Architecture and design aptitude: study guide

Engineering, technical, and design · suite apt-373-architecture-and-design-aptitude · generated 2026-09-15T15:39:59.554Z

Detail	Value
Title	Architecture and design aptitude: study guide
Generated	2026-09-15T15:39:59.554Z
Fixture/version	apt-373-architecture-and-design-aptitude
Sector	Engineering, technical, and design
Guide version	v2

> Private practice result. Not an official exam certificate, employer decision, hiring signal, admissions decision, or guaranteed outcome. Scores stay on this device unless you export them.

> Answer keys and scoring logic stay server-side and are never included in any download or export.

How to use this guide

This file contains the whole study outline for this suite: every skill it draws on, the full lesson for each of those skills, worked examples, practice tips, a glossary, and where each piece of material comes from. Nothing here is a summary of a page you still have to visit.

1. Read the skills section end to end once, without timing yourself.
2. Work the guided practice mode for the suite, using the practice tips as a checklist.
3. Move to the mini-test only when guided practice feels unhurried.
4. Sit the full simulation last, once, in the conditions you expect on the day.

Practice attempts are stored on the device you used, never on an account. Exporting a result is the only way anything leaves that device.

Practice modes and durations

Practice modes for Architecture and design aptitude

Mode	Duration	What it is for
Guided practice	15 minutes	Untimed, with feedback after every item.
Mini-test	18 minutes	A short timed set for checking pace.
Full simulation	45 minutes	Full length and full time, in one sitting.

- Start guided practice: https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-373-architecture-and-design-aptitude&mode=guided
- Start mini-test: https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-373-architecture-and-design-aptitude&mode=mini
- Start full simulation: https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-373-architecture-and-design-aptitude&mode=full

Skills covered, in full

This suite draws on 5 skill constructs. Each one below carries its complete lesson.

Spatial reasoning

Rotation, folding, views, orientation, maps, and three-dimensional visualization.

Spatial reasoning is the ability to turn an object in your head and be right about the result: to tell a rotation from a mirror image, fold a flat net into a solid and know which faces meet, read plan and elevation views back into a three-dimensional shape, and hold your own orientation steady while a route turns underneath you. It is a heavily weighted construct in pilot and aircrew selection, firefighter and military technical batteries, architecture and design admissions, and trades screening where a drawing has to be built. Much of it can be trained by replacing imagination with a checkable procedure. Practice is device-local unless you export it.

What you should be able to do after this lesson:

1. Distinguish a rotation from a reflection by checking whether the clockwise order of three marked features is preserved or reversed.
2. Read a cube net and name which faces end up opposite, using the rule that squares two apart in a straight strip are opposite.
3. Reconstruct a solid from plan, front and side views, count its unit cubes, and identify when the views leave the count undetermined.
4. Predict the hole pattern produced by unfolding a punched sheet, treating each unfold as a reflection across the fold line rather than a rotation.
5. Work with bearings and relative direction, converting between a map frame and a traveller frame without losing track of which is which.
6. Name the cross-section produced by cutting a cylinder, cone or cube on a given plane, including the oblique cuts people default to calling circles.

Worked examples and pitfalls

Rotation preserves handedness; reflection reverses it: A figure carries three distinguishable marks (say a red band, a blue band and a yellow band), which read red, blue, yellow going clockwise around its centre. Rotating that figure in the plane, by any angle, never changes that clockwise order: it still reads red, blue, yellow clockwise from whatever mark you start at. Mirroring it reverses the order, so a reflected copy reads red, yellow, blue clockwise. That single check answers the whole is-this-a-rotation-or-a-reflection family without imagining any motion. It works on letters, which is why no in-plane rotation of the letter R produces a backwards R, and no rotation of a lower-case b produces a d: b and d are mirror images across a vertical line, and turning one will never give you the other. Contrast b and q, which are related by a half turn and therefore genuinely are rotations of each other. The distinction is exactly what the clockwise-order test measures. In three dimensions the equivalent is handedness at a corner: pick three edges meeting at a vertex, and a rotation preserves whether they form a right-handed or left-handed set while a reflection swaps it. This is exactly why most Western dice, viewed at the corner where the 1, 2 and 3 faces meet, show those numbers running anticlockwise, and a die showing them clockwise is a mirror-image die rather than a differently rotated one. The one honest limitation: a figure with its own mirror line looks identical after reflection, so the test decides nothing for symmetric shapes, which is precisely why item writers use asymmetric ones.

Cube nets: two apart in a strip means opposite: A net made of six squares. Four of them, call them A, B, C and D, sit in a horizontal strip left to right. Square E sits directly above B, and square F sits directly below B. Fold it. The strip of four wraps around the cube as a band, so squares two apart in that strip end up on opposite faces: A is opposite C, and B is opposite D. E and F are the remaining pair, and they become the

top and bottom, opposite each other. The two rules that solve almost every net item are exactly these. Squares separated by one square in a straight line end up opposite. Squares sharing an edge on the net end up adjacent on the cube, and can never be opposite. Everything else is applying them carefully. A die adds one more constraint worth carrying: opposite faces of a standard die sum to seven, so 1 faces 6, 2 faces 5 and 3 faces 4, and a net showing 1 and 6 two apart in a strip is consistent while a net showing 1 and 2 two apart is not a standard die. The classic error on these items is trying to fold the net in your head all at once. Do not. Pick one square as the base, identify its opposite by the two-apart rule, then repeat for a second pair, and you have determined the cube in two cheap steps rather than one expensive act of visualisation.

Three views, one solid, and the cubes you cannot see: A solid is built from unit cubes on a base one cube deep and three cubes wide. The column heights, left to right, are 3, 1 and 2. The front view is therefore a staircase profile of heights 3, 1, 2. The side view, looking along the row from the right, is a single column three cubes tall, because the tallest column hides the others behind it. The plan view is a three-by-one rectangle. Total cubes: 3 plus 1 plus 2, which is 6, and here the three views pin it down exactly because the plan view tells you the solid is only one cube deep. Now change one thing. Suppose the plan view is three by two, so the solid is two cubes deep. The front view can still read 3, 1, 2 while the count is anything from 6, one row of cubes with a hollow back, up to 12, a solid block matching the front profile in both rows. The views constrain the shape but do not determine it, and a well-written item will either supply the count or ask for the minimum and maximum rather than a single figure. That is the trap in this family: candidates who assume every hidden position is filled will over-count, and candidates who assume nothing is hidden will under-count. Read the plan view first to establish the footprint, then use the elevations to cap each column, then ask explicitly whether the item wants the minimum, the maximum, or a determined value.

