

Software-developer reasoning: study guide

Technology, software, data, AI, and cybersecurity · suite apt-212-software-developer-reasoning · generated 2026-09-16T01:09:17.883Z

Detail	Value
Title	Software-developer reasoning: study guide
Generated	2026-09-16T01:09:17.883Z
Fixture/version	apt-212-software-developer-reasoning
Sector	Technology, software, data, AI, and cybersecurity
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 Software-developer reasoning

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-212-software-developer-reasoning&mode=guided>
- Start mini-test:
<https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-212-software-developer-reasoning&mode=mini>
- Start full simulation:
<https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-212-software-developer-reasoning&mode=full>

Skills covered, in full

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

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 $F \text{ implies } I$. 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: $\text{not-}I$, so $\text{not-}F$. 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 $F \text{ implies } I$ on one line and its contrapositive $\text{not-}I \text{ implies not-}F$ 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: $B \text{ implies } D$, $D \text{ implies } S$, $S \text{ implies } N$. 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: $\text{not-}N \text{ implies not-}S$, $\text{not-}S \text{ implies not-}D$, $\text{not-}D \text{ implies not-}B$. 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.

Technical reasoning

Applied technical principles, diagrams, tools, systems, measurements, and troubleshooting.

Technical reasoning is what sits above any single trade: reading a system diagram for what it actually does, isolating a fault by measurement instead of by guesswork, and taking the governing number off a drawing or a nameplate without importing assumptions. It is assessed in HVAC, instrumentation, mechatronics, process-operator and engineering-technician selection, and it is the construct that best predicts whether someone can diagnose an unfamiliar machine. Employers care about it because part-swapping is expensive and half-splitting is not. Practice sessions here stay on your device unless you choose to export them.

What you should be able to do after this lesson:

1. Trace a process or signal diagram end to end and state, for a given symptom, which components could physically cause it and which could not.
2. Isolate a fault by half-splitting a chain of stages, and say how many measurements a chain of a given length should take.
3. Convert a dimension with asymmetric tolerance into an acceptance window and decide whether a measured part passes, can be reworked, or is scrap.
4. Match a measuring instrument to a required resolution, and distinguish an instrument's resolution from its accuracy.
5. Describe a closed control loop as sensor, controller, setpoint, actuator and feedback, then design the one measurement that separates a sensor fault from an actuator fault.
6. Read the qualifier attached to a rated number (duty cycle, working load limit, working pressure, nominal versus maximum), and apply it correctly.

Worked examples and pitfalls

Half-splitting beats swapping parts: A conveyor will not stop when its photo-eye is blocked. The chain is: sensor, field cable, junction box, PLC input card, PLC program, output card, interposing relay, contactor. Eight places the signal can die. Swapping parts one at a time means four or five attempts on average, since

the culprit is equally likely to sit anywhere in the eight, and every attempt costs a part. Half-splitting starts in the middle instead: watch the PLC input LED while a colleague blocks the beam. If it toggles, the sensor, field cable, junction box and input card are all proven good in a single observation, and eight candidates become four. Force the output in the PLC and watch the contactor: if it pulls in, the output card, interposing relay and contactor are good as well, so the fault is in the program logic rather than the hardware. Three checks resolve eight stages, because each check halves the remaining suspects. The trap is starting at whichever end is easiest to reach, which resolves one stage per check instead of half of them. The second trap is the sentence 'I replaced the sensor and it still fails', that proves only that the sensor was not the fault, at the price of a part and an hour.

Tolerance: in spec, or scrap?: A drawing calls a shaft 25.00 mm with a tolerance of plus 0.05 and minus 0.10. That is an asymmetric tolerance, so the acceptance window runs from 24.90 mm to 25.05 mm and the nominal is not at its centre. A part measuring 24.92 mm is inside the window and passes, even though it is below nominal: the most common wrong rejection on this style of item, made by anyone who silently reads the tolerance as plus or minus 0.05. A part at 25.06 mm fails by 0.01 mm, but it fails oversize, so material can still be removed and it is rework rather than scrap. A part at 24.85 mm fails undersize and there is no recovering it. One more layer the better items include: if you took that 25.06 reading on a caliper with 0.02 mm resolution, the reading is at the very limit of what the instrument can resolve, and the honest next step is to re-measure with a micrometer before anyone scraps or reworks anything.

