Wonderlic-style short cognitive familiarization: study guide

Commercial assessment-provider familiarization · suite apt-361-wonderlic-style-short-cognitive-familiarization · generated 2026-09-15T15:29:50.686Z

Detail	Value
Title	Wonderlic-style short cognitive familiarization: study guide
Generated	2026-09-15T15:29:50.686Z
Fixture/version	apt-361-wonderlic-style-short-cognitive-familiarization
Sector	Commercial assessment-provider familiarization
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 Wonderlic-style short cognitive familiarization

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-361-wonderlic-style-short-cognitive-familiarization&mode=guided
- Start mini-test: https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-361-wonderlic-style-short-cognitive-familiarization&mode=mini
- Start full simulation: https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-361-wonderlic-style-short-cogniti

ve-familiarization&mode=full

Skills covered, in full

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

Processing speed

Completing simple cognitive operations accurately and efficiently.

Processing speed is how fast you can carry out simple cognitive operations that each require a decision (substituting symbols for digits, adding two small numbers, judging which of two words comes later alphabetically) repeated many times under a clock. It differs from perceptual speed in that a rule has to be applied, not just a match made, and that difference is why automating the rule is the whole game. It appears in short commercial cognitive batteries, game-based assessments, data entry screening and several graduate sifts. Practice here is device-local: no account, and results stay on this device unless you export them.

What you should be able to do after this lesson:

1. Break a speeded item into its three costs (encoding the stimulus, deciding, producing the response), and identify which of the three is your own bottleneck.
2. Work a symbol-to-digit substitution drill, and calculate the point at which memorising the key repays the time spent learning it.
3. Use the section's stated scoring rule to choose a target error rate, and show why the optimum is neither zero errors nor maximum speed.
4. Recognise task-switch cost, and reorder work into same-type blocks wherever the test format allows it.
5. Compare two practice sessions of different lengths using rate and error rate rather than raw totals, and log alongside each result the conditions that genuinely move a speeded score: input device, warm-up, interruptions.
6. Automate the small arithmetic and comparison operations these sections are built from, so the decision step approaches zero.

Worked examples and pitfalls

Substitution: turning a lookup into a recall: A key maps six letters to digits: Q = 1, W = 2, E = 3, R = 4, T = 5, Y = 6. The sequence W E Y Q R T E W becomes 2 3 6 1 4 5 3 2. Trivial, and that is the point. The item content is not the difficulty. Break the per-item cost into three parts: a glance up to the key and back, the match itself, and writing or keying the digit. Say those cost roughly 0.3, 0.4 and 0.5 seconds while you are still reading the key, giving 1.2 seconds per item and 72 seconds for sixty items. Once the six pairs are held in memory, the glance disappears and each item costs about 0.9 seconds, so sixty items take 54 seconds: a saving of about 25 percent. The investment arithmetic matters: at 0.3 seconds saved per item, ten seconds spent deliberately rehearsing six pairs is repaid after about 34 items and everything after that is profit, whereas thirty seconds of rehearsal would need 100 items to break even. So on a short section, learn the key quickly and imperfectly; on a long one, learn it properly. Use your own measured times rather than the illustrative ones above. The structure of the decision is what transfers, not the numbers.

The optimum error rate is not zero: A 90-second section scored one point per correct answer, nothing deducted for a wrong one. Three paces. Careful at 1.8 seconds an item: 90 / 1.8 = 50 attempted, 96 percent right, 48 correct and 2 wrong. Brisk at 1.2 seconds: 75 attempted, 92 percent right, 69 correct and 6 wrong. Reckless at 0.75 seconds: 120 attempted, 70 percent right, 84 correct and 36 wrong. Under raw scoring the totals are 48, 69 and 84, so faster keeps winning right up to the point where accuracy collapses entirely, and

most candidates sit nowhere near that point, which is why 'slow down and be careful' is such common and such bad advice on a no-penalty section. Now change the rule to correct minus incorrect: 48 - 2 = 46, 69 - 6 = 63, 84 - 36 = 48. The brisk pace wins and the reckless pace has fallen back to roughly where the careful one sits. Change it again to correct minus twice incorrect: 44, 57 and 12. Brisk wins by more, and reckless is now catastrophic. Three scoring rules, three different optimal paces, and under none of them is the answer either extreme. The instruction screen gives you the penalty; only your own practice log gives you the accuracy you actually hold at each pace.

Switch cost: the same items, slower, for free: Take forty items, twenty additions and twenty alphabetical comparisons. Present them in two blocks of twenty and people complete them faster than when the same forty items are interleaved one after the other, even though not a single item has changed. The extra time is switch cost: reconfiguring the mental task set on every alternation. If a blocked pace is 1.0 second an item and a mixed pace is 1.25, eighty items cost 80 seconds blocked and 100 seconds mixed, and the 20-second difference bought you nothing. The practical consequences are concrete. On a paper form or a scrollable list where you control the order, do all items of one type first and then the other. On a randomised screen where you cannot, expect the mixed rate and do not interpret it as having got worse. And in real work, batching the same kind of task (all the invoices, then all the emails) is the same effect, which is why an interruption costs far more than the seconds it occupies.

Rate, not raw score: comparing two practice sessions: Session 1: six minutes, 148 correct, 9 wrong. Session 2: four minutes, 112 correct, 4 wrong. On raw correct, session 1 looks better by 36. Convert to rates: 148 / 6 = 24.7 correct per minute against 112 / 4 = 28.0. Convert the errors to rates too: 9 out of 157 attempted is 5.7 percent, against 4 out of 116 which is 3.4 percent. Session 2 is better on both dimensions, faster and more accurate, and the raw comparison had it backwards purely because it ran for two minutes less. This is the most common self-assessment error in speed practice, and it matters because it drives the wrong training decision: the candidate concludes that longer sessions suit them and keeps practising in a way that inflates the number they are watching. Log four figures per session: duration, attempted, correct, wrong. Everything else is derivable, and no comparison should ever be made on a figure that has duration baked into it.

