DeepSoft, Inc. Engineering • Programming • Training
6259 Deep River Canyon, Columbia, MD 21045-2572 USA
Phone: 410-312-2982 Email: tkf@deepsoftinc.com
Fax: 410-312-3067 Web: www.DeepSoftInc.com

Programming Language Choices

There has always been a fairly wide choice of programming languages to choose from. All generally have good reasons for existing and specific types of tasks to which they are well suited. The "Big Four" in terms of popularity are C, C++, Java, and Visual Basic.

Before you can make an intelligent choice of programming language you need to know the hardware platform the application will run on. Which will either be a general-purpose computer or an embedded system; and the generic type of application you will be writing. Application types include command line, GUI, web based, or an embedded application. Possible types of microprocessors include the following list.

CPU	Central Processing Unit, general purpose digital microprocessors
MCU	Microcontroller Unit, single chip digital microcontrollers
DSP	Digital Signal Processor, digital microprocessors for complex analog signals

Hardware Platforms

General Purpose Computers

General-purpose computers include Windows, Unix, and Apple computers used for a wide variety of commercial, technical, and entertainment applications. These systems only account for roughly 2% of all CPU, MCU, and DSP's sold annually worldwide[1].

Embedded Systems

Embedded systems may use general-purpose computers but are much more likely to be built around custom computers or custom single board computers and used to control specific devices. Embedded systems account for roughly 98% of all CPU's, MCU's, and DSP's sold worldwide annually. For example some new luxury automobiles include as many as 70 processors of varying type, power, and sophistication.

Common Generic Types of Applications

Command line

These applications run from the DOS, UNIX, or mainframe computer's command line. Today these are primarily legacy applications that need repair or necessary upgrades. If you're developing new command line applications, you should ask yourself why. These applications typically perform general-purpose tasks such as accounting, database access, engineering & scientific analysis.

[1] Embedded Systems Programming, May, 2003, Breakpoints, by Jack Ganssle

GUI
Graphical User Interface (GUI) applications are similar to command line applications except that the user interface has been changed to a graphical one. This requires a huge change in how the code is designed and written and results in much greater development time than an equivalent command line application but usually improved end user flexibility and ease of use. Examples are Microsoft Windows, Unix X Windows, and Apple Macintosh applications running on general-purpose computers.

Web
Any computer program running on the World Wide Web. Static web pages do not count since they are not executable programs. Portions of these programs will have to interact with general-purpose computers on each end, and/or embedded systems.

Embedded Systems
Embedded systems come in three flavors. Embedded systems based on general-purpose computers (Windows, Unix, and Apple) that are physically tied to external hardware and dedicated to controlling that hardware. An example would be a space simulation vacuum chamber used to test satellites; a blast furnace in a steel mill; the heating, ventilation, air conditioning, electrical, plumbing, and security system of a large high rise building; or the trains and switches of the metro system in Washington DC. These are normally disk-based systems; that is the program runs from a hard disk drive and frequently uses a real-time based Operating System (RTOS). Depending on the system complexity, it may involve multiple systems running over a network. These are typically 32 or 64 bit applications.

A second variation would be large and powerful custom computers used in confined spaces or where weight, power, features, and performance may be at a premium. Examples include the computers on the Space Shuttle, or any commercial or military jet, the US Army's M1A1 main battle tank, and any naval ship or submarine. These systems may be either disk based or ROM based but they frequently do not run standard operating systems (OS). ROM stands for Read Only Memory and is the acronym for the integrated circuit memory chips used to hold programs that run on CPU's, MCU's, or DSP's that are not disk based. These are typically 16 or 32 bit applications.

A third variation would be custom single board computer's (SBC) used in intelligent instruments. These are normally always ROM based applications, that may occasionally include a hard disk drive. Examples include cell phones, PDA's, GPS receivers, diver's decompression computers, blood pressure monitors, and printers. These are typically 8, 16, or sometimes 32 bit applications.

Language Choices

Visual Basic
VB is the GUI implementation of the programming language Basic and runs only under Microsoft Windows. VB is a procedural language with some Object Oriented (OO) capability. It's a good language for non-programmers or programmers who have to complete tasks of somewhat limited complexity quickly. These are tasks that can tolerate the slower execution speed of compiled Basic. With VB you write the code once for a single platform, Windows, and can't run it on any other platforms.

C
C is a procedural language that is the predecessor to several derivative OO languages, namely C++ and Java. C is arguably the fastest and most efficient high-level computer language ever written. It was designed from the ground up to be extremely efficient and in many cases is an acceptable replacement for assembly language. Today it is used heavily for embedded real-time applications, probably accounting for 75-80% of all code written for embedded applications, which accounts for approximately 98% of all processors sold worldwide annually. C is ideal for 8 and 16 bit embedded applications. I think of C as being "lean and mean" and elegant – although not as elegant as C++.

C++

C++ is a direct descendent of C and was originally called “C with classes”. C++ is a superset of C and formally adds object oriented capabilities to C. C++ is far more powerful and complex than C but is almost always larger and slower. It is ideal for GUI applications or any large OO applications running on 16, 32, or 64 bit processors whether for general-purpose computers or embedded applications. For both C and C++ you write your code once and run it everywhere by buying a new compiler and linker for each new target platform. You normally have to make some code changes to account for hardware platform changes.

Java

Java is an intermediate descendent of C, formally adding object oriented capabilities that C does not have but falling short of the OO capabilities that C++ has. Also Java is slower than C++ and much slower than C. Because Java is less powerful and more restrictive than C++ many claim it is easier to learn and use. It has become very popular for web development where you write your code once and run it everywhere by using Java virtual machines on different types of processors. Generally this works but due to a myriad of Java dialects, because it’s not a standard language, it doesn’t work as well as it should. It’s also popular for web development because the web is so slow that Java can keep up with it. It is a poor choice for embedded real-time applications unless they are very slow.

Ted K. Fryberger, PE
BSME, MSME & OE
President, DeepSoft, Inc.