

Grade 5 checklist

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

Computational thinking

- ☐ **Decompose a scenario into sub-problems** Break a given system, such as an online booking service, into separate tasks. (Decomposition and abstraction)

- ☐ **State the benefits of using subprograms** Code is written once and called many times, and is easier to test, maintain and share. (Decomposition and abstraction)

- ☐ **Complete a partly drawn flowchart** Fill in missing decisions, processes or outputs so that it does what is described. (Flowcharts, pseudocode and program code)

- ☐ **Explain the benefits of using constants** The value is changed in one place only, and the name makes the code easier to read. (Variables, constants and data structures (strings, arrays and records))

- ☐ **Use string length, indexing and concatenation** Use `len()`, `word[0]` and `+` to work with strings. (Variables, constants and data structures (strings, arrays and records))

- ☐ **Calculate with modulus and integer division** `17 // 5` is 3 and `17 % 5` is 2. (Arithmetic, relational and logical operators)

- ☐ **Combine conditions with AND, OR and NOT** Join comparisons to test a range or several conditions at once. (Arithmetic, relational and logical operators)

- ☐ **Trace selection inside a loop** Check the if condition on every pass and only update the variables it changes. (Tracing algorithms and trace tables)

- ☐ **Identify the type of a given error** Decide whether it stops the program running, gives a wrong result or makes it crash. (Syntax, logic and runtime errors)

- ☐ **Describe how a binary search works** Compare with the middle item and discard the half that cannot contain the target. (Linear and binary search)

- ☐ **Complete a bubble sort and count swaps** Show the list after every pass and count the comparisons and swaps. (Bubble sort and merge sort)

- ☐ **Use test data to check an algorithm** Compare the expected result with the actual result for normal, boundary and erroneous data. (Efficiency of algorithms)

- ☐ **Count passes and comparisons in a bubble sort** In the basic version, each pass through a list of n items makes $n - 1$ comparisons. (Efficiency of algorithms)

- ☐ **Complete a truth table with three inputs** Eight rows, listed in binary counting order from 000 to 111. (Truth tables and logic)

Data

- ☐ **Find the number of patterns for n bits** n bits give 2^n different patterns (states), so 8 bits give 256. (Binary representation and unsigned binary–denary conversion)

-
- ☐ **State the 8-bit two's complement range** 8-bit two's complement stores whole numbers from -128 to $+127$. (Two's complement)
-
- ☐ **Explain why two's complement is needed** Unsigned binary can only store zero and positive numbers; two's complement can also store negative numbers. (Two's complement)
-
- ☐ **Add two 8-bit binary numbers** Add column by column from the right, carrying 1 into the next column when needed. (Binary addition and overflow)
-
- ☐ **Define overflow** An overflow happens when a result is too large to be stored in the number of bits available. (Binary addition and overflow)
-
- ☐ **State the effect of a shift on value** Each place left multiplies an unsigned number by 2; each place right divides it by 2. (Logical and arithmetic shifts)
-
- ☐ **Convert 8-bit binary to hexadecimal** Split the bits into two nibbles and convert each nibble to one hex digit. (Hexadecimal)
-
- ☐ **Convert hexadecimal to 8-bit binary** Convert each hex digit to its 4-bit pattern and join the patterns together. (Hexadecimal)
-
- ☐ **Work out codes from a given code** Letters and digits have consecutive codes, so if 'A' is 65 then 'D' is 68. (Characters and ASCII)
-
- ☐ **Calculate the bits needed to store text** Multiply the number of characters (including spaces) by 7 for 7-bit ASCII. (Characters and ASCII)
-
- ☐ **Find the number of colours** A colour depth of n bits gives 2^n colours, e.g. 8 bits give 256. (Bitmap images)
-
- ☐ **Encode and decode a simple bitmap** Use the given code for each colour to turn the pixels into binary, row by row. (Bitmap images)
-
- ☐ **Read sample values from a graph** Read the amplitude at each sample time and write it as a binary number. (Sound)
-
- ☐ **Find the number of amplitude values** A bit depth of n bits gives 2^n possible amplitude values. (Sound)
-
- ☐ **Explain why a value cannot be stored** A value outside the range needs more bits than are available, so it causes an overflow. (Limitations of binary representation)
-
- ☐ **Convert between units** Divide by 1024 to go to a larger unit and multiply to go to a smaller one (use 8 between bits and bytes). (Units and file size calculations)
-
- ☐ **Choose the right method for a file** Lossless for text, program code and spreadsheets; lossy for photos, music and video. (Lossless and lossy compression (incl. RLE))
-