Reading a system diagram: what can actually cause this?: A tank fill line is drawn as supply, isolation valve V1, strainer, pump P1, check valve, control valve CV1, tank. A high-level switch LSH-1 is wired to close CV1. The reported symptom is that the tank overflowed. Work the path between the measurement and the element that stops flow: a CV1 that has stuck open, an LSH-1 that never actuated, and a broken wire in the LSH-1 loop are all consistent with the symptom. A blocked strainer is not: restricting the inlet reduces flow, and no amount of restriction causes an overflow. Nor is the check valve, whose function is to prevent reverse flow, not forward flow. Candidates pick the strainer because it is the component they know fouls in service, which is a memory of maintenance history rather than a reading of the diagram. The discipline that earns the mark is directional: a component can only be responsible if it lies on the causal path AND its failure mode pushes the system in the direction of the symptom.

Instrument choice: resolution is not accuracy: A steel rule resolves to roughly 0.5 mm. A vernier caliper marked 0.02 mm resolves to 0.02 mm. A 0 to 25 mm micrometer resolves 0.01 mm on the thimble, or 0.001 mm if it carries a vernier. A dial indicator reads 0.01 mm of relative movement but tells you nothing about absolute size without a reference. Asked to verify a 25.00 mm shaft with a tolerance of plus or minus 0.02 mm, the tolerance band is 0.04 mm wide: two divisions on that caliper, which is not enough to judge anything reliably. The workshop convention is that the instrument should resolve to about a tenth of the tolerance band, here 0.004 mm, so even the micrometer is marginal and comparison against gauge blocks is the defensible answer. The trap the item is built around is a digital display: showing four decimal places is a statement about resolution, not accuracy. An uncalibrated digital caliper will report 25.0000 mm with total confidence and be 0.03 mm out.

Closed loop: which element failed?: A room is meant to hold 21 degrees Celsius. A thermostat containing the sensor and the controller drives a valve on a radiator. The symptom: the room reaches 28 degrees Celsius and the valve stays open. Three explanations survive first inspection. The sensor reads low so the controller still believes the room is cold, the valve is mechanically jammed open, or the controller output has failed in the on state. One measurement separates them. Put an independent thermometer beside the thermostat. If the thermostat displays 17 degrees while the thermometer reads 28, the sensor is lying and everything downstream is behaving correctly. If the thermostat displays 28 and is still calling for heat, the sensor is fine and the fault is in the controller or the valve, which you then split by checking whether the valve actuator is being energised. The tempting non-answer is 'the room is too hot, so lower the setpoint'. That treats the symptom, and if the sensor reads seven degrees low the loop will simply settle seven degrees high again at

the new setpoint.

Nameplates: read the qualifier, not just the number: A welding machine is rated 200 A at 40 percent duty cycle over a ten-minute period. That means four minutes of arc time and six minutes of cooling in every ten, at the full 200 A. It does not mean 40 percent of 200 A, and it does not mean 40 percent of an hour. Both wrong readings feel entirely natural, which is why they make good distractors. Run the machine continuously at 200 A and the thermal cut-out will open. The same discipline transfers across the whole trade: a hoist's working load limit is not its breaking load, a hose's working pressure is not its burst pressure, a motor's service factor describes a short-term overload allowance and not a continuous rating, and a pump curve's flow figure is quoted at a stated head. Whenever an item hands you a number in a table or on a plate, underline the qualifier printed next to it before you calculate anything; the wrong options are usually built by dropping exactly one qualifier.

How to practise this skill

- Trace every diagram with a pencil from input to output and name each block as you pass it. Technical items punish skimming far harder than they punish slow arithmetic.
- Rehearse half-splitting on systems you already know (a home network that has dropped out, a car that will not crank), and count the checks. The habit transfers to the test intact.
- Underline the qualifier beside every number a question supplies: per hour, at 20 degrees Celsius, at 40 percent duty, nominal, maximum. That single mark-up defuses most distractors.
- Convert the whole question to one unit system in one pass before calculating, rather than converting each intermediate result and accumulating rounding.
- Train yourself to state what a measurement PROVES rather than what it reads. '12 V present at the coil' proves the supply and the whole upstream path; it says nothing about whether the coil itself is good.
- Keep a running list of the schematic symbols and component names you personally keep getting wrong, and drill only those. Five focused minutes beats another full untargeted set, and your attempt history stays local to this device so the list is yours alone.

Glossary

Half-split fault finding: Testing at the midpoint of a chain of stages so that each measurement eliminates half the remaining suspects. A chain of eight stages resolves in about three checks rather than four swaps.

Tolerance band: The distance between the upper and lower acceptance limits of a dimension. A tolerance written as plus 0.05 and minus 0.10 has a 0.15 mm band that is not centred on the nominal.