Automating the easy arithmetic: Processing speed sections use arithmetic that is deliberately easy, which means what is being measured is whether it has become automatic. Compensation is the main technique: 38 + 47 becomes 40 + 45 = 85, moving two across so there is no carry to track. Rounding and correcting handles multiplication: 6 x 24 is 6 x 25 - 6 = 150 - 6 = 144, and 49 x 8 is 50 x 8 - 8 = 392. Percentages decompose: 15 percent of 320 is 10 percent, 32, plus half of that, 16, giving 48. Division by four is halving twice, so 96 becomes 48 becomes 24. None of these is clever, and that is exactly why they work under time pressure. Each replaces a multi-step procedure with one you can run without holding intermediate state. Drill them until you stop noticing which method you used. The distinction that matters is between knowing a shortcut and having automated it; on a two-second-per-item section, a shortcut you have to consciously select is barely faster than the long way.

What actually moves a speeded score on the day: Four things reliably cost points and all four are controllable. First, an unfamiliar input device: practising on a laptop and sitting the test with an external mouse, or drilling on a full keyboard with a dedicated numeric pad and meeting a compact one where the digits live on the top row, changes a motor skill you had automated. Second, interruptions: a notification mid-block costs the item you were on plus the next one or two while the task set is rebuilt, which is switch cost arriving uninvited. Third, no warm-up: the first thirty seconds of a cold speeded section are measurably slower, so two minutes of low-stakes practice before a timed attempt is not superstition. Fourth, unrecorded conditions: a bad session with no note of why looks in your log like a decline in ability. Write down the device, the time of day, the sleep and whether you warmed up. A speeded score is sensitive to all of these in a way a reasoning score simply is not, and half of an apparent plateau usually turns out to be uncontrolled conditions rather than a real ceiling.

How to practise this skill

- Warm up for two minutes before any timed attempt. Cold starts under-report by enough to make an untimed comparison meaningless, and the warm-up costs less time than the items it saves.
- Practise on the device and layout you will actually use. If the test is on a desktop with a mouse, do not train on a trackpad; if it involves numeric entry, do not train on the keyboard top row.
- Read the penalty rule first and pick a pace from arithmetic, not nerve. Write your own accuracy at two different speeds in a log so the arithmetic has real inputs.
- Where you control the order of items, batch by type to avoid switch cost. Where you do not, accept the mixed rate rather than trying to force the blocked one and making errors instead.
- Drill the small arithmetic until you stop choosing a method. Automaticity, not knowledge of shortcuts, is what a two-second-per-item section rewards.
- Record duration, attempted, correct and wrong for every session, plus the conditions. Attempts stay on this device unless you export them, and four numbers per session is enough to see a genuine trend within a fortnight.

Glossary

Processing speed: The rate at which simple cognitive operations requiring a decision can be completed accurately. Distinguished from perceptual speed by the presence of a rule to apply rather than only a match to make.

Substitution (coding) task: A format supplying a key that maps symbols to digits or letters, with the candidate transcribing a long sequence through the key. Performance improves sharply once the key is recalled from memory rather than read.

Speed-accuracy trade-off: The relationship in which going faster raises the error rate. It has an optimum for any given scoring rule, and the optimum sits at neither extreme.

Switch cost: The extra time taken per item when task types alternate rather than being grouped, caused by reconfiguring the mental task set. It applies even when the items themselves are unchanged.

Automaticity: The state in which an operation runs without deliberate selection or monitoring. It is the practical goal of drilling small arithmetic, because a consciously chosen shortcut is barely faster than the long method.

Response time: The full interval from stimulus to completed response, including the physical act of keying or writing. Broader than reaction time, which refers only to the interval before the response begins.

Practice effect: The improvement that comes from familiarity with a format rather than from any change in underlying ability. It is largest on the first few attempts, which is why an early practice score is a poor baseline.

Error rate: Wrong answers as a proportion of attempted items. It is the only error figure comparable across sessions of different lengths, and it must be logged alongside rate for either to be interpretable.

Study this next

- Processing speed skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/processing-speed
- Processing speed practice in the Cognitive Skills Lab: https://learn.novusstreamsolutions.com/cognitive-skills/processing-speed
- Perceptual speed lesson: https://learn.novusstreamsolutions.com/aptitude/skills/perceptual-speed/lesson
- Short cognitive assessment familiarization suite: https://learn.novusstreamsolutions.com/aptitude/tests/criteria-style-cognitive-assessment-familiarization
- Game-based assessment familiarization suite: https://learn.novusstreamsolutions.com/aptitude/tests/game-based-assessment-familiarization
- Processing speed lesson on Novus Learn:

https://learn.novusstreamsolutions.com/aptitude/skills/processing-speed/lesson

Where this material comes from

- Every score, rate and break-even calculation above was worked out and re-checked for this lesson, including the three scoring rules applied to the same three paces.
- Per-item timings, the blocked-versus-mixed pace figures and the 0.3-second lookup saving are illustrative arithmetic chosen to show the structure of each trade-off. They are not measured research results, and your own logged times should replace them.
- Concepts (speed-accuracy trade-off, task-switching cost, automaticity, practice effect) cross-checked against standard public references such as the Wikipedia articles 'Speed-accuracy tradeoff', 'Task switching (psychology)' and 'Mental chronometry'.
- 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.