Computers

-
- ☐ **Explain why programs are loaded into RAM** The CPU can only fetch instructions directly from main memory, which is also much faster to access than secondary storage. (Stored program concept and von Neumann architecture)
-
- ☐ **State what each bus carries, and which way** Address bus: addresses, one-way from the CPU; data bus: data and instructions, both ways; control bus: control signals, both ways. (CPU components and the fetch–decode–execute cycle)
-

- ☐ **Explain why a higher clock speed helps** More clock cycles each second means more instructions can be fetched, decoded and executed each second. (CPU components and the fetch–decode–execute cycle)
- ☐ **Describe how magnetic and optical storage work** Magnetised areas on spinning platters read by a moving head; pits and lands on a disc read by a laser. (Main memory and secondary storage)
- ☐ **Describe the typical features of embedded systems** Dedicated, small, cheap, low power, reliable, with a simple interface and a program kept in non-volatile memory. (Embedded systems)
- ☐ **Explain the role of a device driver** It translates the operating system's general commands into commands that one particular device understands. (Operating systems)
- ☐ **Describe how user management protects a computer** Accounts, passwords and access rights control who can log in and what each user is allowed to do. (Operating systems)
- ☐ **Describe how anti-malware protects a computer** It scans for known malware signatures and suspicious behaviour, then quarantines or deletes threats. (Utility software)
- ☐ **Explain the benefits of data compression** Smaller files take up less storage space and are quicker to send or upload. (Utility software)
- ☐ **Explain why robust software is important** Crashes and vulnerabilities can lose data or money, harm people, and damage an organisation's reputation. (Robust software: audit trails and code reviews)
- ☐ **Describe how a code review finds vulnerabilities** Other programmers check the code line by line for errors and weaknesses before it is released. (Robust software: audit trails and code reviews)
- ☐ **Explain why low-level languages are used** They give direct control of hardware and very efficient code, e.g. for embedded systems and device drivers. (Programming languages and translators)

Networks

- ☐ **Give differences between a LAN and a WAN** Compare the area covered and who owns the communication links, e.g. a LAN covers one site and is owned by the organisation. (LANs, WANs and why computers are connected)
- ☐ **Give disadvantages of a network** For example, malware can spread between connected computers, and the equipment and management cost money. (LANs, WANs and why computers are connected)
- ☐ **Recognise a valid IPv4 address** It has exactly four denary numbers, each from 0 to 255, separated by dots, e.g. 81.24.160.3. (The internet: IP addressing and routers)
- ☐ **Define bandwidth and latency** Bandwidth is the maximum amount of data sent per second; latency is the delay before data arrives. (Wired and wireless connectivity and performance)
- ☐ **Compare copper and fibre-optic cable** Fibre is faster, works over longer distances and is not affected by electromagnetic interference, but costs more. (Wired and wireless connectivity and performance)
- ☐ **Convert between kilobits, megabits and gigabits** Multiply or divide by 1000 for each step, e.g. 3 Mbps = 3,000,000 bits per second. (Network speeds and transmission time)
- ☐ **Calculate time from file size and rate** Use $\text{time} = \text{file size} \div \text{transmission rate}$, with the file size and the rate in matching units. (Network speeds and transmission time)
- ☐ **Explain why protocols are needed** Devices from different manufacturers must follow the same rules to understand the data they exchange. (Protocols)