Resolution: The smallest change an instrument can display. A four-decimal readout has fine resolution and may still be inaccurate if the instrument is out of calibration.

Accuracy: How close a reading is to the true value, established by calibration against a traceable standard. Independent of resolution, and the property that decides whether a part passes.

Interlock: A condition wired or programmed to inhibit an action until it is satisfied, such as a guard-door switch that prevents a motor start while the door is open.

Closed-loop control: A control arrangement where a sensor measures the process, a controller compares that measurement to a setpoint, and an actuator drives the process until the difference closes.

Duty cycle: The proportion of a stated period during which equipment may operate at a stated output, for example 40 percent of a ten-minute period, after which it must cool.

Schematic versus pictorial diagram: A schematic shows function and connection logic with no regard to physical layout; a pictorial or exploded view shows physical arrangement and assembly order but hides the logic.

Root cause: The condition whose removal stops a failure recurring, as opposed to the symptom, which is

only what became visible.

Study this next

- Technical reasoning skill hub: <https://learn.novusstreamsolutions.com/aptitude/skills/technical-reasoning>
- HVAC and refrigeration suite: <https://learn.novusstreamsolutions.com/aptitude/tests/hvac-and-refrigeration>
- Instrumentation and control suite:
<https://learn.novusstreamsolutions.com/aptitude/tests/instrumentation-and-control>
- Engineering and technical design careers:
<https://learn.novusstreamsolutions.com/careers/families/engineering-and-technical-design>
- Encyclopedia search: troubleshooting: <https://learn.novusstreamsolutions.com/search?q=Troubleshooting>
- Technical reasoning lesson on Novus Learn:
<https://learn.novusstreamsolutions.com/aptitude/skills/technical-reasoning/lesson>

Where this material comes from

- Diagnostic and metrology examples written for Novus Learn from general maintenance practice: binary-search fault isolation, asymmetric dimensional tolerance, and the ten-to-one instrument selection convention.
- Terminology checked against the public Wikipedia articles 'Troubleshooting', 'Engineering tolerance', 'Accuracy and precision' and 'Control loop'. 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. Ratings and limits described here are illustrative and never override the equipment documentation in front of you.

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.

Abstract reasoning

Inferring rules from shapes, symbols, matrices, and non-verbal patterns.

Abstract reasoning is rule-finding stripped of language and content: you are given shapes, shading, counts and positions, and asked which rule generates them. Because nothing in the item depends on vocabulary or schooling, it is one of the most widely used constructs in graduate screening, general aptitude batteries and matrix-style tests, and it is the section candidates most often describe as unfair. Usually because they were searching for one rule where the figure encodes three independent ones. The skill is systematic attribute scanning, not flashes of insight. Practice 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. Scan a figure against a fixed attribute checklist (count, shape, size, shading, orientation, position, line style, symmetry) instead of searching for a rule at random.
2. Read a three-by-three matrix along rows and columns separately and combine two independent progressions to produce the missing cell.
3. Distinguish the three combination rules that look alike on paper (union, intersection and exclusive-or superposition) by testing each against a complete row.
4. Track several attributes cycling at different periods within one sequence, and predict a frame that no single attribute determines.
5. Tell a rotation from a reflection by checking whether the clockwise order of three marked features is preserved or reversed.
6. Use the answer options as evidence, eliminating on one attribute at a time rather than judging each option as a whole.

Worked examples and pitfalls

A matrix where the rule is arithmetic on side counts: A three-by-three grid of polygons. The top row runs triangle, square, pentagon. The middle row runs square, pentagon, hexagon. The bottom row runs pentagon, hexagon, and an empty cell. Count sides rather than naming shapes and the structure appears immediately: 3, 4, 5 across the top; 4, 5, 6 across the middle; 5, 6, and the answer across the bottom. Sides increase by one along each row AND by one down each column, so the missing figure has seven sides: a heptagon. Writing the numbers into the cells is the whole technique, and it generalises: whenever a matrix contains countable things, replace every cell with its count before you look for anything else, because a numeric grid makes a progression visible that shapes hide. The distractor set on an item like this is instructive. It will contain a hexagon (correct rule, one step short), an octagon (right idea, overshot), a heptagon with the wrong shading, and a heptagon rotated. Three of those four are only wrong on a second attribute, which is why the answer must be checked on every attribute before you commit rather than on the one that solved the puzzle.