Punched and unfolded, with coordinates: Take a square sheet 8 units across, with the origin at its centre so the corners are at plus or minus 4 in each direction. Fold the right half onto the left half across the vertical centre line, then fold the bottom half up onto the top half across the horizontal centre line. What is left visible is one quarter, the region where x runs from minus 4 to 0 and y runs from 0 to 4. Punch a single hole at the point minus 3, 3. Now unfold, one step at a time, and treat every unfold as a reflection. Undoing the horizontal fold reflects across the line y equals 0 and adds a hole at minus 3, minus 3. Undoing the vertical fold reflects both existing holes across x equals 0 and adds holes at 3, 3 and 3, minus 3. The finished sheet has four holes, at all four combinations of plus or minus 3. Two rules generalise from this. The number of holes is one punch multiplied by two for each fold the punch passed through, so two folds give four holes and three folds give eight, provided the punch went through every layer, which it does when the folds are through the middle. And the positions are mirror images across each fold line, never rotations. That second point is where the marks go: candidates who place the fourth hole by rotational symmetry get three holes right and one wrong, which is exactly the answer option the item includes.

Bearings, and the difference between the map and the walker: A walker leaves a gate, goes 300 m due north, then 400 m due east. The straight-line distance back to the gate is the hypotenuse of a 300 by 400 right triangle, which is 500 m: the 3-4-5 triangle, so no calculator is needed. The bearing from the gate to the walker is measured clockwise from north, and it is the angle whose tangent is 400 over 300, which is about 53 degrees, so the bearing is 053. The return bearing is that plus 180, which is 233. Bearings are always three digits and always clockwise from north, so 53 degrees is written 053 and a bearing of 350 is very slightly west of north, not south. Now the other half of this construct, which trips up more people than the trigonometry. You are facing south and you turn left. Which way are you now facing? East. Your left hand points east when you face south, and the standard wrong answer, west, comes from reading left as left-on-the-map while forgetting that the map is drawn north-up and you are not. The general fix is to state your heading before every turn and rotate your frame with the traveller: facing north, left is west; facing south, left is east; facing east, left is north; facing west, left is south. On route items, physically rotate the page so the direction of travel points up before working out a turn, and rotate it back only to read off the final compass direction.

What a cut actually reveals: Cross-section items ask what shape a plane makes as it slices a solid, and the reflex answer of circle is wrong most of the time. A cylinder cut perpendicular to its axis gives a circle. Cut parallel to the axis, it gives a rectangle. Cut obliquely so the plane crosses only the curved surface, it gives an ellipse; cut obliquely so the plane also crosses one flat end, it gives an ellipse with one side sliced flat, a D-shape. A cone is richer, because its sections are the conic sections that give the family its name: perpendicular to the axis gives a circle; an oblique cut that crosses every side of the cone gives an ellipse; a cut parallel to one slanted side gives a parabola; a cut parallel to the axis gives a hyperbola branch; and a cut through the apex gives a triangle. A cube is the one that surprises people. A cut parallel to a face gives a square, obviously. A cut through three vertices that are mutually adjacent to one corner gives an equilateral triangle. And a cut through the midpoints of six edges, perpendicular to a long diagonal of the cube, gives a regular hexagon: a six-sided section from a six-faced solid with no curves anywhere. For a flat-faced solid, work these by counting faces rather than by visualising: the section of a convex polyhedron gains exactly one straight edge for every face the plane passes through, so a cut crossing six faces of a cube must close into a six-sided figure and a cut touching only three cannot have more than three sides. For curved solids the question becomes whether the plane meets the curved surface all the way round, which gives a closed circle or ellipse, or only partway, which leaves an open arc closed off by a straight chord.

How to practise this skill

- Replace visualisation with a rule wherever one exists. Two-apart-means-opposite on nets, and clockwise-order-preserved for rotations, both give a checkable answer in seconds and are reliable when mental imagery gets tired late in a section.
- Physically rotate the paper or the device on route and orientation items so the direction of travel points up, work out the turn, then rotate back to read the compass direction. Fighting the north-up convention in your head is the single largest source of left-right errors.
- On view and cube-count items, read the plan view first to fix the footprint, then use the elevations to cap each column, then decide explicitly whether the question wants a minimum, a maximum, or a determined count.
- Treat every unfold as a reflection across the fold line. Say the word reflection out loud as you draw each new hole; the rotational version of the pattern is a deliberately included wrong answer in almost every folding item.
- Build a physical reference once. Fold a paper cube net, mark the faces, and hold it: ten minutes with a real cube fixes the opposite-face rule more firmly than an hour of diagrams, and the memory transfers to items you have never seen.
- Practise untimed first and check whether your errors cluster in rotation, folding, views or orientation, because the four subskills train differently. Attempts are stored on this device only, so an honest error log costs nothing and is what tells you which drill to repeat.

Glossary

Chirality (handedness): The property that distinguishes a figure from its mirror image. Rotation preserves it and reflection reverses it, which is what makes the clockwise-order test decisive for asymmetric figures.

Net: A flat arrangement of faces that folds into a solid. On a cube net, squares two apart in a straight strip become opposite faces and squares sharing an edge become adjacent ones.

Plan view: The view looking straight down on an object. It fixes the footprint of a stacked-cube solid, which is why it is the first view to read.

Elevation: A view looking horizontally at an object, usually given as front and side. Each elevation caps the heights it shows but hides anything directly behind.

Orthographic projection: A drawing convention showing an object through separate parallel views with no perspective. Together the views constrain a solid but do not always determine it.

Bearing: A direction measured clockwise from north and written with three digits, so due east is 090 and a heading slightly west of north is 350. The reverse bearing differs by 180 degrees.

Egocentric versus allocentric frame: The difference between directions relative to a traveller who is turning and directions relative to a fixed map. Left means different compass headings in the two frames, which is why headings must be restated before every turn.

Cross-section: The shape revealed where a plane cuts a solid. It gains one straight edge for each flat face the plane crosses and one curve for each curved surface, which turns the question into a count.