Verbal reasoning

Drawing conclusions from written information without relying on outside assumptions.

Verbal reasoning is the discipline of answering strictly from a passage: deciding whether a statement is True, False, or Cannot Say on the evidence given, and refusing to import anything you happen to know about the topic. It is the workhorse construct in graduate and civil-service screening, banking and consulting sifts, and language-focused public-service batteries, where a passage about parcel volumes or grant conditions is followed by four statements to classify. The same discipline is what stops a contract review, a grant assessment or an incident summary from quietly acquiring facts nobody wrote down. 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. Classify a statement as True, False, or Cannot Say, and state in one sentence which words in the passage force the classification.
2. Explain the difference that costs the most marks: 'Cannot Say' means undetermined by the passage, while 'False' means contradicted by it.
3. Spot quantifier drift between passage and statement (all, some, most, only, none) and show why 'all A are B' never licenses 'all B are A'.
4. Refuse a causal upgrade: recognise when a passage reports co-occurrence or sequence and a statement claims cause.
5. Separate what a passage asserts from what it reports somebody else asserting, and classify statements about each correctly.
6. Do the small arithmetic hidden inside verbal items - percentage changes, fractions of a stated total - without leaving the passage for outside data.

Worked examples and pitfalls

The three-way rule on one short passage: Passage: 'The Riverton depot handled 4,200 parcels in March, a 12 percent increase on February. A night shift was introduced in January; in March it processed 40 percent of the parcels the depot handled.' Now take four statements. (1) 'The depot handled 3,750 parcels in February.' February is March divided by 1.12: 4,200 / 1.12 = 3,750 exactly, so this is True - the passage determines it even though the number is not printed. (2) 'The night shift caused the March increase.' Cannot Say. The

passage puts the night shift and the increase in the same depot in the same period; it never claims one produced the other, and it never denies it either. Most candidates mark this False, reasoning correctly that correlation is not causation but then choosing the wrong label: False is reserved for statements the passage contradicts. (3) 'The night shift processed more than 1,700 parcels in March.' Forty percent of 4,200 is 1,680, which is not more than 1,700, so this is False - contradicted by arithmetic on the passage's own figures. (4) 'The depot handled more parcels in March than in January.' Cannot Say: January is mentioned only as the month the shift started, and no January volume is given.

Quantifier drift and illicit conversion: Passage: 'All accredited suppliers submit quarterly audits. Some suppliers in the northern region are accredited.' Statement A: 'Some suppliers in the northern region submit quarterly audits.' True, and it is worth seeing why the chain holds: at least one northern supplier is accredited, and every accredited supplier submits, so at least one northern supplier submits. Statement B: 'Every supplier that submits quarterly audits is accredited.' This reverses the first sentence. 'All accredited suppliers submit' leaves the door wide open for unaccredited suppliers to submit as well - perhaps voluntarily, perhaps under another rule - so the answer is Cannot Say. Turning 'all A are B' into 'all B are A' is called illicit conversion and it is the single most productive trap in this format, because the reversed sentence sounds like a paraphrase. Statement C: 'No unaccredited supplier submits quarterly audits' is the same reversal wearing a negative coat, and it is Cannot Say for the same reason. Statement D: 'All northern suppliers submit quarterly audits' is also Cannot Say: 'some are accredited' says nothing about the rest.

Asserted versus reported: Passage: 'The committee's report claims that the scheme cut waiting times by a third. Two of the four regional boards have disputed that figure.' Statement A: 'The scheme cut waiting times by a third.' Cannot Say. What the passage asserts is that a report claims this; the claim's truth is never settled, and the dispute does not settle it either. Statement B: 'At least two regional boards disagree with the report's waiting-time figure.' True, and note 'at least' - the passage says two disputed it, and two of four disputing is consistent with 'at least two'. Statement C: 'All four regional boards accept the figure.' False, directly contradicted. Statement D: 'The report is wrong.' Cannot Say. This item type appears constantly in policy and diplomacy passages, where a paragraph stacks a claim, a source, and a reaction, and the reader has to keep three different truth-conditions apart. The reliable habit is to underline the reporting verb - claims, argues, estimates, alleges, projects - and remember that everything downstream of it belongs to the source, not to the passage.

Percentages are not symmetric: Passage: 'Membership fell from 8,000 to 6,000 over two years, then recovered by 25 percent in the third year.' Statement: 'Membership at the end of year three was higher than at the start.' The fall is 2,000 on a base of 8,000, which is 25 percent. The recovery is 25 percent of the new base: 6,000 x 1.25 = 7,500. That is below 8,000, so the statement is False. The trap is elegant, because a 25 percent fall followed by a 25 percent rise feels like it should cancel. It does not: to get back from 6,000 to 8,000 you need a 33.3 percent rise, since 2,000 / 6,000 = 0.333. Whenever a verbal item states a change as a percentage, write down what the percentage is a percentage of before you touch it. A related version supplies the recovery as an absolute number instead - 'recovered by 2,000 members' - which does return the total to 8,000, and candidates who answered the first version from memory get the second one wrong.

