Game-based assessment familiarization: study guide

Commercial assessment-provider familiarization · suite apt-366-game-based-assessment-familiarization · generated 2026-09-15T15:42:02.822Z

Detail	Value
Title	Game-based assessment familiarization: study guide
Generated	2026-09-15T15:42:02.822Z
Fixture/version	apt-366-game-based-assessment-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 Game-based assessment 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-366-game-based-assessment-familiarization&mode=guided
- Start mini-test: https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-366-game-based-assessment-familiarization&mode=mini
- Start full simulation: https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-366-game-based-assessment-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.

Working memory

Holding, updating, and manipulating short sequences, rules, or instructions.

Working memory is the small, fast, easily disrupted workspace you use to hold a few items while you do something to them: reversing a digit string, keeping a running total, tracking which rule is currently in force. It is not general memory and it is not intelligence; it is capacity plus the ability to update and resist interference. Assessment batteries probe it with backward span, letter-number sequencing, n-back and running-memory tasks, and game-based screens lean on it heavily. At work it shows up whenever you hold a customer's account number while navigating three screens, or carry a partial calculation while someone talks at you. Everything you practise here is device-local: no account, and nothing leaves the device unless you export it.

What you should be able to do after this lesson:

1. Reverse and re-order a presented sequence without writing it down, using chunking to keep the load inside span rather than fighting it item by item.
2. Track an n-back stream and correctly reject the 1-back lure that catches most first-time candidates.
3. Hold and update a rule that changes mid-item, and state which stored value is now stale.
4. Decompose multi-step mental arithmetic so that only one partial result is ever in play at a time.
5. Explain why acoustically similar and multi-syllable material eats span faster than the item count alone predicts.
6. State honestly what working-memory practice does and does not transfer to, so preparation targets task familiarity rather than a promised capacity increase.

Worked examples and pitfalls

Backward span: chunk first, then reverse the chunks: Read this once and answer without looking back: 7 2 9 4 1 6 3 8, reported backwards. Item-by-item reversal fails around six digits because you are re-reading a decaying trace eight times. Chunk instead. Pair them at input: 72, 94, 16, 38 - four items, comfortably inside span. Now reverse the chunk order and the pair inside each: 38 becomes 83, 16 becomes 61, 94 becomes 49, 72 becomes 27. Output 8 3 6 1 4 9 2 7. Check it against the original read left to right and it matches. The trap is a partial reversal: candidates reverse the chunks but forget to reverse inside them and answer 3 8 1 6 9 4 7 2, which looks right because every digit is present and the first digit is plausibly near the end of the original. That answer is tempting precisely because the gist check - same digits, roughly backwards - passes. Only a position check catches it, so verify the FIRST digit you output equals the LAST digit you heard, then stop worrying and recite.

N-back: the lure sits one position away: A 2-back task on the letter stream T, L, T, K, R, K, R, R, M. You press only when the current letter matches the one two positions earlier. Walk it. Position 3 is T against position 1, T - target. Position 4 is K against position 2, L - no. Position 5 is R against position 3, T - no. Position 6 is K against position 4, K - target. Position 7 is R against position 5, R - target. Position 8 is R against position 6, K - no, and this is the whole point of the item: position 8 matches position 7, an immediate repeat that feels overwhelmingly like a hit. It is a 1-back lure. Position 9 is M against position 7, R - no. Three targets: positions 3, 6 and 7. Familiarity is the wrong signal here, because the recent repeat is the most familiar thing in the stream; the task requires position, not familiarity. The practical defence is to run a strict two-slot window and physically discard the older slot as each new letter arrives, rather than letting a queue of five or six letters build up.

Letter-number sequencing: two sorts, one pass: Presented: R 4 B 9 K 1 M 7. The instruction is to report the numbers in ascending order first, then the letters in alphabetical order. Splitting at input is what makes this tractable: as each character arrives, drop it into one of two mental piles instead of storing the presented order at all. Numbers pile: 4, 9, 1, 7, sorted to 1 4 7 9. Letters pile: R, B, K, M, sorted to B K M R. Answer: 1 4 7 9 B K M R. Candidates who store all eight characters in the order given then try to sort afterwards are holding eight items and running two sorts on a decaying trace, which is roughly double the load for no benefit. The common wrong answer preserves presented order inside each pile - 4 9 1 7 B K M R - because the sort step was dropped under load while the split step survived. If you can only do one thing well under pressure, sort at input and let presentation order go.