Study this next

- Spatial reasoning skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/spatial-reasoning
- Spatial reasoning practice bank: https://learn.novusstreamsolutions.com/cognitive-skills/spatial-reasoning
- Mechanical comprehension lesson: https://learn.novusstreamsolutions.com/aptitude/skills/mechanical-comprehension/lesson
- Firefighter spatial reasoning suite: https://learn.novusstreamsolutions.com/aptitude/tests/firefighter-spatial-reasoning
- Architecture and design aptitude suite: https://learn.novusstreamsolutions.com/aptitude/tests/architecture-and-design-aptitude
- Spatial reasoning lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/spatial-reasoning/lesson

Where this material comes from

- Worked items written for Novus Learn from standard introductory geometry: cube-net opposite-face rules, orthographic view reconstruction, reflection across fold lines, right-triangle distances and bearings, and the conic and polyhedral cross-sections. The 3-4-5 distance, the 053 and 233 bearings, and the four hole coordinates were each computed and checked.
- Terminology checked against the public Wikipedia articles 'Net (polyhedron)', 'Multiview orthographic projection', 'Chirality', 'Bearing (navigation)' and 'Conic section'. Definitions only; every figure described above is original.
- Novus Learn aptitude construct registry (catalog seed) for the construct scope and related suite mapping.
- Public educational framing only: not affiliated with any official exam board, publisher or employer, and no copyrighted test item is reproduced.

> Educational preparation only. Novus Learn does not administer official exams and does not guarantee scores or hiring outcomes.

Diagrammatic reasoning

Following transformations, flows, processes, systems, and rule diagrams.

Diagrammatic reasoning is the ability to execute a diagram: to run an input through a chain of operator boxes, trace a flowchart with a real value, follow a state machine through a sequence of events, or read a dependency network and say what slips if a task runs late. It appears in technology and engineering screening, business-analyst and process-improvement selection, and any battery that uses transformation-box items. It is also, of the abstract constructs, the one closest to daily work: an approval flow, a deployment pipeline and a fault-finding decision tree are all diagrams somebody has to execute correctly. Practice is device-local and nothing leaves this device unless you export it.

What you should be able to do after this lesson:

1. Apply a chain of defined operators to an ordered input in the correct sequence, and explain why reordering the operators changes the output.

2. Infer an unknown operator from input and output pairs, and identify when a single pair leaves several operators indistinguishable.
3. Trace a flowchart containing a loop and a decision node with a specific starting value, recording the state after every pass.
4. Read a process diagram with threshold decisions and resolve boundary cases correctly against the wording of each condition.
5. Follow a state machine through an input sequence and report both the final state and the number of times a given transition fired.
6. Compute the critical path of a small dependency network, work out the float on each other path, and predict the project impact of a specific delay.

Worked examples and pitfalls

Running an operator chain, in order: Three operators act on an ordered list of four items. Triangle reverses the order of the whole list. Circle swaps the first two items. Square flips the colour of the last item between black and white. The input is: white circle, black square, white triangle, black star. Apply triangle, then circle, then square. After triangle the list is black star, white triangle, black square, white circle. After circle, which swaps the first two, it is white triangle, black star, black square, white circle. After square, which flips the last item, the white circle becomes a black circle, so the output is white triangle, black star, black square, black circle. Now change the order to square, then circle, then triangle. Square flips the last item of the ORIGINAL list, the black star, to a white star. Circle swaps the first two: black square, white circle, white triangle, white star. Triangle reverses: white star, white triangle, white circle, black square. Completely different, because square and triangle both act on positions and reversing the list changes which item is last. That is the whole lesson of operator items. The operators are not commutative, and a candidate who applies them in the order they find easiest will produce a legal-looking output that is wrong. Write the intermediate list after every box. The bookkeeping is the item.

Deducing an operator, and knowing when one example is not enough: You are shown two examples of an unlabelled operator. The list 1, 2, 3, 4 becomes 2, 1, 4, 3. The list A, B, C, D becomes B, A, D, C. The operator swaps items in adjacent pairs: positions one and two exchange, and positions three and four exchange. Test it on a third input, W, X, Y, Z, and it should give X, W, Z, Y. Now the honest caveat that these items are built on. Suppose you had only been shown a two-item example, 1, 2 becoming 2, 1. At least three different operators produce that: swap adjacent pairs, reverse the whole list, and rotate the list by one position. On a two-item list they are indistinguishable, and any answer you give is a guess dressed as a deduction. This is why transformation items normally supply at least two examples of each unknown operator, and why the productive move when you feel certain after one example is to ask what other operator would have produced the same result. Where a test does give you only one example, check the answer options: if two options are consistent with the example, the item is relying on some further constraint elsewhere in the diagram, and that constraint is where the answer lives.

Tracing a flowchart with a loop: A flowchart: START, read a whole number n, set a counter c to zero. Then the loop. Is n equal to 1? If yes, output c and STOP. If no, is n even? If yes, replace n with n divided by 2; if no, replace n with three times n plus one. Then add one to c and return to the top of the loop. Trace it with n equal to 6, writing the state after each pass: n is 6 and even so it becomes 3, c is 1. Three is odd so it becomes 10, c is 2. Ten becomes 5, c is 3. Five is odd so it becomes 16, c is 4. Sixteen becomes 8, c is 5. Eight becomes 4, c is 6. Four becomes 2, c is 7. Two becomes 1, c is 8. Now the top of the loop finds n equal to 1, and the output is 8. Three errors account for nearly all wrong answers here. Testing n even before testing n equal to 1 makes the loop run one pass too far. Incrementing c inside both branches as well as after them double-counts. And abandoning the trace once the value starts rising, six goes up to 10 and then to 16 before it comes down, leads people to assume they have made a mistake. Keep a two-column table of n and c, one row per pass, and the item becomes clerical rather than clever.

Threshold decisions, and the boundary that decides them: A purchase approval flow. A request arrives. First decision: is the amount 500 or less? If yes, the team lead approves and a purchase order is raised. If no, second decision: is the amount 5,000 or less? If yes, the department head approves, finance reviews, and a purchase order is raised. If no, a director approves, finance reviews, and a purchase order is raised. Now three requests. A request for 500.00 exactly takes the first yes branch, because the condition is 500 or less and 500 satisfies it: team lead only, no finance review. A request for 500.01 fails the first decision and passes the second: department head plus finance review. A request for 5,000.00 exactly passes the second decision, so it is also department head plus finance review: not the director. That last one is the item most candidates lose, because 5,000 reads as the top of the range and feels like it should trigger the higher tier. Read the operator, not the number. Or less, up to, and not exceeding include the boundary; under, below and more than exclude it. When a diagram uses a bare inequality symbol the same discipline applies, and if the diagram genuinely does not say, that ambiguity is itself the answer to a well-written question about process risk.