Verbal analogies: name the relation as a sentence: Item: 'ANTISEPTIC is to INFECTION as ___ is to ___.' Options: (a) vaccine : immunity, (b) insulation : heat loss, (c) medicine : illness, (d) bandage : wound. Write the stem relation as a full sentence first: 'An antiseptic is applied in order to prevent an infection from occurring.' Now test each option against that exact sentence. Option (a) runs the other way - a vaccine is given to produce immunity, not to prevent it. Option (c) is the right family but the wrong specificity: medicine typically treats an illness that has already started. Option (d) is applied after the wound exists. Option (b) fits precisely: insulation is applied in order to prevent heat loss from occurring. The general method is the same on the simpler item types. 'BOOK is to LIBRARY as PAINTING is to ___' has canvas (the material), artist (the maker), and gallery (the place a collection is kept and shown) among its options; all three are genuine relations to 'painting', and only the third matches the stem sentence. Options in analogy items are chosen to

be true statements about the words, just not the stated relation.

Two premises with 'some' prove nothing: Argument: 'Some depot managers are qualified engineers. Some qualified engineers hold a rigging certificate. Therefore some depot managers hold a rigging certificate.' This is invalid, and the way to prove it is a counter-model rather than an argument. Let the depot managers be Ana and Ben; let the qualified engineers be Ben and Cara; let the certificate holders be Cara alone. Premise one holds: Ben is a manager and an engineer. Premise two holds: Cara is an engineer with the certificate. The conclusion fails: neither Ana nor Ben holds the certificate. Since a single consistent world makes both premises true and the conclusion false, the argument cannot be valid, so in a True / False / Cannot Say framing the conclusion is Cannot Say. Two premises that both begin 'some' never combine into a conclusion, because 'some' gives you an overlap without telling you where the overlap sits. Contrast a valid pair: 'All accredited labs are inspected annually. No inspected facility is exempt from the fee.' Every accredited lab is inspected, and no inspected facility is exempt, so no accredited lab is exempt - True.

How to practise this skill

- Answer from the passage even when you know the subject. Candidates with a background in the topic score worse on verbal items than they expect, because their own knowledge quietly supplies the missing premise that turns a Cannot Say into a True.
- Run the two-worlds test on every candidate 'Cannot Say': can you imagine a world where all the passage's sentences hold and the statement is true, and another where they hold and it is false? If both, it is Cannot Say. If only the false world exists, it is False.
- Underline the quantifiers and hedges in the statement (all, some, only, most, may, must, likely) and find their counterparts in the passage. Roughly half of the wrong answers in this construct come from a single word swapped between the two.
- Budget about 25 to 30 seconds per statement rather than per passage, and read the passage once for structure before touching the statements. Re-reading the whole passage for each statement is what causes candidates to run out of time on the last set.
- Keep a wrong-answer log with one label per error: quantifier, causation, reported-versus-asserted, arithmetic, outside knowledge. A clear pattern almost always emerges within about forty items, and it is usually a single label.
- Work untimed until the three-way rule is automatic, then add the clock. Attempts are stored on this device only, so a slow first pass through a passage set costs you nothing but the time you spend on it.

Glossary

Entailment: A statement is entailed by a passage when it cannot be false while every sentence of the passage is true. Entailment is the only thing that earns a 'True' in this format; being plausible, likely, or well known does not.

Cannot Say: The verdict for a statement the passage neither entails nor contradicts. It is a claim about the passage, not about the world, which is why a statement you know to be true in real life can still be Cannot Say.

Quantifier: A word fixing how much of a group a claim covers: all, most, some, few, none, only. Swapping one quantifier for another changes the logical content completely while barely changing how the sentence reads.

Illicit conversion: The invalid move from 'all A are B' to 'all B are A', or from 'if P then Q' to 'if Q then P'. It preserves the words and destroys the logic, which is why converted sentences make such effective wrong answers.

Counter-model: A concrete, consistent scenario in which the premises hold and the conclusion fails. Producing one is the fastest possible proof that an argument is invalid, and it takes two or three named

individuals.

Hedge: A qualifier such as may, could, is expected to, or is associated with. A hedged sentence in a passage cannot support an unhedged statement, and an unhedged passage sentence is not weakened by a hedged statement.

Reporting verb: A verb such as claims, argues, estimates, or alleges that attributes the following content to a source. Everything downstream of it is the source's assertion, and the passage takes no position on it.

Analogy relation: The specific link between the two stem words in an analogy item - tool and user, item and container, action and purpose. Naming it as a full sentence before reading the options removes almost all of the guesswork.

Study this next

- Verbal reasoning skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/verbal-reasoning
- Practise the verbal reasoning bank: https://learn.novusstreamsolutions.com/cognitive-skills/verbal-reasoning
- Critical thinking lesson: https://learn.novusstreamsolutions.com/aptitude/skills/critical-thinking/lesson
- Reading comprehension lesson: https://learn.novusstreamsolutions.com/aptitude/skills/reading-comprehension/lesson
- General civil-service aptitude suite: https://learn.novusstreamsolutions.com/aptitude/tests/general-civil-service-aptitude
- Verbal reasoning lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/verbal-reasoning/lesson

Where this material comes from

- Worked items written for Novus Learn. Every passage, statement set and figure above is original and invented for this lesson; no published or copyrighted test item is reproduced.
- Terminology follows standard, widely published usage in introductory logic and assessment writing - entailment, quantifier, illicit conversion, counter-model.
- 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.

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.

Logical reasoning

Conditional logic, syllogisms, ordering, grouping, argument structure, and constraint problems.

