

Grade 5 checklist

79 skills. Tick each one when you can do it. For grade 5 you need the skills for the grades below it too.

Computer systems (Paper 1)

- ☐ **Know which registers hold data or addresses** The PC and MAR hold addresses; the MDR holds data or an instruction; the accumulator holds data (the results of calculations). (Architecture of the CPU)
- ☐ **Explain how each characteristic improves performance** Link each one to more instructions processed per second or less time waiting for RAM. (CPU performance)
- ☐ **Explain why a device uses an embedded system** Link each characteristic to the device, e.g. low power, so a fitness tracker's battery lasts for days. (Embedded systems)
- ☐ **Compare RAM and ROM** RAM is volatile and can be read from and written to; ROM is non-volatile and read-only. (Primary storage (RAM, ROM, cache, virtual memory))
- ☐ **Compare HDDs and SSDs** SSDs are faster, more durable and use less power; HDDs are cheaper per GB and available with larger capacities. (Secondary storage)
- ☐ **Calculate the storage needed for several files** Convert every file size to the same unit, add them up and compare the total with a device's capacity. (Units and capacity)
- ☐ **Convert between binary and hexadecimal** Split the 8 bits into two nibbles; each nibble is one hex digit from 0 to F. (Binary, denary and hexadecimal)
- ☐ **Add two 8-bit binary numbers** Work from right to left, writing each carry clearly under the next column. (Binary addition, overflow and shifts)
- ☐ **Carry out a left or right shift** Move every bit the given number of places and fill the empty places with 0s. (Binary addition, overflow and shifts)
- ☐ **Use a given code to find others** Codes run in sequence, so if 'A' is 65 then 'D' is 68. (Characters: ASCII and Unicode)
- ☐ **Explain why Unicode was introduced** ASCII has too few codes for other alphabets and symbols; Unicode uses more bits so it can represent far more characters. (Characters: ASCII and Unicode)
- ☐ **Explain what metadata is and why it's needed** Data about the image, such as width, height and colour depth, so the computer can display it correctly. (Images)
- ☐ **Work out colours from colour depth** n bits per pixel give 2^n colours, e.g. 3 bits give 8 colours. (Images)
- ☐ **Explain how sample rate affects quality** More samples per second, so the digital recording is closer to the original sound. (Sound)
- ☐ **Explain how bit depth affects quality** More bits per sample give more possible values, so each sample is recorded more accurately. (Sound)
- ☐ **Give suitable uses of each type** Lossy: photos, music and video; lossless: text, program code and spreadsheets. (Compression)
- ☐ **Define bandwidth and latency** Bandwidth is the amount of data that can be transferred per second; latency is the delay before data arrives. (LANs, WANs and network performance)