Union, intersection and exclusive-or look identical until you test them: Many matrices build the third cell of each row by combining the first two. Take a row where cell one contains a dot in the top-left corner and a cross in the centre, and cell two contains a cross in the centre and a dot in the bottom-right corner. Cell three contains a dot in the top-left and a dot in the bottom-right, with no cross. That is exclusive-or superposition:

elements present in exactly one of the two inputs survive, and elements present in both cancel. If cell three had contained both dots AND the cross, the rule would be union. Everything from both inputs is kept. If it had contained only the cross, the rule would be intersection: only what appears in both survives. All three rules produce plausible-looking figures, so guessing from one row is unreliable. The reliable procedure is to identify a row where the two inputs share at least one element and differ in at least one other, because that is the only configuration where union, intersection and exclusive-or give three different answers. Confirm the rule there, then apply it to the row with the missing cell. Items in this family also hide a fourth variant, where shared elements survive but change colour; catching that one requires checking shading as a separate attribute rather than treating a black cross and a white cross as the same object.

Three attributes on three different clocks: A six-frame sequence. Frame 1: an arrow pointing up, unshaded, with one internal bar. Frame 2: arrow pointing right, shaded, two bars. Frame 3: arrow pointing down, unshaded, three bars. Frame 4: arrow pointing left, shaded, one bar. Frame 5: arrow pointing up, unshaded, two bars. What is frame 6? Take the attributes one at a time. Direction rotates ninety degrees clockwise every frame, a cycle of four, so after up comes right. Shading alternates, a cycle of two, so after unshaded comes shaded. The bar count runs 1, 2, 3, 1, 2, a cycle of three, so after two comes three. Frame 6 is a right-pointing shaded arrow with three bars. The reason this item defeats people is that the whole figure does not repeat until frame 13: the individual attributes come back round every four, two and three frames, but they only come back into phase together after twelve, so the sequence as printed looks like it has no period at all. Treat each attribute as an independent counter with its own cycle length and the difficulty evaporates. The practical habit is to rule three columns on your paper (direction, shading, count) fill them in for every given frame, and extend each column separately before you look at a single answer option.

Odd one out, where the obvious difference is the decoy: Five figures. (a) A square with one diagonal drawn. (b) A triangle with a line from the apex to the base. (c) A pentagon with a line joining two non-adjacent vertices. (d) A hexagon with two lines crossing inside it. (e) A circle with a chord. Which is the odd one out? The difference that jumps out is that (e) is curved and the others are straight-edged, and that is the decoy. It is a real difference, but it is not the one the item is built on. Count enclosed regions instead. Figures (a), (b), (c) and (e) are each divided into exactly two regions by a single line. Figure (d) has two crossing lines and is divided into more than two. The rule is region count, and (d) is the answer. Odd-one-out items reliably plant a salient irrelevant difference (one figure is the only curved one, or the only shaded one, or the only one with a right angle), and hide the operative rule in something countable. The defence is procedural: before choosing, write the value of at least three attributes for all five figures, for example number of enclosed regions, number of straight edges, number of line intersections. The correct answer is the figure that stands alone on exactly one row of that table while the others agree, and if two rows both isolate a figure, the item is ambiguous and you should prefer the countable rule over the categorical one.

Rotation or reflection, settled by clockwise order: A flag-like figure carries three distinguishable marks: a red band, a blue band and a yellow band, which read red, blue, yellow going clockwise around its centre. You are shown a candidate answer figure that also has three bands. Is it a rotation of the original or a reflection of it? Rotating a flat figure in the plane never changes the clockwise order of its features, so any genuine rotation still reads red, blue, yellow clockwise, from whatever starting point you choose. A mirror reflection reverses that order, so a reflected figure reads red, yellow, blue clockwise. Read the cyclic order and the question is answered without imagining any motion at all. This matters because reflections are the standard trap in rotation items: the reflected option looks exactly as plausible as the rotated one, and no amount of mental turning will produce it. The same test works on letters, no rotation of R in the plane produces a backwards R, and on three-dimensional figures, where the equivalent check is whether a right-handed set of three edges at a corner stays right-handed. Note the one honest caveat: a figure with a mirror line of its own is unchanged by reflection, so this test only decides the question for asymmetric figures, which is exactly why item writers use asymmetric ones.