Logical reasoning is the ability to take a set of statements (conditionals, quantifiers, ordering and grouping constraints), and work out exactly what they force, what they permit, and what they leave open. It is the backbone of graduate screening batteries, law and policy admissions tests, analyst and investigator selection, and the constraint sections of programming aptitude tests. The same discipline runs through eligibility policy, contract clauses, access-control rules and any specification written as if-then. 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. Translate everyday phrasing (only if, unless, no A is B, none but) into a single arrow of the form antecedent implies consequent, and get the direction right every time.
2. Derive the contrapositive of any conditional and use it to reason backwards from a denied consequent.
3. Name and reject the two invalid conditional moves, affirming the consequent and denying the antecedent, in items designed to make both feel natural.
4. Solve a linear ordering item to a single arrangement by placing the most restrictive block first and eliminating cases exhaustively.
5. Answer a grouping item by identifying who is ruled out, not just who is possible, and distinguish must be true from could be true.
6. Apply quantifier rules correctly: recognise that some means at least one and does not imply not all, and that two overlapping some statements never guarantee a third overlap.

Worked examples and pitfalls

One conditional, four moves, two of them wrong: Take a warehouse rule: if a pallet is flagged by the scanner, then it is inspected before dispatch. Write it as F implies I. Four things can now happen. You learn a pallet was flagged: F is true, so I is true: inspected. That is modus ponens and it is valid. You learn a pallet was not inspected: not-I, so not-F. It was not flagged. That is modus tollens, equally valid, and it is the move most candidates never reach for. Now the two traps. You learn a pallet WAS inspected and conclude it must have been flagged. Invalid: the rule says nothing about why else a pallet might be inspected: random audit, a customer complaint, a damaged corner. That is affirming the consequent. You learn a pallet was NOT flagged and conclude it was not inspected. Invalid for the same reason; that is denying the antecedent. Both feel right because in ordinary conversation people say if when they mean if and only if. Assessment items exploit exactly that slip, so the fix is mechanical: the instant you meet a conditional, write F implies I on one line and its contrapositive not-I implies not-F underneath. Those two lines are everything the statement gives you. Anything else on the page is a distractor.

Chaining conditionals, then running the chain backwards: Three statements from a release policy. If the build fails, the deploy is blocked. If the deploy is blocked, the release date slips. If the release date slips, the client is notified. Write them: B implies D, D implies S, S implies N. Chaining forwards is easy: a failed build guarantees a notified client. The item almost never asks that. It says: the client was not notified. What follows? Take contrapositives and chain them the other way: not-N implies not-S, not-S implies not-D, not-D implies not-B. So the build did not fail, the deploy was not blocked, and the date did not slip. All three follow. Now the near-miss version, and the one candidates get wrong: the client WAS notified. What follows? Nothing about the build. The chain only runs one way, and a client can be notified for reasons the three statements never mention. Answer: none of the above must be true. A useful habit for chain items is to draw the arrows on one line, B implies D implies S implies N, and remember that you can travel left-to-right when you are given a truth and right-to-left when you are given a falsehood, never the reverse.

The four phrases that flip the arrow: Most lost marks in conditional items come from translation, not from reasoning. Only if introduces the consequent: you may board only if you hold a boarding pass means board implies pass. It does NOT mean a pass gets you on the plane. The pass is necessary, not sufficient, and you still need the gate to be open and your name off the no-fly list. Unless is read as if not: unless you check in you cannot board is not-check-in implies not-board, whose contrapositive is board implies check-in, again the check-in is necessary. None but staff may enter is enter implies staff. No temporary badge grants server-room access is temporary badge implies not server-room access. Compare all four with a plain sufficient condition: swiping a manager card opens the door is card implies open, and here the card really is enough. The discipline is to ask one question of every clause. Is this thing the guarantee or the requirement? A guarantee sits at the tail of the arrow, a requirement sits at the head. Get that one decision right and the rest of the item is bookkeeping.

An ordering item, solved to a single arrangement: Five people present in five consecutive slots, one each:

Aisha, Ben, Chen, Dana, Eli. Constraints: (1) Ben presents in the slot immediately before Dana. (2) Aisha is neither first nor last. (3) Chen presents at some point before Ben. (4) Eli is not immediately before or immediately after Dana. (5) Dana is not last. Start with the most restrictive item, the Ben-Dana block, and test each placement. Slots 1-2 is impossible: Chen must come before Ben and there is no slot before 1. Slots 4-5 is impossible because Dana would be last, breaking (5). Slots 3-4: Chen takes 1 or 2, and Aisha must take 2 because she cannot be first or last, so Chen is 1 and Eli is forced into 5, but 5 is immediately after Dana in slot 4, breaking (4). Dead. Slots 2-3: Chen must be 1. Aisha and Eli take 4 and 5, and since Aisha cannot be last, Aisha is 4 and Eli is 5. Check (4): Dana is in 3, the neighbouring slots are 2 and 4, and Eli is in 5: fine. The order is Chen, Ben, Dana, Aisha, Eli, and it is the only one. Two habits made that quick. First, place the block before the singletons; a two-slot block only has four possible positions in a five-slot line, so it does most of the elimination for you. Second, when a case dies, write down which constraint killed it. Later questions in the same set often ask what happens if one rule is dropped, and your dead-case notes answer them instantly.

A grouping item where the real answer is who is excluded: Exactly three of six people are picked: Farah, Gus, Hana, Ivo, Jo, Kit. Rules: (1) If Farah is picked, Gus is not. (2) Hana is picked only if Ivo is. (3) At least one of Gus and Jo is picked. (4) Kit and Ivo are never both picked. The question: if Hana is picked, which of the following must be true? Work it. Rule (2) is Hana implies Ivo, so Ivo is in. That is two of the three seats. Rule (4) then puts Kit out. The third seat goes to Farah, Gus or Jo. Suppose it went to Farah: then neither Gus nor Jo is picked, breaking (3). So the third seat is Gus or Jo, and both of those complete a legal team. Check Gus: rule (1) is vacuous because Farah is out, (2) holds, (3) holds, (4) holds. Check Jo: same. So the answer is not who joins Hana; that is genuinely open. The answer is that Farah is NOT picked, and it must be true in every legal arrangement. Grouping items are built around that distinction. Could be true means one arrangement exists; must be true means no counter-arrangement exists. The fastest way to kill a must-be-true option is to build a single legal team that violates it, which is why sketching two complete valid teams before reading the options is usually faster than reasoning about the options one at a time.