A state machine, and why two coins buy one pass: A turnstile has two states, Locked and Unlocked, and accepts two inputs, coin and push. From Locked, a coin moves it to Unlocked and a push leaves it Locked. From Unlocked, a push moves it to Locked and lets one person through, while a coin leaves it Unlocked. Start in Locked and run the input sequence push, coin, coin, push, push, coin. Step through it. Push in Locked: still Locked, nobody through. Coin: Unlocked. Coin again: still Unlocked: the second coin buys nothing, because the state was already Unlocked and the transition from Unlocked on a coin loops back to Unlocked. Push: Locked, one person through. Push again: still Locked, nobody through. Coin: Unlocked. Final state is Unlocked, and exactly one person passed. The two questions a state-machine item asks are the final state and the count of some transition, and the second is where the marks go. Candidates count coins and answer two, because in the real world two coins ought to buy two entries; the diagram says otherwise, and the diagram is the authority. The general habit is to write the state after every single input in a single row, including the inputs that change nothing, since a transition that loops back to the same state is exactly the one a rushed trace silently drops.

Critical path, float, and which delay actually costs you: A small project. Task A takes 3 days and starts immediately. Task B takes 4 days and follows A. Task C takes 2 days and also follows A. Task D takes 5 days and follows B. Task E takes 6 days and follows C. Task F takes 1 day and cannot start until both D and E are finished. Two routes run from start to finish. A, B, D, F totals 3 plus 4 plus 5 plus 1, which is 13 days. A, C, E, F totals 3 plus 2 plus 6 plus 1, which is 12 days. The project takes 13 days and the critical path is A, B, D, F. The other route has one day of float: it can absorb a single day of delay without moving the finish date, because F still has to wait for D. Now the question these items are built around. Task E slips by two days. Does the project slip? The C-E route becomes 3 plus 2 plus 8 plus 1, which is 14 days, and 14 is greater than 13, so the project finishes one day late. The first day of slip was absorbed by the float and the second was not. The trap answers are no delay, from assuming that a non-critical task is free, and two days, from assuming the delay passes straight through. Float is the buffer, and only the delay beyond it reaches the finish date. Note too that the critical path has now moved to A, C, E, F, which is why the path must be recomputed after any change rather than assumed to be fixed.

How to practise this skill

- Write the intermediate result after every operator box. Diagrammatic items are lost to bookkeeping far more often than to misunderstanding, and an unwritten intermediate state is an unverifiable one.
- Never assume operators commute. If the diagram routes an input through boxes in a particular order, apply them in that order even when a different order looks tidier, because position-based operators interact.
- Trace loops in a two-column table, one row per pass, with the loop variable and the counter side by side. Then check the exit condition is tested where the diagram tests it, not where it feels natural.
- Circle every threshold word in a decision diamond and decide explicitly whether the boundary value is

inside or outside before you evaluate any case. Or less and under are different diagrams.
- On state machines, record the state after every input including the ones that change nothing. Self-loops are the transitions rushed traces drop, and they are usually what the counting question is about.
- Recompute the critical path after applying any delay rather than assuming it stays where it was. Attempts are recorded on this device only, so run a network twice with different delays and watch the path move.

Glossary

Operator box: A labelled transformation applied to an input as it passes through a diagram. Operators are generally non-commutative, so the routing order changes the output.

Decision node: A diamond in a flowchart carrying a condition with two or more labelled exits. The wording of the condition decides which branch a boundary value takes.

Loop: A path in a flowchart that returns to an earlier node. Correct tracing depends on where the exit condition is tested relative to where the variables are updated.

State machine: A diagram of states and labelled transitions between them. The current state plus the next input fully determine what happens, and nothing else in the item does.

Self-loop: A transition that returns a state to itself, such as inserting a coin into an already unlocked turnstile. It consumes the input but changes nothing, and it is the most commonly dropped step in a trace.

Critical path: The longest chain of dependent tasks through a network. Its length is the shortest possible project duration, and any delay on it delays the whole project.

Float (slack): The amount by which a non-critical task can slip before it affects the finish date. Delay beyond the float passes through to the project, and consuming the float can move the critical path.

Swimlane: A horizontal or vertical band in a process diagram assigning each step to a role or team. Handoffs between lanes are where process items usually locate the question.

Study this next

- Diagrammatic reasoning skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/diagrammatic-reasoning
- Diagrammatic reasoning practice bank: https://learn.novusstreamsolutions.com/cognitive-skills/diagrammatic-reasoning
- Abstract reasoning lesson: https://learn.novusstreamsolutions.com/aptitude/skills/abstract-reasoning/lesson
- Business analyst suite: https://learn.novusstreamsolutions.com/aptitude/tests/business-analyst
- Troubleshooting and fault finding suite: https://learn.novusstreamsolutions.com/aptitude/tests/troubleshooting-and-fault-finding
- Diagrammatic reasoning lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/diagrammatic-reasoning/lesson

Where this material comes from

- Every operator chain, flowchart trace, state sequence and network above was constructed and executed by hand for Novus Learn. The loop example follows the Collatz rule, a standard public teaching example, and the trace from n equal to 6 was checked step by step.
- Terminology checked against the public Wikipedia articles 'Flowchart', 'Finite-state machine', 'Critical path method' and 'Float (project management)'. Definitions only; all diagrams described are original.
- Novus Learn aptitude construct registry (catalog seed) for the construct scope and related suite mapping.
- Public educational framing only: not affiliated with any official exam board, publisher or employer, and no copyrighted test item is reproduced.

> Educational preparation only. Novus Learn does not administer official exams and does not guarantee scores or hiring outcomes.

Numerical reasoning

Arithmetic, fractions, percentages, ratios, rates, estimation, word problems, and number relationships.

