

Grade 4 checklist

64 skills. Tick each one when you can do it. For grade 4 you need the skills for the grades below it too.

Computational thinking

- ☐ **Give benefits of decomposition** Sub-problems are simpler, can be shared across a team, tested separately and reused. (Decomposition and abstraction)
- ☐ **Identify sequence, selection and repetition** Spot each construct in a flowchart, in pseudocode or in Python code. (Flowcharts, pseudocode and program code)
- ☐ **Follow an algorithm to find its output** Work through a flowchart or program step by step with the inputs given. (Flowcharts, pseudocode and program code)
- ☐ **Access an array item by its index** Indexes start at 0, so `scores[2]` is the third item. (Variables, constants and data structures (strings, arrays and records))
- ☐ **Use relational operators to compare values** Use `==`, `!=`, `<`, `>`, `<=` and `>=` to give True or False. (Arithmetic, relational and logical operators)
- ☐ **Trace a simple loop in a trace table** Record each variable's new value on each pass through the loop. (Tracing algorithms and trace tables)
- ☐ **Describe each type of error** Syntax breaks the language rules; logic gives wrong results; runtime crashes while running. (Syntax, logic and runtime errors)
- ☐ **State that binary search needs sorted data** A binary search only works if the list is in order. (Linear and binary search)
- ☐ **Carry out one pass of a bubble sort** Compare adjacent pairs from left to right and swap any pair in the wrong order. (Bubble sort and merge sort)
- ☐ **Describe how a bubble sort works** Repeat passes of comparing and swapping adjacent items until a pass makes no swaps. (Bubble sort and merge sort)
- ☐ **Count comparisons in a linear search** Count how many items are checked before the target is found or the list ends. (Efficiency of algorithms)
- ☐ **Complete a truth table with two inputs** Four rows: 00, 01, 10, 11. (Truth tables and logic)

Data

- ☐ **Convert 8-bit binary to denary** Add the place values (128, 64, 32, 16, 8, 4, 2, 1) of every column that holds a 1. (Binary representation and unsigned binary–denary conversion)
- ☐ **Convert denary 0 to 255 into binary** Work from 128 downwards, writing 1 for each place value you can subtract and 0 for each you cannot. (Binary representation and unsigned binary–denary conversion)
- ☐ **Recognise a negative two's complement number** In two's complement, a 1 in the most significant bit means the number is negative. (Two's complement)
- ☐ **Use the four binary addition rules** $0 + 0 = 0$, $0 + 1 = 1$, $1 + 1 = 0$ carry 1, and $1 + 1 + 1 = 1$ carry 1. (Binary addition and overflow)

-
- ☐ **Apply a logical shift left** Move every bit left by the given number of places and fill the gaps on the right with 0s. (Logical and arithmetic shifts)
-
- ☐ **Apply a logical shift right** Move every bit right by the given number of places and fill the gaps on the left with 0s. (Logical and arithmetic shifts)
-
- ☐ **Recall the hex digits 0–9 and A–F** A = 10, B = 11, C = 12, D = 13, E = 14 and F = 15. (Hexadecimal)
-
- ☐ **State that one hex digit is 4 bits** 16 different digits need 4 bits, because $2^4 = 16$. (Hexadecimal)
-
- ☐ **State how many characters 7-bit ASCII represents** 7 bits give $2^7 = 128$ different codes, from 0 to 127. (Characters and ASCII)
-
- ☐ **Define resolution and colour depth** Resolution is the number of pixels (width × height); colour depth is the number of bits per pixel. (Bitmap images)
-
- ☐ **Define sample rate and bit depth** Sample rate is the number of samples per second (Hz); bit depth is the number of bits per sample. (Sound)
-
- ☐ **State the range of values for n bits** Unsigned: 0 to $2^n - 1$; two's complement: -2^{n-1} to $2^{n-1} - 1$. (Limitations of binary representation)
-
- ☐ **Recall the size of each unit** 4 bits = 1 nibble, 8 bits = 1 byte, 1024 bytes = 1 KiB, and each larger unit is 1024 of the one before. (Units and file size calculations)
-
- ☐ **Define lossless compression** File size is reduced without losing any data, so the original can be restored exactly. (Lossless and lossy compression (incl. RLE))
-
- ☐ **Define lossy compression** Some data is permanently removed, so the original cannot be restored exactly. (Lossless and lossy compression (incl. RLE))
-

Computers

-
- ☐ **Name the parts of the von Neumann architecture** A CPU (control unit, ALU, registers and clock), main memory (RAM), and the address, data and control buses. (Stored program concept and von Neumann architecture)
-
- ☐ **Describe the stored program concept** Instructions and data are stored together in main memory as binary, and are fetched and executed one at a time. (Stored program concept and von Neumann architecture)
-
- ☐ **State the role of each CPU component** The CU decodes and controls, the ALU calculates and compares, registers hold values and the clock keeps everything in step. (CPU components and the fetch–decode–execute cycle)
-
- ☐ **Put the stages of the cycle in order** Fetch, then decode, then execute, repeated for every instruction. (CPU components and the fetch–decode–execute cycle)
-
- ☐ **Explain why secondary storage is needed** RAM is volatile, so the operating system, programs and files must be kept permanently on non-volatile storage. (Main memory and secondary storage)
-
- ☐ **Identify inputs and outputs of an embedded system** Inputs are usually sensors and buttons; outputs are motors, heaters, valves, displays and buzzers. (Embedded systems)
-
- ☐ **Describe what file management does** It organises files in folders, lets them be saved, named, moved and deleted, and controls who can access them. (Operating systems)
-