Some, all and none: the overlap nobody gave you: Two premises: all compliance officers have completed the ethics module, and some people who completed the ethics module work in the Leeds office. Does it follow that some compliance officers work in Leeds? No, and the diagram shows why. Draw a big circle for ethics-module completers. Compliance officers sit entirely inside it. Leeds staff overlap it somewhere, but nothing places that overlap on top of the compliance officers, so a world where every Leeds completer is a lab technician satisfies both premises and refutes the conclusion. Two overlapping particular statements never chain. Now three rules worth memorising. First, some means at least one and is silent about the rest, so the premise some invoices are overdue does not rule out all of them being overdue, though it does not establish that either. An option reading all invoices are overdue is therefore could-be-true rather than must-be-true, and a candidate who takes some to mean some but not all will wrongly mark it impossible. Second, some does convert: if some completers are Leeds staff then some Leeds staff are completers, and that is valid. All does not convert. All compliance officers are completers tells you nothing about most completers. Third, a valid chain needs a universal: all A are B plus no B are C gives no A are C, and that one is airtight. When an item mixes quantifiers, sketch three circles before you read a single answer option; the arrangement that satisfies the premises while breaking the conclusion is usually visible in about ten seconds.

How to practise this skill

- Notate before you reason. Every conditional gets two lines on your paper, the statement and its contrapositive, written in symbols. Items that look like paragraphs collapse to four or five arrows, and the arrows are what the question is actually about.
- Underline only if, unless, none but and no as you read. Those four phrasings cause more errors than every inference rule combined, and they are the cheapest thing on the page to get right.
- On ordering sets, draw a numbered row of slots and place the largest fixed block first. Keep a note of which constraint killed each dead case; follow-up questions in the same set reuse them.

- On must-be-true options, attack rather than verify. One complete legal arrangement that breaks the option kills it outright, and building one is usually faster than proving the option holds everywhere.
- Train the invalid moves deliberately. Write out an affirming-the-consequent item and a denying-the-antecedent item of your own each session until the shape of them is instantly recognisable under time pressure.
- Work untimed until you can state which rule licensed each step, then add the clock. Attempts are stored on this device only, so a slow, fully-narrated first pass through a constraint set costs nothing but your own time.

Glossary

Antecedent and consequent: In a conditional if P then Q, P is the antecedent and Q is the consequent. Getting the two the right way round during translation is where most conditional items are won or lost.

Contrapositive: For P implies Q, the statement not-Q implies not-P. It is always logically equivalent to the original, which is what lets you reason backwards from a denied consequent.

Modus ponens: The valid inference from P implies Q together with P to the conclusion Q. The most common inference in assessment items, and the safest.

Modus tollens: The valid inference from P implies Q together with not-Q to the conclusion not-P. Under-used by candidates, and the move most backwards-chaining items are built around.

Affirming the consequent: The invalid move from P implies Q together with Q to the conclusion P. It is tempting because ordinary speech often means if and only if when it says if.

Necessary condition: Something that must hold for an outcome to occur, but does not by itself bring it about. Signalled by only if, unless, none but, and required for.

Sufficient condition: Something whose presence guarantees the outcome. Signalled by a plain if, whenever, or any time that. A condition can be necessary, sufficient, both, or neither.

Must be true versus could be true: Must be true holds in every arrangement the constraints permit; could be true holds in at least one. Nearly every grouping and ordering answer option is one of these two, and mistaking which is being asked costs the mark.

Study this next

- Logical reasoning skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/logical-reasoning
- Logical reasoning practice bank: https://learn.novusstreamsolutions.com/cognitive-skills/logical-reasoning
- Deductive reasoning lesson: https://learn.novusstreamsolutions.com/aptitude/skills/deductive-reasoning/lesson
- Programming logic suite: https://learn.novusstreamsolutions.com/aptitude/tests/programming-logic
- Law, compliance and justice careers: https://learn.novusstreamsolutions.com/careers/families/law-compliance-and-justice
- Logical reasoning lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/logical-reasoning/lesson

Where this material comes from

- Worked items written for Novus Learn from standard introductory logic: conditional translation, the contrapositive, categorical syllogisms, and linear ordering under constraints. Every puzzle above was solved and checked for a unique answer before publication.
- Terminology checked against the public Wikipedia articles 'Modus ponens', 'Modus tollens', 'Contraposition', 'Affirming the consequent' and 'Syllogism'. Definitions only; all items 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.

Problem solving

Selecting methods, combining information, troubleshooting, and reaching practical solutions.

Problem solving is the construct that asks you to choose a method, not just execute one: to combine a table with a rule, decide whether an estimate settles the question or an exact calculation is needed, narrow a fault by halving the search space, and check the answer against the constraint that actually binds. It shows up across operations, logistics, manufacturing, technician and analyst selection, and in the situational sections of public-safety batteries. It is also the construct where the marking rewards a defensible route as much as a number. Everything you practise stays on this device unless you export it.

