Basin , and the National Bureau of Standards (now the National Institute of Standards and Technology) . The committee was chaired by Joseph Wegstein of the US National Bureau of Standards . Work began by investigating data description , statements , existing applications and user experiences .

The committee mainly examined the FLOW @-@ MATIC , AIMACO and COMTRAN programming languages . The FLOW @-@ MATIC language was particularly influential because it had been implemented and because AIMACO was a derivative of it with only minor changes . FLOW @-@ MATIC 's inventor , Grace Hopper , also served as a technical adviser to the committee . FLOW @-@ MATIC 's major contributions to COBOL were long variable names , English words for commands and the separation of data descriptions and instructions .

IBM 's COMTRAN language , invented by Bob Bemer , was regarded as a competitor to FLOW @-@ MATIC by a short @-@ range committee made up of colleagues of Grace Hopper . Some of its features were not incorporated into COBOL so that it would not look like IBM had dominated the design process , and Jean Sammet said in 1981 that there had been a " strong anti @-@ IBM bias " from some committee members (herself included) . In one case , after Roy Goldfinger , author of the COMTRAN manual and intermediate @-@ range committee member , attended a subcommittee meeting to support his language and encourage the use of algebraic expressions , Grace Hopper sent a memo to the short @-@ range committee reiterating Sperry Rand 's efforts to create a language based on English . In 1980 , Grace Hopper commented that " COBOL 60 is 95 % FLOW @-@ MATIC " and that COMTRAN had had an " extremely small " influence . Furthermore , she said that she would claim that work was influenced by both FLOW @-@ MATIC and COMTRAN only to " keep other people happy [so they] wouldn 't try to knock us out " . Features from COMTRAN incorporated into COBOL included formulas , the PICTURE clause , an improved IF statement , which obviated the need for GO TOs , and a more robust file management system .

The usefulness of the committee 's work was subject of great debate . While some members thought the language had too many compromises and was the result of design by committee , others felt it was better than the three languages examined . Some felt the language was too complex ; others , too simple . Controversial features included those some considered useless or too advanced for data processing users . Such features included boolean expressions , formulas and table subscripts (indices) . Another point of controversy was whether to make keywords context @-@ sensitive and the effect that would have on readability . Although context @-@ sensitive keywords were rejected , the approach was later used in PL / I and partially in COBOL from 2002 . Little consideration was given to interactivity , interaction with operating systems (few existed at that time) and functions (thought of as purely mathematical and of no use in data processing) .

The specifications were presented to the Executive Committee on September 4 . They fell short of expectations : Joseph Wegstein noted that " it contains rough spots and requires some additions " , and Bob Bemer later described them as a " hodgepodge " . The subcommittee was given until December to improve it .

At a mid @-@ September meeting , the committee discussed the new language 's name . Suggestions included " BUSY " (Business System) , " INFOSYL " (Information System Language) and " COCOSYL " (Common Computer Systems Language) . The name " COBOL " was suggested by Bob Bemer .

In October , the intermediate @-@ range committee received copies of the FACT language specification created by Roy Nutt . Its features impressed the committee so much that they passed a resolution to base COBOL on it . This was a blow to the short @-@ range committee , who had made good progress on the specification . Despite being technically superior , FACT had not been created with portability in mind or through manufacturer and user consensus . It also lacked a demonstrable implementation , allowing supporters of a FLOW @-@ MATIC @-@ based COBOL to overturn the resolution . RCA representative Howard Bromberg also blocked FACT , so that RCA 's work on a COBOL implementation would not go to waste .

It soon became apparent that the committee was too large for any further progress to be made quickly . A frustrated Howard Bromberg bought a $ 15 tombstone with " COBOL " engraved on it and sent it to Charles Phillips to demonstrate his displeasure . A sub @-@ committee was formed to analyze existing languages and was made up of six individuals :

William Selden and Gertrude Tierney of IBM ,

Howard Bromberg and Howard Discount of RCA ,

Vernon Reeves and Jean E. Sammet of Sylvania Electric Products .

The sub @-@ committee did most of the work creating the specification , leaving the short @-@ range committee to review and modify their work before producing the finished specification .

The specifications were approved by the Executive Committee on January 3 , 1960 , and sent to the government printing office , which printed these as COBOL 60 . The language 's stated objectives were to allow efficient , portable programs to be easily written , to allow users to move to new systems with minimal effort and cost , and to be suitable for inexperienced programmers . The CODASYL Executive Committee later created the COBOL Maintenance Committee to answer questions from users and vendors and to improve and expand the specifications .

During 1960 , the list of manufacturers planning to build COBOL compilers grew . By September , five more manufacturers had joined CODASYL (Bendix , Control Data Corporation , General Electric (GE) , National Cash Register and Philco) , and all represented manufacturers had announced COBOL compilers . GE and IBM planned to integrate COBOL into their own languages , GECOM and COMTRAN , respectively . In contrast , International Computers and Tabulators planned to replace their language , CODEL , with COBOL .