Numerical reasoning is the ability to take a small set of supplied numbers (a price list, a staffing table, a fuel figure), and reach a defensible answer in around ninety seconds without a formula sheet. It is the most widely used quantitative section in graduate, management and public-sector screening, and it appears in banking, retail, armed forces and healthcare entry batteries alike. The arithmetic itself is deliberately ordinary: percentages, ratios, rates and averages. What is actually being measured is whether you pick the right operation quickly, keep the units straight, and answer the question that was asked rather than the one you started computing. Practice here is device-local: no account, and nothing leaves this device unless you export it.

What you should be able to do after this lesson:

1. Move between fractions, decimals, percentages and ratios on sight, and pick whichever form makes a given item fastest (12.5 percent as one eighth, 33.3 percent as one third).
2. Apply percentage change in both directions, including recovering an original figure from a post-increase total, and combine successive changes multiplicatively rather than by adding them.
3. Split a total in a stated ratio by counting parts first, and work backwards from a stated difference between two shares.
4. Solve rate problems (speed, unit price, consumption, combined work rates) by carrying units through every intermediate line so the units of the answer prove the method.
5. Produce a one-significant-figure estimate before computing, and use it to eliminate the distractors built from the common wrong operations.
6. Read a multi-step stem and name, in one sentence, the single quantity the final question asks for before touching the numbers.

Worked examples and pitfalls

Reverse percentages: the item most candidates get backwards: A subscription price rose by 15 percent and now stands at 57.50 pounds. What was it before? The correct move is to divide, not subtract: the new price is 115 percent of the old, so the old price is 57.50 / 1.15 = 50.00. Check it forwards: 15 percent of 50 is 7.50, and 50 + 7.50 = 57.50. The tempting wrong answer takes 15 percent of the NEW figure, 0.15 x 57.50 = 8.625, and reports 48.88. That distractor is always on the option list because it is what most people do under time pressure, and it is wrong by 2.25 percent, small enough to look plausible. The same asymmetry drives the successive-change item: a price that rises 20 percent and then falls 20 percent does not return to where it started. Multiply the factors: 1.20 x 0.80 = 0.96, a net fall of 4 percent. Starting at 500 pounds you get 600 then 480, not 500. Percentages compose by multiplication; they never add.

Ratio splits: count the parts before you divide: A 4,830 pound budget is split between three teams in the ratio 3:5:6. Add the parts first: 3 + 5 + 6 = 14. One part is 4,830 / 14 = 345. The shares are 3 x 345 = 1,035, 5 x 345 = 1,725 and 6 x 345 = 2,070, and they sum back to 4,830, which is the check you should always run. Two classic failures. The first is dividing by 3 because there are three teams. That gives 1,610 each and ignores the ratio entirely. The second is treating 5 as a fraction and computing five sixths or five fourteenths of something other than the total. The more interesting version of this item gives you a difference instead of the total: 'the second team receives 690 pounds more than the first, what is the whole budget?' The difference between the shares is 5 - 3 = 2 parts, so one part is 690 / 2 = 345, and the budget is 14 x 345 = 4,830. Same number, reached from the other end, and the arithmetic is trivial once you have made the parts explicit.

Rates and units: 2 h 15 min is 2.25, not 2.15: A delivery van covers 174 km in 2 hours 15 minutes. Average speed is distance over time, and the time must be in hours: 15 minutes is 15/60 = 0.25 h, so 174 / 2.25 = 77.3 km/h. Type 2.15 into the calculator instead and you get 80.9 km/h: an error of roughly 4.7 percent that sits comfortably inside the plausible range and will match one of the options. Now extend it. The van consumes 8.6 litres per 100 km, so the trip needs 174 x 8.6 / 100 = 14.964 litres, and at 1.48 pounds per litre that is 14.964 x 1.48 = 22.15 pounds. Notice the units doing the work: (km) x (L / 100 km) leaves litres; (L) x (pounds / L) leaves pounds. If your intermediate line has km still attached at the end, you have divided when you should have multiplied. Write the unit next to every number and the method checks itself.

Unit pricing: normalise before you compare: Three pack sizes of the same fluid. Pack A: 750 ml for 3.60 pounds. Pack B: 2 litres for 9.40. Pack C: 1.5 litres for 7.20. Convert all three to price per litre. A: 3.60 / 0.75 = 4.80 per litre. B: 9.40 / 2 = 4.70. C: 7.20 / 1.5 = 4.80. So B is cheapest by 10p a litre, and A and C are identical despite looking like different deals. The 'bigger pack is cheaper' heuristic happens to hold here but is not a rule, and test writers know it. Half of these items are built specifically so the largest pack loses. Now add the multibuy that these questions love: pack A is on three-for-two. Three packs give 2.25 litres for the price of two, 7.20 pounds, which is 7.20 / 2.25 = 3.20 per litre and beats everything. The trap in the multibuy version is dividing by the number of packs paid for rather than the volume received.

Combined work rates: add rates, never times: Printer A completes a 3,000-page run in 50 minutes; printer B completes the same run in 75 minutes. Running together, how long? Convert to rates: A prints 3,000 / 50 = 60 pages per minute, B prints 3,000 / 75 = 40 pages per minute, and together they print 100 pages per minute, so the job takes 3,000 / 100 = 30 minutes. Verify by counting output: in 30 minutes A produces 1,800 pages and B produces 1,200, which is 3,000 exactly. The two wrong answers you will see on the option list are 62.5 minutes (the average of 50 and 75) and 125 minutes (the sum). Both are impossible on inspection: two machines working together must finish faster than the faster machine alone, so any answer above 50 minutes is wrong before you compute anything. The general form is 1 / (1/50 + 1/75), and it is worth being able to write that line directly, but the sanity bound catches the error faster than the algebra does.

Estimate first, then compute: killing distractors in ten seconds: 'A department spent 847,300 pounds in 2024. In 2025 the budget fell by 12.5 percent. What was the 2025 spend?' Options: 741,387.50 / 953,212.50 / 105,912.50 / 762,570. Estimate before anything else: 12.5 percent is exactly one eighth, one eighth of roughly 850,000 is roughly 106,000, so the answer is roughly 744,000. That single line eliminates three options. 953,212.50 applied the change upwards. 105,912.50 is the size of the reduction, not the resulting spend: the 'answered the wrong question' distractor, and the most commonly selected wrong option on items of this shape. 762,570 is a 10 percent cut, planted for anyone who misread the rate. Exact working: 847,300 x 7/8 = 5,931,100 / 8 = 741,387.50. Doing it as a fraction avoids the decimal multiplication altogether. Learn the fraction equivalents cold (12.5 percent is 1/8, 16.7 percent is 1/6, 37.5 percent is 3/8, 62.5 percent is 5/8) because a fraction turns most percentage items into one division.