What you should be able to do after this lesson:

1. Combine a pricing or specification table with a written brief to reach an answer neither source contains on its own, and find the break-even point where the better option changes.
2. Narrow a fault on a chain of components by bisection, and state the worst-case number of tests before you begin.
3. Solve a working-backwards problem from a known end state, then verify by running the sequence forwards.
4. Decide whether an order-of-magnitude estimate settles a feasibility question or whether the margin is close enough to require exact arithmetic.
5. Identify the binding constraint in a resource-allocation problem and explain why a per-unit heuristic on the non-binding resource gives the wrong answer.
6. Separate a root cause from a symptom, and describe the test that would confirm the cause rather than merely correlate with it.

Worked examples and pitfalls

Two price structures and the distance where they cross: A delivery of 55 km. Courier A charges 8.00 base plus 0.45 per km. Courier B charges 20.00 flat for the first 40 km, then 0.90 per km beyond that. Which is cheaper? Courier A costs 8 plus 0.45 times 55, which is 8 plus 24.75, giving 32.75. Courier B costs 20 plus 0.90 times 15, which is 20 plus 13.50, giving 33.50. Courier A wins by 0.75. Now the question the item is really testing: is A always cheaper? At 45 km, A costs 8 plus 20.25 which is 28.25, while B costs 20 plus 4.50 which is 24.50. B wins comfortably. So the answer flips somewhere between, and finding where is one line of algebra. Above 40 km, B costs 20 plus 0.9 times distance minus 40, which simplifies to 0.9d minus 16. Set that equal to A's 8 plus 0.45d: 8 plus 0.45d equals 0.9d minus 16, so 24 equals 0.45d, so d is about 53.3 km. Below 53.3 km B is cheaper; above it A is. Check at the crossover: A is 8 plus 24 which is 32.00, and B is 48 minus 16 which is also 32.00. The general lesson is that whichever option has the lower per-unit rate always wins eventually, regardless of the base charges, and the base charges only decide where eventually starts. A single quoted distance never answers a which-is-cheaper question for a fleet.

Halving the search space beats walking the chain: Forty sensors sit on a single daisy-chained cable and exactly one connection is broken, cutting off everything downstream. How many tests do you need in the worst case? Walking the chain from one end and testing each sensor in turn takes up to 40 tests and 20 on average. Test the midpoint instead. If sensor 20 responds, the break is in the upper half and you have eliminated 20 candidates in one measurement; if it does not, the break is below and you have eliminated the other 20. Repeat on the surviving half: 40 becomes 20, then 10, then 5, then 3, then 2, then 1. Six tests,

worst case, because each test halves what is left and two to the power six is 64, comfortably more than 40. The saving grows as the problem grows: 1,000 candidates need only ten tests. Two conditions have to hold for bisection to be valid, and stating them is part of the answer. The fault must be monotone, meaning everything on one side of the break behaves differently from everything on the other; and a single test at any point must tell you which side you are on. When there are two independent breaks, or when a test is only meaningful at the ends, bisection does not apply and a different strategy is needed. Candidates who reach for bisection reflexively on a problem that fails those conditions lose more than they save.

Working backwards from the number you were given: A team started a quarter with an unknown budget. In week one it spent half of it. In week two it spent 400 of what remained. In week three it spent a third of what remained after that. It finished with 1,200. How much did it start with? Attempting this forwards means carrying an unknown through three operations. Backwards it is arithmetic. After week three, 1,200 is what is left having spent a third, so 1,200 represents two thirds of the week-three opening balance, which was therefore 1,800. Before week two's spend of 400, the balance was 1,800 plus 400, which is 2,200. That 2,200 is what remained after spending half, so the starting budget was 4,400. Verify forwards, always: 4,400 less half is 2,200; less 400 is 1,800; less a third of 1,800, which is 600, leaves 1,200. Correct. The mechanical rule is to invert each operation and apply the inversions in reverse order. The inverse of spending a third is dividing by two thirds, not multiplying by three, and that specific slip is the most common wrong answer in this family. Working backwards is the right tool whenever the end state is known exactly and the operations are individually invertible, which covers most budget, mixture and journey problems phrased as how much did it start with.

When an estimate is enough, and when it is not: A maintenance window is three hours. Two technicians must service 1,850 units, each taking about 4.5 minutes, plus a shared 20-minute setup and 15-minute teardown. Feasible? Estimate first: 1,850 units at 4.5 minutes is 8,325 technician-minutes; split between two people that is about 4,163 minutes, which is roughly 69 hours. The window is 3 hours. The answer is no by a factor of more than twenty, and no refinement of the setup and teardown figures could change it. Precision here would be wasted effort. Now the same problem with 90 units. Ninety at 4.5 minutes is 405 technician-minutes, halved to 202.5 minutes, plus 35 minutes of setup and teardown, giving 237.5 minutes. That is 3 hours and 57 and a half minutes against a 3-hour window. Still no, but only just, and now every assumption matters: whether the technicians can genuinely work in parallel, whether setup is shared or duplicated, whether 4.5 minutes is a mean or a best case. The judgement being assessed is knowing which regime you are in. If your rough figure misses the target by an order of magnitude, stop and answer. If it lands within a factor of two, the estimate has not settled anything and you must do the exact arithmetic and name your assumptions.