- ☐ **Explain why HTTPS is used** It encrypts data sent between a browser and a web server, so intercepted data cannot be read. (Protocols)

- ☐ **Describe the role of each layer** Say what each layer does to the data, e.g. the internet layer adds IP addresses and routes packets. (The TCP/IP model)

- ☐ **Describe how data travels in each topology** Bus: along the backbone to every device; star: through the switch to one device; mesh: by one of several routes. (Network topologies)

- ☐ **Define penetration testing and ethical hacking** Both are authorised attempts to find weaknesses in a system so that they can be fixed. (Network security and identifying vulnerabilities)

- ☐ **Explain how access control protects a network** Usernames, passwords and access levels mean only authorised users can reach only the data they need. (Protecting networks)

- ☐ **Define two-factor authentication** Proving your identity in two different ways, e.g. a password plus a code sent to your phone. (Protecting networks)

Issues and impact

- ☐ **Explain the harm caused by e-waste** Toxic substances such as lead and mercury can leak from landfill into soil and water, and valuable metals are wasted. (Environmental issues)

- ☐ **Explain why data centres use so much energy** Thousands of servers run 24 hours a day and must be kept cool, often using electricity generated from fossil fuels. (Environmental issues)

- ☐ **Explain how computers can help the environment** For example, video calls cut travel, smart meters help people use less energy and sensors monitor pollution. (Environmental issues)

- ☐ **Explain privacy and consent issues in context** Say who could see or use the data, whether the person agreed, and what harm could follow. (Personal data: privacy, ownership, consent and data protection)

- ☐ **Describe what data protection law requires** Data must be used openly and only for its stated purpose, limited to what is needed, accurate, not kept too long and secure. (Personal data: privacy, ownership, consent and data protection)

- ☐ **Explain how algorithmic bias happens** If the training data is unrepresentative or reflects past prejudice, the system learns to treat some groups unfairly. (Artificial intelligence, machine learning and robotics)

- ☐ **Explain differences between copyright and patents** Copyright is automatic and protects original work; a patent must be applied for and protects an invention. (Intellectual property and licensing)

- ☐ **Explain how viruses differ from worms** A virus needs a host file and a user action to spread; a worm spreads itself across networks. (Malware, social engineering and technical vulnerabilities)

- ☐ **Explain how anti-malware and encryption protect data** Anti-malware detects and removes malware; encryption makes stolen or intercepted data unreadable without the key. (Protecting digital systems and data (incl. encryption))

Problem solving with programming

- ☐ **Choose normal, boundary and erroneous test data** Pick actual values for a test plan and give the expected result of each test. (Developing code: readability, errors, validation and testing)

- ☐ **Choose count- or condition-controlled loops** Use for when the number of repeats is known and while when it depends on a condition. (Sequence, selection and iteration)

- ☐ **Trace loops to find the output** Use a trace table to follow each variable through every pass of a loop.
(Sequence, selection and iteration)

 - ☐ **Convert between data types by casting** Use `int()`, `float()` and `str()` before calculating or joining text. (Data types and data structures)

 - ☐ **Access and change items in a list** Use an index that starts at 0, `len()` to count items and `append()` to add one. (Data types and data structures)

 - ☐ **Open a file in the right mode** Use "r" to read, "w" to write a new file and "a" to add to the end of a file.
(Input, output and file handling)

 - ☐ **Read and split a CSV file** Loop through the lines, strip the newline and split each line at the commas.
(Input, output and file handling)

 - ☐ **Use integer division and modulus** `17 // 5` gives 3 (the whole number part) and `17 % 5` gives 2 (the remainder). (Operators)

 - ☐ **Combine conditions with and, or, not** E.g. `age >= 13 and age <= 19` is True only for ages 13 to 19.
(Operators)

 - ☐ **Import and use library subprograms** E.g. `import random`, then use `random.randint(1, 6)` for a dice roll.
(Subprograms and libraries)

 - ☐ **Give benefits of using subprograms** Code can be reused, tested on its own and is easier to read and maintain. (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.