How to practise this skill

- Memorise the fraction-to-percentage table up to eighths and twelfths. Almost every 'nasty' percentage on these tests is a clean fraction in disguise, and the fraction route is two or three times faster than decimal multiplication.
- Write the unit beside every intermediate number, including the ones you keep in your head. Most numerical errors on timed sections are unit errors, not arithmetic errors, and units are the only cheap way to catch them.
- Rehearse reverse percentages as their own drill until dividing by 1.15 feels more natural than subtracting 15 percent. It is a single, identifiable technique that accounts for a disproportionate share of lost marks.
- Before selecting, re-read the last clause of the stem out loud. 'By how much did it fall' and 'what was it after the fall' have different answers, and both are always on the option list.
- Keep an error log with four columns (misread the question, wrong operation, unit slip, arithmetic slip), and

tally which one you actually commit. Three sessions is usually enough to show that one column dominates, and that is the column to train.
- Work untimed until your method is stable, then add the clock in stages: generous, then realistic, then 20 percent tighter than the real test. Attempts stay on this device, so a slow first pass costs nothing.

Glossary

Percentage point: The arithmetic difference between two percentages. A rate moving from 4 percent to 5 percent rises by one percentage point but by 25 percent in relative terms; questions specify which they want and the two answers are both on the option list.

Reverse percentage: Recovering an original amount from a figure that already includes a percentage change, by dividing by the multiplier (1.15 for a 15 percent rise, 0.88 for a 12 percent fall) rather than subtracting the percentage from the new figure.

Multiplier (decimal factor): The single number a quantity is multiplied by to apply a percentage change: 1.20 for plus 20 percent, 0.80 for minus 20 percent. Successive changes are handled by multiplying the factors, which is why plus 20 then minus 20 gives 0.96, not 1.

Ratio part: One unit of a ratio split. Dividing the total by the sum of the ratio terms gives the value of one part; every share and every difference between shares is then a whole number of parts.

Unit rate: A quantity expressed per one of something: price per litre, pages per minute, litres per 100 km. Comparisons are only valid once every option has been converted to the same unit rate.

Combined rate: The sum of two or more individual rates working simultaneously. Times cannot be added or averaged; only rates add, and the combined time is the total work divided by the combined rate.

Significant figure estimate: Rounding every input to one meaningful digit to get an approximate answer in a few seconds. Its purpose is not accuracy but elimination: it removes any option that is off by a factor or has the sign of the change wrong.

Distractor: A wrong option written deliberately to match a specific predictable error: subtracting instead of dividing, reporting the change instead of the result, averaging instead of combining rates. Recognising the pattern is often faster than recomputing.

Study this next

- Numerical reasoning skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/numerical-reasoning
- Numerical reasoning practice in the Cognitive Skills Lab: https://learn.novusstreamsolutions.com/cognitive-skills/numerical-reasoning
- Data interpretation lesson: https://learn.novusstreamsolutions.com/aptitude/skills/data-interpretation/lesson
- Office numeracy suite: https://learn.novusstreamsolutions.com/aptitude/tests/office-numeracy
- Finance, accounting and insurance careers: https://learn.novusstreamsolutions.com/careers/families/finance-accounting-and-insurance
- Numerical reasoning lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/numerical-reasoning/lesson

Where this material comes from

- All worked figures above were written and checked for Novus Learn. Every division, ratio split and rate calculation in this lesson was verified by recomputing it in the reverse direction.
- Conventions only (percentage point, unit rate, significant figures) cross-checked against standard public references such as the Wikipedia articles 'Percentage', 'Ratio' and 'Rate (mathematics)'. No item text is drawn from any published test.

- Novus Learn aptitude construct registry (catalog seed) for construct scope and suite mapping.
- Public educational framing only: not affiliated with any official exam board, publisher or employer, and no copyrighted test item is reproduced.

> Educational preparation only. Novus Learn does not administer official exams and does not guarantee scores or hiring outcomes.

Critical thinking

Evaluating evidence, assumptions, arguments, credibility, and alternative explanations.

Critical thinking assessments ask you to take an argument apart: what is being concluded, what it rests on, what evidence would settle it, and whether a proposed conclusion actually follows. Published critical-thinking batteries typically run five task types - inference, recognition of assumptions, deduction, interpretation, and evaluation of arguments - and they are deliberately built so that agreeing with a conclusion and judging the argument as strong come apart. The construct is heavily weighted in policy analysis, audit, legal, investigative and graduate selection, and it is the same skill that stops a plausible chart from turning into a bad decision. Practice stays on this device; there is no account and nothing is uploaded unless you export it.

What you should be able to do after this lesson:

1. State an argument's conclusion in your own words before evaluating it, and identify which sentences are premises and which are background.
2. Use the negation test to separate a required assumption from a statement that would merely strengthen the argument.
3. Distinguish validity from truth, and identify affirming the consequent as distinct from the valid modus tollens form.
4. Compute a posterior probability on a screening example and explain why the rate of true positives among all positives is far lower than intuition suggests.
5. Generate at least two alternative explanations - selection, reverse causation, a common third factor - for any claimed causal effect, and name the comparison that would rule them out.
6. Judge argument strength on relevance and directness rather than on agreement, including marking arguments you personally reject as strong.

Worked examples and pitfalls

The negation test finds assumptions; nothing else does: Argument: 'The council should fit bin sensors across the district. In the trial depot they cut collection trips by a fifth.' Candidate assumption A: 'The trial depot's waste pattern is broadly representative of the rest of the district.' Negate it - suppose the trial depot is nothing like the rest of the district - and the argument collapses, because a result that does not transfer supports nothing about the district. So A is a required assumption. Candidate assumption B: 'Bin sensors are the cheapest available technology for this purpose.' Negate it - suppose they are the most expensive - and the argument still stands as given, since it argued from a reduction in trips, not from cost. B is not required; it would strengthen a cost-based argument that was never made. Candidate C: 'Fewer collection trips reduce total operating cost.' Negate it and the recommendation loses its point, so C is required too, even though the argument never mentions cost. That last case is the one candidates miss, because the required assumption is what bridges the evidence to the recommendation, and bridging assumptions are by definition unstated. The failure mode across this whole item type is picking statements that support the conclusion instead of statements the argument cannot survive without.