The binding constraint decides, and it is rarely the obvious one: A van carries at most 900 kg and at most 6 cubic metres. Pallet type X weighs 150 kg, occupies 0.8 cubic metres and is worth 200. Pallet type Y weighs 90 kg, occupies 1.4 cubic metres and is worth 260. What mix maximises value? The instinctive heuristic is value per kilogram: X gives 200 over 150, about 1.33, while Y gives 260 over 90, about 2.89, so load Y. That is wrong, and the reason is that weight is not what runs out. Value per cubic metre tells the opposite story: X gives 250 and Y gives about 186. Enumerate the feasible whole-pallet mixes. Six X uses 900 kg and 4.8 cubic metres, worth 1,200. Five X and one Y uses 840 kg and 5.4 cubic metres, worth 1,260. Four X and two Y uses 780 kg and exactly 6.0 cubic metres, worth 1,320. Three X and three Y needs 6.6 cubic metres and does not fit. Two X and three Y uses 570 kg and 5.8 cubic metres, worth 1,180. Four Y alone uses 5.6 cubic metres and is worth 1,040. The best mix is four X and two Y at 1,320. Look at what binds it: volume is used to the last cubic metre while 120 kg of payload goes unused. Once volume is identified as the binding constraint, X's superior value per cubic metre explains the X-heavy answer, and the spare weight explains why two Y still get on board. The transferable move is to compute the usage of every constraint at your proposed answer and see which one is exhausted; a per-unit ratio computed against a non-binding resource is worse than no heuristic at all.

Cause, symptom, and the test that tells them apart: A pump trips out twice a shift. The obvious fix is to reset it, which works for about four hours. Ask why once and you find the thermal overload relay is tripping. Ask again and the motor is running hot. Again and the bearing is running hot. Again and the grease has dried out. Again and the lubrication interval was set for a duty cycle far lighter than the pump has actually been running since the line was rebalanced last year. Each level supports a different fix: resetting the trip costs nothing and lasts four hours; replacing the relay costs a part and lasts until the bearing seizes; regreasing lasts weeks; changing the lubrication schedule to match the real duty cycle is the one that stops the fault recurring. Stop asking why when the next answer stops being something you can act on, or when it leaves the boundary of the system you can change. There is one discipline that separates this from storytelling. A correlation is a lead, not a cause: the fact that the trips started after the line was rebalanced is suggestive, not proof. The confirming test is to intervene and observe: restore the original duty cycle, or apply the corrected greasing interval to this pump and not to the identical one on the parallel line, and see which one trips. If a proposed cause cannot be tested by changing it, treat it as a hypothesis and say so rather than closing the investigation.

How to practise this skill

- Write the question you are actually being asked at the top of your working, in your own words. Problem-solving items usually supply more information than the question needs, and the commonest failure is a correct calculation of something nobody asked for.
- Before any comparison of two options, ask where they cross. One quoted quantity never settles which is cheaper or faster, and the break-even is usually a single line of algebra away.
- State the worst-case number of steps before starting a search or a fault trace. If bisection applies, say so and say why; if the fault is not monotone or a test only works at the ends, say that instead and pick a different method.
- Estimate before you compute, then decide explicitly which regime you are in. Missing the target by an order of magnitude ends the question; landing within a factor of two means the estimate proved nothing and the exact numbers are now compulsory.
- Finish every allocation or capacity problem by listing how much of each constraint your answer consumes. The constraint you exhaust is the one that explains the answer, and the one you do not is where a per-unit heuristic will mislead you.
- Always verify a backwards solution by running it forwards. Attempts stay on this device, so the extra minute costs nothing and catches the inverted-fraction error that makes this family's most attractive wrong answer.

Glossary

Break-even point: The value at which two options cost or take the same. Below it one option wins and above it the other does, which is why a single quoted case never answers a comparison question.

Binding constraint: The limit that is fully consumed by the chosen solution. Identifying it explains the answer and reveals which per-unit ratios are relevant and which are noise.

Slack: The unused portion of a constraint at a proposed solution. Spare capacity in one resource is what lets a mix include items that look inefficient on the binding resource.

Bisection: Halving a search space with each test. It reduces a search of n candidates to about log base two of n tests, but only where the fault is monotone along the chain.

Working backwards: Solving from a known end state by inverting each operation in reverse order. The reliable method whenever the final value is given and every step is individually invertible.

Order-of-magnitude estimate: A rough calculation intended to settle feasibility rather than produce a figure. It is decisive when the answer is off by a factor of ten and useless when the margin is within a factor of two.

Root cause: The condition whose removal stops the fault recurring, as opposed to a symptom whose treatment suppresses it temporarily. A candidate cause that cannot be tested by changing it remains a hypothesis.

Monotone fault: A fault where everything on one side of a break behaves differently from everything on the other. The precondition that makes bisection valid, and the assumption most often left unstated.

Study this next

- Problem solving skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/problem-solving
- Problem solving practice bank: https://learn.novusstreamsolutions.com/cognitive-skills/problem-solving
- Diagrammatic reasoning lesson: https://learn.novusstreamsolutions.com/aptitude/skills/diagrammatic-reasoning/lesson
- Maintenance technician suite: https://learn.novusstreamsolutions.com/aptitude/tests/maintenance-technician
- Supply chain planner suite: https://learn.novusstreamsolutions.com/aptitude/tests/supply-chain-planner
- Problem solving lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/problem-solving/lesson

Where this material comes from

- Every figure above was computed and checked for Novus Learn: the courier break-even at about 53.3 km, the six-test bisection bound for 40 candidates, the 4,400 starting budget verified forwards, and the four-X-two-Y loading enumerated across all feasible whole-pallet mixes.
- Terminology checked against the public Wikipedia articles 'Binary search algorithm', 'Break-even', 'Shadow price' and 'Root cause analysis'. Definitions only; every scenario 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.

Recommended preparation

- Practice processing speed: https://learn.novusstreamsolutions.com/aptitude/skills/processing-speed/lesson
- Practice verbal reasoning: https://learn.novusstreamsolutions.com/aptitude/skills/verbal-reasoning/lesson
- Strengthen 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/wonderlic-style-short-cognitive-familiarization/study-outline