Meanwhile , RCA and Sperry Rand worked on creating COBOL compilers . The first COBOL program ran on 17 August on an RCA 501 . On December 6 and 7 , the same COBOL program (albeit with minor changes) ran on an RCA computer and a Remington @-@ Rand Univac computer , demonstrating that compatibility could be achieved .

The relative influences of which languages were used continues to this day in the recommended advisory printed in all COBOL reference manuals :

COBOL is an industry language and is not the property of any company or group of companies , or of any organization or group of organizations .

No warranty , expressed or implied , is made by any contributor or by the CODASYL COBOL Committee as to the accuracy and functioning of the programming system and language . Moreover , no responsibility is assumed by any contributor , or by the committee , in connection therewith . The authors and copyright holders of the copyrighted material used herein are as follows :

FLOW @-@ MATIC (trademark of Unisys Corporation) , Programming for the UNIVAC (R) I and II , Data Automation Systems , copyrighted 1958 , 1959 , by Unisys Corporation ; IBM Commercial Translator Form No . F28 @-@ 8013 , copyrighted 1959 by IBM ; FACT , DSI 27A5260 @-@ 2760 , copyrighted 1960 by Minneapolis @-@ Honeywell .

They have specifically authorized the use of this material , in whole or in part , in the COBOL specifications . Such authorization extends to the reproduction and use of COBOL specifications in programming manuals or similar publications .

= = = COBOL @-@ 61 to COBOL @-@ 65 = = =

Many logical flaws were found in COBOL 60 , leading GE 's Charles Katz to warn that it could not be interpreted unambiguously . A reluctant short @-@ term committee enacted a total cleanup and , by March 1963 , it was reported that COBOL 's syntax was as definable as ALGOL 's , although semantic ambiguities remained .

Early COBOL compilers were primitive and slow . A 1962 US Navy evaluation found compilation speeds of 3 ? 11 statements per minute . By mid @-@ 1964 , they had increased to 11 ? 1000 statements per minute . It was observed that increasing memory would drastically increase speed and that compilation costs varied wildly : costs per statement were between $ 0 @.@ 23 and $ 18 @.@ 91 .

In late 1962 , IBM announced that COBOL would be their primary development language and that development of COMTRAN would cease .

The COBOL specification was revised three times in the five years after its publication . COBOL @-@ 60 was replaced in 1961 by COBOL @-@ 61 . This was then replaced by the COBOL @-@ 61 Extended specifications in 1963 , which introduced the sort and report writer facilities . The added facilities corrected flaws identified by Honeywell in late 1959 in a letter to the short @-@ range committee . COBOL Edition 1965 brought further clarifications to the specifications and introduced facilities for handling mass storage files and tables .

= = = COBOL @-@ 68 = = =

Efforts began to standardize COBOL to overcome incompatibilities between versions . In late 1962 , both ISO and the United States of America Standards Institute (now ANSI) formed groups to create standards . ANSI produced USA Standard COBOL X3.23 in August 1968 , which became the cornerstone for later versions . This version was known as American National Standard (ANS) COBOL and was adopted by ISO in 1972 .

= = = COBOL @-@ 74 = = =

By 1970 , COBOL had become the most widely used programming language in the world .

Independently of the ANSI committee , the CODASYL Programming Language Committee was working on improving the language . They described new versions in 1968 , 1969 , 1970 and 1973 , including changes such as new inter @-@ program communication , debugging and file merging facilities as well as improved string @-@ handling and library inclusion features . Although CODASYL was independent of the ANSI committee , the CODASYL Journal of Development was used by ANSI to identify features that were popular enough to warrant implementing . The Programming Language Committee also liaised with ECMA and the Japanese COBOL Standard committee .

The Programming Language Committee was not well @-@ known , however . The vice @-@ president , William Rinehuls , complained that two @-@ thirds of the COBOL community did not know of the committee 's existence . It was also poor , lacking the funds to make public documents , such as minutes of meetings and change proposals , freely available .

In 1974 , ANSI published a revised version of (ANS) COBOL , containing new features such as file organizations , the DELETE statement and the segmentation module . Deleted features included the NOTE statement , the EXAMINE statement (which was replaced by INSPECT) and the implementer @-@ defined random access module (which was superseded by the new sequential and relative I / O modules) . These made up 44 changes , which rendered existing statements incompatible with the new standard . The report writer was slated to be removed from COBOL , but was reinstated before the standard was published . ISO later adopted the updated standard in 1978 .

= = = COBOL @-@ 85 = = =

In June 1978 , work began on revising COBOL @-@ 74 . The proposed standard (commonly called COBOL @-@ 80) differed significantly from the previous one , causing concerns about incompatibility and conversion costs . In January 1981 , Joseph T. Brophy , Senior Vice @-@ President of Travelers Insurance , threatened to sue the standard committee because it was not upwards compatible with COBOL @-@ 74 . Mr. Brophy described previous conversions of their 40 @-@ million @-@ line code base as " non @-@ productive " and a " complete waste of our programmer resources " . Later that year , the Data Processing Management Association (DPMA) said it was " strongly opposed " to the new standard , citing " prohibitive " conversion costs and enhancements that were " forced on the user " .