Two dots on the perimeter, moving at different rates: A three-by-three grid where only the eight perimeter

cells are used. Number them clockwise starting at the top-left corner: 0 top-left, 1 top-middle, 2 top-right, 3 right-middle, 4 bottom-right, 5 bottom-middle, 6 bottom-left, 7 left-middle. Frame 1 has both dots at cell 0. Frame 2 has one dot at cell 2 and the other at cell 7. Frame 3 has them at cells 4 and 6. Frame 4 has them at cells 6 and 5. What is frame 5? Track them separately. The first dot goes 0, 2, 4, 6, two cells clockwise each frame, so it lands on 8, which wraps to cell 0, the top-left corner. The second goes 0, 7, 6, 5, one cell anticlockwise each frame, so it lands on cell 4, the bottom-right corner. Frame 5 has one dot top-left and one bottom-right. Two counting errors dominate this family. The first is treating the perimeter as nine positions because the grid has nine cells; the centre is not on the path and the perimeter is a loop of eight. The second is assuming both markers share a direction or a speed, which is precisely the assumption the item is testing. When markers move at different rates, the frames where they coincide or sit adjacent are coincidences of the arithmetic, not part of the rule, and reading meaning into them is how a solvable item becomes impossible.

How to practise this skill

- Keep a written attribute checklist beside you until it is automatic: count, shape, size, shading, orientation, position, line style, symmetry. Scanning a fixed list is faster than free-associating, and it is what separates candidates who finish the section from those who stall on item four.
- Replace every countable cell with a number before looking for a rule. A matrix of shapes is hard to read; the same matrix written as 3, 4, 5 over 4, 5, 6 solves itself.
- Read matrices along rows and down columns as two separate passes. A large share of items encode one progression in each direction, and a rule that only works across rows is usually half the answer.
- Use the options as data. Sort them by one attribute, discard the group that cannot be right, then sort the survivors by the next. Eliminating on single attributes is far more reliable under time pressure than judging whole figures.
- Check the cyclic order of three features before accepting any rotation option, and remember that a reflected asymmetric figure can never be produced by turning the original in the plane.
- Practise untimed until you can state the rule in one sentence for every item you answer, including the ones you got right by feel. Attempts are stored on this device only, so an unhurried pass where you narrate each rule aloud costs nothing and is what actually builds the scanning habit.

Glossary

Attribute: A single independently varying property of a figure, such as shading, count or orientation. Most difficult items vary three or more attributes at once, each on its own cycle.

Matrix item: A grid of figures, usually three by three, with one cell missing. Rules typically run along rows, down columns, or both at once.

Superposition: A rule that builds one figure by combining two others. The three common variants keep the union of elements, the intersection, or the elements appearing in exactly one input.

Exclusive-or rule: The superposition variant in which elements common to both inputs cancel and only the unshared elements survive. Distinguished from union and intersection by testing a row whose inputs both share and differ in elements.

Distractor: An incorrect answer option built to match the correct one on the attribute that solved the item while differing on another. It is why a figure must be checked on every attribute before selection.

Chirality: The handedness of an asymmetric figure. Rotation preserves it and reflection reverses it, so the clockwise order of three marked features tells the two transformations apart.

Cycle length: The number of frames after which an attribute repeats. When several attributes have different cycle lengths, the whole figure only repeats after their least common multiple.

Enclosed region: An area fully bounded by lines within a figure. It is one of the most common hidden rules in odd-one-out items because it is countable and visually unobtrusive.

Study this next

- Abstract reasoning skill hub: <https://learn.novusstreamsolutions.com/aptitude/skills/abstract-reasoning>
- Abstract reasoning practice bank:
<https://learn.novusstreamsolutions.com/cognitive-skills/abstract-reasoning>
- Inductive reasoning lesson:
<https://learn.novusstreamsolutions.com/aptitude/skills/inductive-reasoning/lesson>
- Diagrammatic reasoning lesson:
<https://learn.novusstreamsolutions.com/aptitude/skills/diagrammatic-reasoning/lesson>
- General academic aptitude suite:
<https://learn.novusstreamsolutions.com/aptitude/tests/general-academic-aptitude>
- Abstract reasoning lesson on Novus Learn:
<https://learn.novusstreamsolutions.com/aptitude/skills/abstract-reasoning/lesson>

Where this material comes from

- Every figure, matrix and sequence described above was constructed and verified for Novus Learn. Side counts, superposition outcomes, attribute cycle lengths and perimeter positions were each checked by hand.
- Matrix-style non-verbal items are a long-established public assessment format; this lesson describes the format in general terms and reproduces no item from any published test.
- Terminology checked against the public Wikipedia articles 'Chirality', 'Exclusive or' and 'Least common multiple'. Definitions only.
- 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 logical reasoning: <https://learn.novusstreamsolutions.com/aptitude/skills/logical-reasoning/lesson>
- Build technical reasoning:
<https://learn.novusstreamsolutions.com/aptitude/skills/technical-reasoning/lesson>
- Practice situational judgement:
<https://learn.novusstreamsolutions.com/aptitude/skills/situational-judgement/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/software-developer-reasoning/study-outline>