

Grade 6 checklist

57 skills. Tick each one when you can do it. For grade 6 you need the skills for the grades below it too.

Computer systems (Paper 1)

- ☐ **Describe the fetch stage in order** The address in the PC is copied to the MAR, the instruction at that address is copied from RAM into the MDR, and the PC is incremented. (Architecture of the CPU)
- ☐ **Explain why more cores don't always help** Some programs can only use one core, some tasks cannot be split up, and cores take time to coordinate. (CPU performance)
- ☐ **Compare embedded and general-purpose computers** A laptop runs many programs that the user can change; an embedded system runs one fixed program. (Embedded systems)
- ☐ **Describe how virtual memory works** When RAM is full, data that is not currently needed is moved to an area of secondary storage, and moved back when needed. (Primary storage (RAM, ROM, cache, virtual memory))
- ☐ **Recommend storage for a given scenario** Choose a device and justify it with characteristics that are linked to how it will be used. (Secondary storage)
- ☐ **Calculate the size of a text file** Bits per character $\times$ number of characters, then divide by 8 for bytes. (Units and capacity)
- ☐ **Convert between denary and hexadecimal** Hex to denary: first digit $\times$ 16 + second digit; denary to hex: divide by 16 and use the remainder. (Binary, denary and hexadecimal)
- ☐ **Explain why hexadecimal is used** It is shorter and easier for people to read and write than binary, so fewer mistakes are made, and it converts easily. (Binary, denary and hexadecimal)
- ☐ **Explain what an overflow error is** The result of a calculation is too large to be stored in the number of bits available, e.g. more than 255 in 8 bits. (Binary addition, overflow and shifts)
- ☐ **State the effect of a shift** Each place of left shift multiplies by 2; each place of right shift divides by 2. (Binary addition, overflow and shifts)
- ☐ **Link bits per character to characters represented** n bits give 2^n characters, so each extra bit doubles the number of characters. (Characters: ASCII and Unicode)
- ☐ **Explain effects of resolution and colour depth** Increasing either improves the quality of the image but increases the file size. (Images)
- ☐ **Explain what affects the file size** Increasing the sample rate, the bit depth or the duration increases the file size. (Sound)
- ☐ **Choose a compression type for a file** Pick lossy where a small loss of quality is acceptable and lossless where the file must stay exactly the same. (Compression)
- ☐ **Explain why a busy network slows down** Link the cause to the effect, e.g. more users share the bandwidth, so each gets less and transfers take longer. (LANs, WANs and network performance)
- ☐ **Explain why a model suits a scenario** Link features such as cost, central backups or security to the needs of the organisation described. (Client-server and peer-to-peer)

- ☐ **Compare copper, fibre and wireless media** Fibre has the highest bandwidth and range and no electrical interference but costs more; wireless gives mobility but is slower. (Network hardware)

- ☐ **Give advantages and disadvantages of the cloud** Access from anywhere and no hardware to maintain, but it needs an Internet connection and a third party holds the data. (The Internet, DNS, hosting and the cloud)

- ☐ **Explain how a mesh network re-routes data** There are several routes between devices, so if one link fails the data is sent a different way. (Star and mesh topologies)

- ☐ **Compare Ethernet and Wi-Fi in detail** Make each point a comparison: bandwidth, reliability, interference, security, mobility and installation cost. (Ethernet, Wi-Fi and Bluetooth)

- ☐ **Give examples of where encryption is used** HTTPS web pages, Wi-Fi networks and the storage on laptops and phones. (Encryption)

- ☐ **Explain how routers and switches use addresses** Routers use IP addresses to route data between networks; switches use MAC addresses to deliver data within a LAN. (IP and MAC addressing)

- ☐ **Compare POP with IMAP** Both retrieve email; POP downloads it and usually deletes it from the server, IMAP keeps it on the server and in sync. (Standards and common protocols)

- ☐ **Explain two benefits of layers in detail** E.g. a change to one layer does not affect the others, so new hardware works without changing applications. (The concept of layers)

- ☐ **Explain why people are the weak point** People can be tricked, choose weak passwords or be careless, and technical defences cannot stop this. (Threats to computer systems and networks)

- ☐ **Explain the purpose of penetration testing** Authorised, simulated attacks find weaknesses so they can be fixed before criminals exploit them. (Identifying and preventing vulnerabilities)

