

# [Common business oriented language](https://assignbuster.com/common-business-oriented-language/)

COBOL (Common Business Oriented Language) was the first widely used high-level
programming language for business applications. Many payroll, accounting, and
other business application programs written in COBOL over the past 35 years are
still possible that there are more existing lines of programming code in COBOL
than in any other programming language has been update over the years. Today we
already stepped in internet Age, most of old style business also have been
combined with intent to create the e-business, so we suppose COBOL is not useful
anymore but before we made a conclusion wed better know hows COBOL worked,
and hows COBOL will work in future then we could make decision. COBOL was an
effort to make a programming language that was like natural English, easy to
write and easier to read the coed after youd written it, and COBOL is one of
the oldest, and arguably the most successful and popular of all programming
languages. The earliest version of the language, COBOL-60 and -61, evolved to
the COBOL-85 standard sponsored by the Conference on Data Systems Language (CODASYL).

COBOL has been declared dead so many times since April 1968 till now, but COBOL
lives on. Nevertheless, the somber pronouncements of COBOLs demise continue,
and the pace has picked up with such developments as clients-sever technology,
Visual Basic, Java, and the chaos associated with the Year-2000 problems. Since
the year 2000 (Y2K) problem is common in many business applications and most of
these are written in COBOL, programmers with COBOL skills have become sought
after by major corporations and contractors. A number of companies have updated
COBOL and sell development tools to meet the requirements about COBOL
applications using in e-business. Since the COBOL use for Oriented of business
it was mostly serviced for big company. Even now many large companies have a
huge pool of COBOL-based applications that constitute their core business
systems, even in todays e-business. (For instances, The Seagram Co. Ltd., The
Federal Express, and Canada Trust. Etc.) If COBOL declare to death the
alternative is too awful to contemplate, the number of lines of COBOL
application code ranges from 200 billion to 5 trillion. Rewriting application in
Java may give purists a feeling of euphoria, but in reality no one is doing
this, or if you feel really masochistic, you could try and rewrite them in C++.

Every big company has to start from where it is. This is means taking old,
arthritic applications and wrapping them so they become a stable starting
point for future developments. These applications also need to be integrated,
and e-business means that these systems need to be integrated and connected to
the outside world. As a matter of fact, this is the most pressing issue facing
many large companies, forget about Java and XML, many IT manger would be happy
if their applications could talk to each other and provide an interface to the
outside world. Several computer companies already developed some economical,
practical product could combine with COBOL to working in e-business. At August
14, 1995 Micro Focus announced Visual Object COBOL version 1. 0 for window
95, in 1998 CASEGN system Ltd announces that COBOL for windows is available
free-of-charge to promote the use of COBOL as a powerful windows95/NT graphical
development tool, now IBM introduced VisualAge COBOL 2000 enterprise for OS/2
and Windows NT, and other COBOL supporters like CORBRA, MERANT, etc. those
supporters software are enhanced following features: a working station
development environment that enables you to seamlessly develop, modify,
reengineer, maintain, modernize, and port host applications right at your
desktop. As far as I am concerned, there is no good reason not use one of the
contemporary versions of COBOL for developing the server end of e-business
applications, and in reality many organizations will. After all, Merant is
growing at around 40 percent a year, so quite a few companies must be using its
products. As is usually the case in this industry, someone will spend years
cobbling together a Java-based e0business, applying copious amounts of tape and
string, and by the skin of their teeth get something that performs a useful
business function; and we will all marvel at it. The fact that it could have
been written in COBOL in a tenth of the time will be totally overlooked: after
all, wee must celebrate how rapidly we are advancing in this industry. We might
not buy so much otherwise. So its not surprising that they would have a
favorable. After I do this research the following statement is a useful advice
indeed. What the legacy argument does not address is the substantial amount of
new code being written today in small organizations organizations that never
have had, and never will have, a mainframe. Many of these organizations will
work hard to avoid writing any of the applications that COBOL has been used for,
because its much more economical to buy a package to do the job. But if a
fledgling 10-person company with one programmer (or a modest 100-person company
with 5-10 programmers) does decide to write an application system today for its
PC-based. Internet-aware environment, my guess is that the odds of COBOL being
used are about one in a thousand. So if you are a large company and you want to
move forward from where you are without taking huge risks, COBOL is worth at
least a look.