-
- ☐ **Give advantages and disadvantages of each model** E.g. client-server has central backups and security but costs more and the server is a single point of failure. (Client-server and peer-to-peer)
-
- ☐ **Describe how a switch uses MAC addresses** A switch reads the destination MAC address and sends the data only to the device it is for. (Network hardware)
-
- ☐ **Describe how a router uses IP addresses** A router reads the destination IP address and forwards packets between networks towards their destination. (Network hardware)
-
- ☐ **Describe the steps of a DNS lookup** The browser sends the domain name to a DNS server, which finds the IP address (asking other servers if needed) and returns it. (The Internet, DNS, hosting and the cloud)
-
- ☐ **Describe the roles of web servers and clients** The browser requests a page; the web server sends the page's files back; the browser displays it. (The Internet, DNS, hosting and the cloud)
-
- ☐ **Give advantages and disadvantages of each topology** E.g. a star is easy to add to but fails if the switch fails; a mesh is reliable but expensive. (Star and mesh topologies)
-
- ☐ **Describe what Bluetooth is used for** Short-range, low-power wireless links between nearby devices, e.g. headphones, keyboards and smartwatches. (Ethernet, Wi-Fi and Bluetooth)
-
- ☐ **Explain how encryption protects intercepted data** Intercepted data is meaningless without the key; only the intended receiver can decrypt it. (Encryption)
-
- ☐ **Encrypt and decrypt with a Caesar cipher** Shift each letter a fixed number of places along the alphabet, wrapping round from Z to A. (Encryption)
-
- ☐ **Describe IPv4 and IPv6 formats** IPv4 is 32 bits written as four numbers from 0 to 255; IPv6 is 128 bits written as eight groups of four hex digits. (IP and MAC addressing)
-
- ☐ **Explain why IPv6 was introduced** IPv4 gives only about 4.3 billion addresses, which is not enough for all connected devices. (IP and MAC addressing)
-
- ☐ **Choose the right protocol for a task** E.g. FTP to upload website files, HTTPS to enter card details, IMAP to keep email in sync across devices. (Standards and common protocols)
-
- ☐ **State benefits of using layers** Layers are self-contained, so they can be changed or developed separately. (The concept of layers)
-
- ☐ **Describe the main types of malware** Virus, worm, trojan, ransomware and spyware, and how each one spreads or causes harm. (Threats to computer systems and networks)
-
- ☐ **Explain how phishing and social engineering work** They trick people into giving away information or access, by exploiting trust, fear or urgency. (Threats to computer systems and networks)
-
- ☐ **Describe how a firewall works** It monitors incoming and outgoing traffic and blocks data that does not meet its rules. (Identifying and preventing vulnerabilities)
-
- ☐ **Describe how anti-malware software works** It scans files, compares them with a database of known malware, and quarantines or deletes threats. (Identifying and preventing vulnerabilities)
-
- ☐ **Describe memory management and multitasking** Loads programs into RAM, gives each its own area, frees memory, and shares the processor between programs. (Operating systems)
-
- ☐ **Explain the role of a device driver** Software that translates the OS's instructions into commands a particular device understands. (Operating systems)
-

- ☐ **Describe how defragmentation improves performance** It moves the parts of each file together, so the read/write head moves less and files load faster. (Utility software)

- ☐ **Explain why compression software is used** Smaller files take up less storage and transfer faster, e.g. to fit an email attachment limit. (Utility software)

- ☐ **Describe environmental impacts of technology** E-waste, energy used by devices and data centres, and mining for raw materials, plus benefits such as less travel. (Ethical, cultural, environmental and privacy issues)

- ☐ **Explain privacy issues of data collection** Location tracking, data sold to advertisers and data breaches can expose personal information. (Ethical, cultural, environmental and privacy issues)

- ☐ **Identify the law broken in a situation** E.g. reading a colleague's emails using their password breaks the Computer Misuse Act 1990. (Legislation)

- ☐ **Describe the principles of the Data Protection Act** Personal data must be used fairly and lawfully, for stated purposes, kept accurate, secure and no longer than needed. (Legislation)

- ☐ **Describe advantages and disadvantages of each** E.g. open source is usually free and customisable but may lack official support. (Open source and proprietary software)

Algorithms and programming (Paper 2)

- ☐ **Apply abstraction to a scenario** Say which details a program or model keeps and which it ignores, using the scenario's own objects. (Computational thinking)

- ☐ **Complete a partly drawn structure diagram** Add the missing boxes at the correct level, under the correct parent box. (Inputs, processes, outputs and structure diagrams)

- ☐ **Complete missing parts of an algorithm** Fill gaps in pseudocode or a flowchart so that it meets the description. (Pseudocode and flowcharts)

- ☐ **Complete a trace table for a loop** Record each new value of every variable, and the output, in the correct order. (Trace tables and common errors)

- ☐ **Carry out a binary search on a list** Find the middle item, compare, discard half the list, and repeat until found or empty. (Linear and binary search)

- ☐ **Describe the steps of a bubble sort** Compare pairs, swap, finish the pass, then repeat passes until a pass makes no swaps. (Bubble sort)

- ☐ **Merge two sorted lists into one** Compare the front item of each list and move the smaller one across, repeatedly. (Merge sort)