Base rates: 90 percent accurate, 15 percent right: A production line runs an automated defect test. Defects occur in 1 percent of units. The test flags 90 percent of genuinely defective units, and wrongly flags 5 percent of good units. A unit has just been flagged - how likely is it to be defective? Work in whole units rather than

probabilities. Take 10,000 units: 100 are defective and 9,900 are good. Of the 100 defective, the test flags 90. Of the 9,900 good, it wrongly flags 5 percent, which is 495. Total flags: 90 + 495 = 585. Of those, 90 are genuine, so the answer is 90 / 585 = 15.4 percent. Better than the 1 percent base rate, and nowhere near the 90 percent that intuition offers. The error has a name - confusing the probability of a flag given a defect with the probability of a defect given a flag - and it has a practical consequence: the rework queue has to be sized for 585 units a batch, not 100, and 495 of the 585 units sitting in it are perfectly good. Whenever a test, screening rule or model accuracy figure appears in an item, ask how many of the negatives there are, because with a rare condition the false positives from a large clean population swamp the true positives from a small affected one.

Three alternative explanations, and the comparison that settles it: Claim: 'Employees who attended the optional resilience workshop took 30 percent fewer sick days last year, so the workshop works.' Alternative one, selection: attendance was optional, so the people who signed up may have been the healthier and more engaged to begin with, and would have taken fewer sick days regardless. Alternative two, reverse direction: employees who were frequently unwell were the least able to attend a full-day workshop, so illness determined attendance rather than the other way round. Alternative three, a common third factor: the workshop ran on Thursday afternoons at head office, so attendance largely tracks being an office-based rather than shift-based worker, and shift workers take more sick leave for reasons that have nothing to do with resilience training. What would settle it: randomly assigning the offer to half the workforce and comparing the two groups. Failing that, the cheapest useful check already exists in the payroll data - compare attendees' and non-attendees' sick days in the year before the workshop. If attendees were already 30 percent lower, selection is doing the entire job. The habit worth building is to name the comparison group before you accept any before-and-after number.

Deduction: affirming the consequent versus modus tollens: Rule: 'If an invoice is over 10,000 euro, it requires two signatures.' Argument A: 'This invoice has two signatures, therefore it is over 10,000 euro.' Invalid - this is affirming the consequent. The rule makes two signatures necessary for large invoices; it says nothing that prevents a cautious manager from double-signing a 400 euro invoice, so a two-signature invoice of any size is consistent with the rule. Argument B: 'This invoice has only one signature, therefore it is not over 10,000 euro.' Valid - this is modus tollens, denying the consequent to deny the antecedent, and it holds assuming the rule was followed. Argument C: 'This invoice is not over 10,000 euro, therefore it does not have two signatures.' Invalid again, denying the antecedent. Two further points that deduction items test directly. First, validity is about form alone: 'All banks close on Sundays; Riverton Mutual is a bank; therefore Riverton Mutual closes on Sundays' is perfectly valid even if the first premise is false, and a valid argument with a false premise can deliver a false conclusion. Second, in these items you must accept the premises as given even when you know them to be untrue, because the question is whether the conclusion follows, not whether it is true.

Interpretation: what a survey figure does and does not license: Data: all 1,200 employees of a firm were surveyed. 62 percent said they would accept a four-day week at 90 percent pay. Of those who said yes, 71 percent were under 35. Conclusion one: 'A majority of this firm's employees would accept the trade.' This follows - 62 percent of a complete census of the firm is a majority, and the conclusion is properly limited to this firm. Conclusion two: 'A majority of the firm's under-35 employees would accept the trade.' This does not follow, and the arithmetic shows why. Sixty-two percent of 1,200 is 744 accepters, and 71 percent of 744 is about 528 accepters aged under 35. That is the composition of the accepters, not the acceptance rate within the under-35 group, and without knowing how many under-35s the firm employs the rate is undetermined: if there are 900, the rate is 528 / 900 = 59 percent; if there are 600, it is 88 percent; if there were 1,100 the rate would be below half. Conclusion three: 'The firm should move to a four-day week.' This does not follow either - a stated preference about pay says nothing about output, shift cover, customer hours or cost. Notice that conclusion two is the base-rate error from the screening example, wearing survey clothing.

Strong and weak arguments on the same question: Question: should the agency publish raw inspection

scores for every site? Strong argument for: 'In the neighbouring authority, publication was followed by a measurable fall in repeat violations, which is the stated aim of the inspection regime.' It is directly about this decision, it addresses the regime's own objective, and it offers evidence - and if true it would change your view. Weak argument for: 'Yes, because transparency is always good.' Sweeping, unsupported, and it would apply identically to publishing anything at all, which is the tell. Strong argument against: 'Raw scores omit the severity weighting, so a site with one critical failure can rank above a site with six minor ones, and a member of the public cannot see the difference.' Specific, mechanistic, directly about the proposal, and it would change your view. Weak argument against: 'No, because businesses will object.' It may be true and it may matter politically, but it does not bear on whether publication achieves the regime's aim, and no evidence is offered. Two tests separate strong from weak: is the argument about the exact question asked, and would accepting it change the decision? Note that you can hold a firm personal view on publication and still be required to mark the argument on the other side as the strong one - that split is what the item type is measuring.

How to practise this skill

- Negate every candidate assumption out loud. If the argument survives the negation, it was never an assumption, however supportive it sounds. This single habit converts the assumption item type from guesswork into a mechanical check.
- Write the conclusion in your own words before reading any option. Half of the wrong answers on inference and evaluation items are responses to a conclusion the argument never reached.
- Ask 'percentage of what?' on every percentage in the stimulus and reconstruct the denominator in whole units. Base-rate and composition errors both dissolve the moment you write out counts instead of rates.
- Deliberately practise marking arguments you disagree with as strong and arguments you agree with as weak. Assessments in this construct are built to catch agreement masquerading as evaluation, and the effect is largest on politically loaded stimuli.
- For any causal claim, write two alternatives - selection and a third factor - and name the comparison group that would rule them out, before you decide whether the evidence supports the claim.
- Log wrong answers by task type rather than by topic: inference, assumption, deduction, interpretation, evaluation. Candidates are rarely weak across all five, and the profile tells you where the next hour of practice belongs.