Sequential arithmetic: never hold two partial results: Instructions given aloud, one line at a time: start at 17, multiply by 4, subtract 12, divide by 8, add 19. Run it strictly in order. 17 x 4 = 68. 68 - 12 = 56. 56 / 8 = 7. 7 + 19 = 26. The answer is 26. Two traps live here. First, order of operations: this is a sequence of instructions, not an algebraic expression, so nobody does the division before the subtraction, and a candidate who silently applies precedence rules gets a different and confidently wrong number. Second, load: the discipline is that after each line, only ONE number exists. The moment you are carrying both 68 and 12 as separate live values you have doubled the load for no reason. The same discipline turns awkward multiplications into easy ones by rounding to a friendly anchor and correcting once: 68 x 7 becomes 70 x 7 = 490, minus 2 x 7 = 14, giving 476. One anchor, one correction, one live value at a time.

Rule updating: the stale value is what gets marked wrong: The standing rule is to sort a support queue by priority, High before Low. An exception is added: if the queue name begins with a vowel, sort by age instead, oldest first. The queue is Onboarding, holding ticket A (High, 2 days old), ticket B (Low, 9 days) and ticket C (High, 5 days). Onboarding begins with O, so the exception is live and priority is now irrelevant. Sorted by age, oldest first: B, C, A. The tempting answer is A, C, B - the two High tickets first, oldest of them leading - because the priority labels are the most visually salient thing on the screen and the exception was a single clause read several seconds ago. This is what working-memory items actually test: not whether you can sort, but whether the currently active rule successfully overwrote the default. The habit that fixes it is to say the live rule out loud, in one short phrase, immediately before you act - 'vowel, so age, oldest first' - which forces a retrieval of the rule rather than a retrieval of the habit.

Why some lists are heavier than others of the same length: Span is not measured in items alone. Hold B, C, D, G, P, T, V - seven letters that all rhyme in English. Now hold F, K, L, Q, R, Y, W - also seven letters, but acoustically distinct. Almost everyone recalls the second set more accurately, and the errors on the first set are confusions within the rhyming set rather than lost items. That is the phonological similarity effect, reported by Conrad and Hull in 1964, and it tells you the verbal store holds something sound-like. Length matters the same way: five short country names - Chad, Burma, Greece, Cuba, Malta - are easier to hold than five long ones such as Nicaragua, Afghanistan, Venezuela, Madagascar and Yugoslavia, an effect Baddeley, Thomson and Buchanan reported in 1975 and tied to how much can be rehearsed in about two seconds. The practical consequence for a test: when material is confusable, stop rehearsing it as sound and re-encode it visually or spatially - imagine the digits on a keypad, or the letters at clock positions - because that moves the load to a different store instead of stacking it in the one that is already saturated.

How to practise this skill

- Chunk at input, not at output. Pairing digits or grouping letters as they arrive costs nothing and turns an eight-item load into a four-item one; trying to chunk a trace you are already struggling to hold does not work.
- Train updating separately from holding. Backward span trains holding; n-back and running-memory tasks train the discard step, and it is the discard step that fails first under time pressure.
- Deliberately practise with a distractor running - count backwards by threes, or say 'the, the, the' while you rehearse. That is articulatory suppression, and it is a fair simulation of an assessment centre where an invigilator is talking.
- When a rule can change mid-item, re-state the live rule in three or four words immediately before answering. Most lost marks on rule-updating items are the previous rule applied fluently, not confusion about what to do.
- Do not chase a promised capacity increase. Meta-analytic reviews of working-memory training, including Melby-Lervag and Hulme (2013), have found reliable improvement on the trained task and its close relatives but little evidence of transfer to general reasoning - so practise for format familiarity and lower anxiety, which are real gains, and treat anything else as a bonus.
- Run untimed until the strategy is automatic, then add the clock. Attempts are stored on this device only, so an ugly first pass through n-back costs you nothing.