- ☐ **Show the list after each insertion** Insert the second item, then the third, and so on, writing the list after each one. (Insertion sort)

- ☐ **Describe the steps of an insertion sort** Start with the second item, shift larger sorted items right, and insert the item in the gap. (Insertion sort)

- ☐ **Trace a sequence of assignments** Work out each variable's value line by line, remembering the right side is worked out first. (Variables, constants, input, output and assignment)

- ☐ **Compare count- and condition-controlled loops** A for loop repeats a set number of times; while and do ... until repeat depending on a condition. (Sequence, selection and iteration)

-
- ☐ **Work out the output of a loop** Follow for loops with step values and while loops, including the last time round. (Sequence, selection and iteration)
-
- ☐ **Evaluate expressions using MOD, DIV and ^** $17 \text{ MOD } 5 = 2$, $17 \text{ DIV } 5 = 3$ and $2^3 = 8$. (Arithmetic, comparison and Boolean operators)
-
- ☐ **Combine conditions with AND, OR, NOT** Work out whether a combined condition is True or False. (Arithmetic, comparison and Boolean operators)
-
- ☐ **Explain what casting is** Converting a value from one data type to another, such as the string "12" to the integer 12. (Data types and casting)
-
- ☐ **Use length, upper and lower** `word.length` gives the number of characters; `.upper` and `.lower` change the case. (String manipulation)
-
- ☐ **Extract part of a string** Use `.substring(start, n)`, `.left(n)` and `.right(n)`, counting positions from 0. (String manipulation)
-
- ☐ **Open, read and close a file** Use `open()`, `readLine()` and `close()` in the correct order. (File handling)
-
- ☐ **Create a file and write to it** Use `newFile()`, `open()` and `writeLine()`, then close the file. (File handling)
-
- ☐ **Declare and assign values in an array** Use `array scores[5]` to declare it, then `scores[0] = 12` to assign a value. (Arrays and records)
-
- ☐ **Explain what a record is** A group of related fields about one item, which can be of different data types. (Arrays and records)
-
- ☐ **State the output of a simple query** Pick out the fields named after `SELECT` for every record that meets the `WHERE` condition. (SQL)
-
- ☐ **Write a query using SELECT, FROM and WHERE** Choose the fields, the table and a condition such as `Minutes < 120`. (SQL)
-
- ☐ **Tell functions and procedures apart** A function returns a value; a procedure does not. (Functions and procedures)
-
- ☐ **Give benefits of using sub programs** Code is reused, easier to test and maintain, and can be shared between programmers. (Functions and procedures)
-
- ☐ **Give uses of random numbers** Games, simulations, choosing random items and generating test data. (Random number generation)
-
- ☐ **Describe ways to make code maintainable** Sub programs, meaningful names (naming conventions), indentation and comments. (Defensive design)
-
- ☐ **Explain what authentication is** Checking that users are who they say they are, for example with a username and password. (Defensive design)
-
- ☐ **Complete a test plan with expected results** Give test data, its type and whether it should be accepted or rejected. (Testing)
-
- ☐ **Compare iterative and final testing** Iterative happens during development; final testing is of the whole program at the end. (Testing)
-
- ☐ **Complete a truth table for two gates** Add a column for the first gate, then work out the final output. (Logic gates, diagrams and truth tables)
-

- ☐ **Give features of high-level languages** English-like, easier to write and debug, and portable between processor types. (Languages and translators)

- ☐ **Give features of low-level languages** Processor-specific, direct control of hardware, fast and memory-efficient. (Languages and translators)

- ☐ **Describe features of the editor** Syntax highlighting, auto-indentation, auto-completion, line numbers. (The Integrated Development Environment)

- ☐ **Complete a test plan for a scenario** Choose normal, boundary, invalid and erroneous data with expected results. (Paper 2 Section B practice: design, write, test and refine)

- ☐ **Complete or correct lines of given code** Fill gaps or fix errors so the code does what the scenario needs. (Paper 2 Section B practice: design, write, test and refine)

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.