Glossary

Conclusion: The claim an argument is trying to establish. It is not always last, is often signalled by therefore, so, or should, and locating it correctly determines every subsequent judgement about the argument.

Assumption: An unstated premise the argument requires in order to work. Identified by negation: negate it and a genuine assumption brings the argument down, while a merely helpful statement leaves it standing.

Validity and soundness: An argument is valid when the conclusion cannot be false while the premises are true, which is a property of form alone. It is sound when it is valid and the premises are actually true.

Affirming the consequent: The invalid pattern 'if P then Q; Q; therefore P'. It is the most common deductive distractor because it differs from the valid modus tollens form by only the position of a negation.

Base rate: How common something is in the population before any test or evidence is applied. Ignoring it makes accurate-sounding tests appear far more informative than they are when the condition is rare.

Confounder: A third factor associated with both the supposed cause and the outcome, capable of producing the entire observed relationship on its own. Ruled out by randomisation or by an explicit comparison group.

Selection effect: A difference between groups created by how people entered them rather than by the treatment under study. Optional programmes and voluntary surveys are where it appears most often.

Falsifiability: The property of a claim that some observation could show it to be wrong. A claim compatible

with every possible result carries no information, which is why 'what would change your mind?' is a diagnostic question.

Study this next

- Critical thinking skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/critical-thinking
- Practise the critical thinking bank: https://learn.novusstreamsolutions.com/cognitive-skills/critical-thinking
- Verbal reasoning lesson: https://learn.novusstreamsolutions.com/aptitude/skills/verbal-reasoning/lesson
- Government audit suite: https://learn.novusstreamsolutions.com/aptitude/tests/government-audit
- Investigator and detective reasoning suite: https://learn.novusstreamsolutions.com/aptitude/tests/investigator-and-detective-reasoning
- Critical thinking lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/critical-thinking/lesson

Where this material comes from

- Worked items written for Novus Learn. The arguments, survey figures, defect-test numbers and evaluation options above are original and invented for this lesson; no published or copyrighted test item is reproduced.
- The five task types named in the summary - inference, assumption, deduction, interpretation, evaluation of arguments - describe a structure used publicly across several critical-thinking assessments; no affiliation with any publisher is claimed or implied.
- Terminology follows standard, widely published usage in introductory logic and research methods - validity, soundness, affirming the consequent, base rate, confounder, selection effect.
- Novus Learn aptitude construct registry (catalog seed) for the construct scope and the suite mapping shown in the related links.
- Public educational framing only - not affiliated with any official exam board, publisher or employer.

> Educational preparation only. Novus Learn does not administer official exams and does not guarantee scores or hiring outcomes.

Situational judgment

Evaluating workplace responses against role-relevant principles.

Learn a repeatable way to compare workplace responses: establish the facts, identify duties and risks, respect role boundaries, then choose a proportionate first action. This is educational preparation, not an official scoring guide.

What you should be able to do after this lesson:

1. Separate facts stated in a scenario from assumptions that the scenario does not support.
2. Rank response options by immediate risk, policy or role obligations, proportionality, and follow-through.
3. Explain why a strong first action is better than passive, punitive, or unauthorized alternatives.

Worked examples and pitfalls

Worked scenario: an unverified safety concern: A colleague reports a possible equipment fault while a deadline is approaching. First distinguish the known fact, the report, from the unverified cause. A strong response protects people and affected work, checks the concern through the right channel, tells the relevant lead, and records what was done. Ignoring the report underreacts; shutting down unrelated work or accusing someone before checking the facts overreacts.

Method: facts, duties, risks, response: Write four short notes before ranking options: what is known, who may

be affected, which duty or boundary applies, and what safe next step is available now. Prefer an action that addresses the immediate issue and creates useful follow-through. Do not reward an option merely because it sounds decisive.

How to practise this skill

- Answer the question asked: best first action, worst action, or complete response are different tasks.
- Check whether an option acts within the person's authority and escalates only as far as the risk requires.
- When two options look reasonable, prefer the one that gathers missing facts and communicates ownership.

Glossary

Proportionality: Matching the urgency and scope of a response to the evidence, likely impact, and authority available.

Role boundary: The limit of what a person may decide or do without approval, specialist help, or escalation.

Follow-through: Confirming ownership, recording the decision, and checking that the issue was actually resolved.

Study this next

- Situational judgement test: https://learn.novusstreamsolutions.com/search?q=Situational%20judgement%20test
- Workplace ethics: https://learn.novusstreamsolutions.com/search?q=Workplace%20ethics
- Decision-making: https://learn.novusstreamsolutions.com/search?q=Decision-making
- Situational judgment lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/situational-judgement/lesson

Where this material comes from

- Novus Learn original situational-judgment suite scenarios and published construct mapping.
- Novus educational framework: facts, duties, risks, role boundaries, proportional action, and follow-through.

> Educational preparation only. Novus Learn does not administer official exams and does not guarantee scores or hiring outcomes.

Recommended preparation

- Practice spatial reasoning: https://learn.novusstreamsolutions.com/aptitude/skills/spatial-reasoning/lesson
- Practice diagrammatic reasoning: https://learn.novusstreamsolutions.com/aptitude/skills/diagrammatic-reasoning/lesson
- Practice numerical reasoning: https://learn.novusstreamsolutions.com/aptitude/skills/numerical-reasoning/lesson

Answer keys, scoring and privacy

> Answer keys and scoring logic stay server-side and are never included in any download or export.

Formal suite answer keys and scoring logic stay on the server and are not part of any download, in any format. Downloadable keys exist only for the open, untimed practice material (the practice packs on the

puzzles, cognitive-skills and reasoning practice lab pages), where the answers are already public teaching content.

Novus Learn needs no account. Your practice history lives in this browser's storage on this device, and is sent to a server only if you create an account and switch on backup. This file contains no attempt, session or result link, so it is safe to share.

Private practice result. Not an official exam certificate, employer decision, hiring signal, admissions decision, or guaranteed outcome. Scores stay on this device unless you export them.

Answer keys and scoring logic stay server-side and are never included in any download or export.

Source: https://learn.novusstreamsolutions.com/aptitude/tests/architecture-and-design-aptitude/study-outline