- ☐ **Compare a GUI with a command-line interface** A GUI is easier for beginners; a command line uses fewer resources and is faster for experts. (Operating systems)

- ☐ **Describe how encryption software protects files** Files are scrambled using a key, so a thief cannot read them without the key or password. (Utility software)

- ☐ **Consider the views of different stakeholders** E.g. how one change affects customers, staff, the company and society in different ways. (Ethical, cultural, environmental and privacy issues)

- ☐ **Describe the three Computer Misuse Act offences** Unauthorised access; unauthorised access with intent to commit further offences; unauthorised acts to impair a computer. (Legislation)

- ☐ **Describe what licences allow and forbid** Proprietary licences usually forbid copying, sharing and modifying; open source licences allow these under conditions. (Open source and proprietary software)

Algorithms and programming (Paper 2)

- ☐ **Decompose a given problem into sub-problems** Split a system, such as a booking app or a game, into sensible named tasks. (Computational thinking)

- ☐ **Draw a structure diagram for a scenario** Put the whole system at the top and break it down over two or more levels. (Inputs, processes, outputs and structure diagrams)

- ☐ **Write an algorithm with selection and iteration** Write pseudocode or draw a flowchart that uses decisions and loops correctly. (Pseudocode and flowcharts)

-
- ☐ **Find and correct a logic error** Use the expected output to find the line that gives the wrong result, then rewrite it. (Trace tables and common errors)
-
- ☐ **Compare linear and binary search** Binary search is much quicker on large sorted lists; linear search works on unsorted lists. (Linear and binary search)
-
- ☐ **Explain how the sort knows it has finished** A pass with no swaps means every pair is in order, so the list is sorted. (Bubble sort)
-
- ☐ **Show every stage of a merge sort** Show all the splits down to single items, then each round of merging. (Merge sort)
-
- ☐ **Recognise insertion sort code** Spot an outer loop from the second item and an inner loop that shifts items right. (Insertion sort)
-
- ☐ **Write code that inputs, processes and outputs** Take input, cast it to the right type, calculate a result and output it. (Variables, constants, input, output and assignment)
-
- ☐ **Write code using if, elseif and else** Choose the right order of conditions so that each case gives the correct result. (Sequence, selection and iteration)
-
- ☐ **Write a condition for a range** For 13 to 19 inclusive: `age >= 13 AND age <= 19`. (Arithmetic, comparison and Boolean operators)
-
- ☐ **Explain why casting is needed in code** Input is a string, so it must be cast before arithmetic, or `+` will join the strings. (Data types and casting)
-
- ☐ **Justify storing phone numbers as strings** They can start with 0 or contain spaces, and no arithmetic is done on them. (Data types and casting)
-
- ☐ **Use ASC and CHR** `ASC("A")` gives the character code 65; `CHR(66)` gives "B". (String manipulation)
-
- ☐ **Read every line until end of file** Use a while loop with `NOT file EOF`. (File handling)
-
- ☐ **Loop through a 1D array** Use for `i = 0` to `length - 1` to total, count or search the items. (Arrays and records)
-
- ☐ **Use * to select all fields** `SELECT *` returns every field of each matching record. (SQL)
-
- ☐ **Identify parameters and return values** Parameters receive values passed in; return sends a value back to where it was called. (Functions and procedures)
-
- ☐ **Write code that uses random numbers** Store a random number in a variable and use it in selection or a loop. (Random number generation)
-
- ☐ **Write validation code with a loop** Keep asking for the input until it meets the rule, giving a message if it is invalid. (Defensive design)
-
- ☐ **Explain why boundary data is tested** Errors often happen at the limits, for example `<` used instead of `<=`. (Testing)
-
- ☐ **Write the expression for a logic diagram** Work from the inputs to the output, using brackets to show which gate comes first. (Logic gates, diagrams and truth tables)
-
- ☐ **Draw a logic diagram from an expression** Draw the gates in brackets first, then join their outputs to the final gate. (Logic gates, diagrams and truth tables)
-
- ☐ **Compare compilers and interpreters** A compiler translates the whole program into an executable; an interpreter translates and runs line by line. (Languages and translators)
-

- ☐ **Explain how error diagnostics help** They highlight errors and give messages with line numbers, so bugs are found quickly. (The Integrated Development Environment)
 - ☐ **Write a program with input, selection and iteration** Combine loops, if statements and variables to meet every requirement. (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.