

Topic 4: Software

EAL keyword sheet - Cambridge IGCSE Computer Science 0984. Write the word in your own language in the last column.

Keyword	Meaning	Example sentence	In my language
software	Programs and instructions that tell the hardware what to do. You cannot touch it.	<i>A game is software.</i>	
system software	Software that runs and manages the computer, e.g. the operating system.	<i>System software controls the hardware.</i>	
application software	Software that lets users do a task, e.g. writing a document.	<i>A web browser is application software.</i>	
operating system (OS)	System software that manages the hardware, files, memory and users.	<i>Windows is an operating system.</i>	
utility software	Programs that help look after the computer, e.g. antivirus.	<i>Disk defragmenter is utility software.</i>	
device driver	Software that lets the operating system talk to a piece of hardware.	<i>A printer needs a device driver.</i>	
firmware	Permanent software stored in a hardware device that tells it how to start and work.	<i>The router has firmware.</i>	
bootloader	A small program that starts the computer and loads the operating system.	<i>The bootloader runs when you press the power button.</i>	
user interface	How the user interacts with the computer, e.g. icons and menus.	<i>A good user interface is easy to use.</i>	
memory management	An OS job: deciding which program uses which part of RAM.	<i>Memory management stops programs using the same space.</i>	
multitasking	Running several programs at the same time.	<i>The OS allows multitasking.</i>	
interrupt	A signal that tells the CPU to stop and deal with something important.	<i>Pressing a key sends an interrupt to the CPU.</i>	
interrupt handler	A small program that deals with an interrupt.	<i>The interrupt handler runs before the CPU returns to its task.</i>	

Keyword	Meaning	Example sentence	In my language
high-level language	A programming language that is close to human language, e.g. Python.	<i>High-level languages are easier to read.</i>	
low-level language	A programming language close to the machine, e.g. assembly or machine code.	<i>Low-level languages control the hardware directly.</i>	
machine code	Instructions written in binary that the CPU can run directly.	<i>The CPU only understands machine code.</i>	
assembly language	A low-level language that uses short words (mnemonics) such as ADD.	<i>Assembly language is easier than machine code.</i>	
assembler	A program that translates assembly language into machine code.	<i>The assembler translates my program.</i>	
compiler	A program that translates all the code at once into machine code before it runs.	<i>A compiler makes a file that can run alone.</i>	
interpreter	A program that translates and runs the code one line at a time.	<i>An interpreter stops at the first error.</i>	
translator	A program that changes code into machine code. Assemblers, compilers and interpreters are translators.	<i>A translator is needed for high-level code.</i>	
source code	The program as written by the programmer.	<i>The source code is written in Python.</i>	
IDE (integrated development environment)	A program with tools for writing, testing and fixing code.	<i>An IDE can show errors in my code.</i>	
debugging	Finding and fixing errors in a program.	<i>Debugging took me an hour.</i>	
syntax error	A mistake in the rules of the language that stops the program from running.	<i>A missing bracket is a syntax error.</i>	