During the first public review period , the committee received 2 @,@ 200 responses , of which 1 @,@ 700 were negative form letters . Other responses were detailed analyses of the effect COBOL @-@ 80 would have on their systems ; conversion costs were predicted to be at least 50 cents per line of code . Fewer than a dozen of the responses were in favor of the proposed standard .

In 1983 , the DPMA withdrew its opposition to the standard , citing the responsiveness of the committee to public concerns . In the same year , a National Bureau of Standards study concluded that the proposed standard would present few problems . A year later , a COBOL @-@ 80 compiler was released to DEC VAX users , who noted that conversion of COBOL @-@ 74 programs posed few problems . The new EVALUATE statement and inline PERFORM were particularly well received and improved productivity , thanks to simplified control flow and debugging .

The second public review drew another 1 @,@ 000 (mainly negative) responses , while the last drew just 25 , by which time many concerns had been addressed .

In late 1985 , ANSI published the revised standard . Sixty features were changed or deprecated and many were added , such as :

Scope terminators (END @-@ IF , END @-@ PERFORM , END @-@ READ , etc .)

Nested subprograms

CONTINUE , a no @-@ operation statement

EVALUATE , a switch statement

INITIALIZE , a statement that can set groups of data to their default values

Inline PERFORM loop bodies ? previously , loop bodies had to be specified in a separate procedure

Reference modification , which allows access to substrings

I / O status codes .

The standard was adopted by ISO the same year . Two amendments followed in 1989 and 1993 , the first introducing intrinsic functions and the other providing corrections . ISO adopted the amendments in 1991 and 1994 respectively , before subsequently taking primary ownership and development of the standard .

= = = COBOL 2002 and object @-@ oriented COBOL = = =

In 1997 , Gartner Group estimated that there were a total of 200 billion lines of COBOL in existence , which ran 80 % of all business programs .

In the early 1990s , work began on adding object @-@ orientation in the next full revision of COBOL . Object @-@ oriented features were taken from C + + and Smalltalk . The initial estimate was to have this revision completed by 1997 , and an ISO Committee Draft (CD) was available by 1997 . Some vendors (including Micro Focus , Fujitsu , and IBM) introduced object @-@ oriented syntax based on drafts of the full revision . The final approved ISO standard was approved and published in late 2002 .

Fujitsu / GTSoftware , Micro Focus and RainCode introduced object @-@ oriented COBOL compilers targeting the .NET Framework .

There were many other new features , many of which had been in the CODASYL COBOL Journal of Development since 1978 and had missed the opportunity to be included in COBOL @-@ 85 . These other features included :

Free @-@ form code

User @-@ defined functions

Recursion

Locale @-@ based processing

Support for extended character sets such as Unicode

Floating @-@ point and binary data types (until then , binary items were truncated based on their declaration 's base @-@ 10 specification)

Portable arithmetic results

Bit and boolean data types

Pointers and syntax for getting and freeing storage

The SCREEN SECTION for text @-@ based user interfaces

The VALIDATE facility

Improved interoperability with other programming languages and framework environments such as .NET and Java .

Three corrigenda were published for the standard : two in 2006 and one in 2009 .

= = = COBOL 2014 = = =

Between 2003 and 2009 , three technical reports were produced describing object finalization , XML processing and collection classes for COBOL .

COBOL 2002 suffered from poor support : no compilers completely supported the standard . Micro Focus found that it was due to a lack of user demand for the new features and due to the abolition of the NIST test suite , which had been used to test compiler conformance . The standardization process was also found to be slow and under @-@ resourced .

COBOL 2014 includes the following changes :

Portable arithmetic results have been replaced by IEEE 754 data types

Major features have been made optional , such as the VALIDATE facility , the report writer and the screen @-@ handling facility .

Method overloading

Dynamic capacity tables (a feature dropped from the draft of COBOL 2002)

= = = Legacy = = =

COBOL programs are used globally in governments and businesses and are running on diverse operating systems such as z / OS , VME , Unix , OpenVMS and Windows . In 1997 , the Gartner Group reported that 80 % of the world 's business ran on COBOL with over 200 billion lines of code and 5 billion lines more being written annually .

Near the end of the 20th century , the year 2000 problem (Y2K) was the focus of significant

COBOL programming effort , sometimes by the same programmers who had designed the systems decades before . The particular level of effort required to correct COBOL code has been attributed to the large amount of business @-@ oriented COBOL , as business applications use dates heavily , and to fixed @-@ length data fields . After the clean @-@ up effort put into these programs for Y2K , a 2003 survey found that many remained in use . The authors said that the survey data suggest " a gradual decline in the importance of Cobol in application development over the [following] 10 years unless ... integration with other languages and technologies can be adopted " .

In 2006 and 2012 , Computerworld surveys found that over 60 % of organizations used COBOL (more than C + + and Visual Basic .NET) and that for half of those , COBOL was used for the majority of their internal software . 36 % of managers said they planned to migrate from COBOL , and 25 % said they would like to if it was cheaper . Instead , some businesses have migrated their systems from expensive mainframes to cheaper , more modern systems , while maintaining their COBOL programs .

= = Features = =

= = = Syntax = = =