- ☐ **Describe what backup software does** It copies files to another device or place so they can be restored if the originals are lost or damaged. (Utility software)
- ☐ **State what an audit trail records** Who did what, and when: for example who changed the code, or who logged in and what they accessed. (Robust software: audit trails and code reviews)
- ☐ **Give characteristics of low-level languages** Specific to one type of processor, give direct control of hardware and memory, and are hard to read. (Programming languages and translators)
- ☐ **Give characteristics of high-level languages** English-like keywords, portable, one statement becomes many machine code instructions, must be translated. (Programming languages and translators)

Networks

- ☐ **Give reasons for connecting computers in a network** Sharing resources, sharing files, communicating and managing the computers centrally. (LANs, WANs and why computers are connected)
- ☐ **State the purpose of an IP address** An IP address identifies a device on a network so that data can be delivered to it. (The internet: IP addressing and routers)
- ☐ **Give advantages and disadvantages of wireless** Users can move around and devices are easy to add, but it is slower, has a shorter range and is less secure. (Wired and wireless connectivity and performance)
- ☐ **Convert between bits and bytes** Multiply a number of bytes by 8 to get bits, and divide a number of bits by 8 to get bytes. (Network speeds and transmission time)
- ☐ **Match protocols to tasks** For example, SMTP sends email, FTP transfers files and HTTP requests web pages. (Protocols)
- ☐ **Match protocols to the layers** HTTP, HTTPS, FTP, SMTP, POP3 and IMAP are application; TCP is transport; IP is internet; Ethernet and Wi-Fi are link. (The TCP/IP model)
- ☐ **Give an advantage and disadvantage of each** For example, a bus is cheap, but a break in the backbone stops the whole network. (Network topologies)
- ☐ **Give impacts of a security breach** Data theft, financial loss, fines, damage to reputation and systems being unavailable. (Network security and identifying vulnerabilities)
- ☐ **Give examples of physical security** Locked server rooms, key cards, biometric locks, CCTV and security guards. (Protecting networks)

Issues and impact

- ☐ **Give ways to cut computers' energy use** Switch devices off when not in use, use sleep and power-saving settings, and buy energy-efficient hardware. (Environmental issues)
- ☐ **State why organisations collect personal data** To provide a service, personalise it, target adverts, improve products or sell the data to others. (Personal data: privacy, ownership, consent and data protection)
- ☐ **Describe machine learning** A type of AI that learns patterns from training data instead of following rules written for every situation. (Artificial intelligence, machine learning and robotics)
- ☐ **Describe open-source and proprietary software** Open-source code can be viewed, changed and shared; proprietary code is kept secret and its use is restricted. (Intellectual property and licensing)

- ☐ **Describe the five types of malware** Say how each one gets onto a system and what it does once it is there. (Malware, social engineering and technical vulnerabilities)

- ☐ **Describe common social engineering techniques** Phishing, pretexting, baiting, quid pro quo and shoulder surfing. (Malware, social engineering and technical vulnerabilities)

- ☐ **Encrypt and decrypt with a Caesar cipher** Shift each letter a fixed number of places along the alphabet, wrapping round from Z to A. (Protecting digital systems and data (incl. encryption))

- ☐ **Describe what an acceptable use policy contains** Rules users agree to, such as never sharing passwords or installing software, and the consequences of breaking them. (Protecting digital systems and data (incl. encryption))

Problem solving with programming

- ☐ **Make code easier to read and maintain** Add comments, meaningful identifiers, indentation and white space without changing what the program does. (Developing code: readability, errors, validation and testing)

- ☐ **Identify presence, length, range and pattern checks** Say which validation check a line of code carries out and why it is needed. (Developing code: readability, errors, validation and testing)

- ☐ **Write if, elif and else statements** Choose between two or more paths using conditions, colons and correct indentation. (Sequence, selection and iteration)

- ☐ **Use range() to control a for loop** Know that range(5) gives 0 to 4 and range(1, 6) gives 1 to 5. (Sequence, selection and iteration)

- ☐ **Choose a suitable data type for data** E.g. real for a price, Boolean for yes or no, string for a phone number. (Data types and data structures)

- ☐ **Explain the difference between variables and constants** A variable's value can change while the program runs; a constant's value cannot. (Data types and data structures)

- ☐ **Convert input to the correct data type** Wrap input() in int() or float() before doing any calculation with it. (Input, output and file handling)

- ☐ **Use the six relational operators** Compare values with ==, !=, <, >, <= and >= to get True or False. (Operators)

- ☐ **Tell functions and procedures apart** A function returns a value; a procedure carries out a task and does not return a value. (Subprograms and libraries)

Free notes, worksheets, practice papers and mock exams at chhetriacademy.com. Written for this site; not taken from past papers.
Stuck? 1-to-1 tutoring with a free 30-minute trial lesson.