Glossary

Phonological loop: The component of Baddeley and Hitch's 1974 model that holds speech-like material for a couple of seconds and refreshes it by silent rehearsal. Its limits produce the phonological similarity and word-length effects.

Visuospatial sketchpad: The parallel store for visual and spatial material - shapes, layouts, routes. Because it is separate from the verbal store, re-encoding a confusable list as a spatial image genuinely offloads it.

Central executive: The control component that allocates attention, switches between tasks and keeps the currently active rule in force. Rule-updating items load this far more than they load storage.

Chunk: A group of items bound into a single unit by meaning or familiarity. Capacity is counted in chunks, not raw items, which is why 1-9-4-5 costs one slot to someone who reads it as a year.

Span: The longest sequence reproduced reliably. Forward span measures storage; backward and re-ordered span add manipulation and are consistently shorter for the same person.

N-back: A continuous task where you respond when the current item matches the one n positions earlier. It stresses updating and discarding rather than storage, and its characteristic error is the immediate-repeat lure.

Articulatory suppression: Speaking irrelevant sounds while trying to hold verbal material. It blocks rehearsal and sharply cuts verbal span, which is why noisy rooms and talkative invigilators are not a trivial complaint.

Proactive interference: Older material intruding on new material of the same kind. It is why the fifth practice trial of a digit-span block is harder than the first, and why varying the material between blocks helps.

Study this next

- Working memory skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/working-memory
- Cognitive Skills Lab: working memory practice: https://learn.novusstreamsolutions.com/cognitive-skills/working-memory
- Game-based assessment familiarization suite:

https://learn.novusstreamsolutions.com/aptitude/tests/game-based-assessment-familiarization
- Multitasking and monitoring lesson: https://learn.novusstreamsolutions.com/aptitude/skills/multitasking-monitoring/lesson
- Processing speed lesson: https://learn.novusstreamsolutions.com/aptitude/skills/processing-speed/lesson
- Working memory lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/working-memory/lesson

Where this material comes from

- Worked sequences, n-back streams and rule-updating items written for Novus Learn. Every string and number above is original; no published test item is reproduced.
- Model terminology follows the public account of Baddeley and Hitch's multicomponent working-memory model (1974) and the capacity debate between Miller (1956) and Cowan (2001). Concepts only.
- The phonological similarity and word-length effects are attributed to Conrad and Hull (1964) and Baddeley, Thomson and Buchanan (1975); the transfer caveat reflects the meta-analytic review by Melby-Lervag and Hulme (2013). No effect sizes are quoted.
- 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 not a clinical or diagnostic measure.

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

Attention and concentration

Sustained focus, selective attention, and accuracy under time pressure.

Attention and concentration is the skill of still noticing on minute forty of a checking block as reliably as you did on minute two, and of pointing the noticing at the right thing when two things compete. It has three distinguishable parts - selective attention, sustained attention or vigilance, and divided attention - and assessments load them differently: cancellation and checking tasks load vigilance, conflict tasks load selection, and dispatch-style monitoring loads division. It is the construct that decides whether a records clerk, a dispatcher, an air-side controller or a quality inspector catches the one wrong digit in a shift. Practice here is device-local, with no account and nothing uploaded.

What you should be able to do after this lesson:

1. Count occurrences of a target in a dense string without double-counting or losing your place, using a fixed scan path rather than free looking.
2. Separate hits, misses and false alarms in your own results and work out whether you are set too eager or too cautious for the scoring rule in force.
3. Predict where in a long block your error rate will rise, and schedule micro-resets against that curve instead of pushing through it.
4. Recognise a transposition error, the class that survives a same-characters gist check and is therefore missed most often.
5. Explain why an automatic process such as reading interferes with a controlled one, and use that to anticipate which distractors will actually cost you time.
6. Run a two-stream monitoring task by alternating on a fixed cadence rather than attempting genuine simultaneity.

Worked examples and pitfalls

A cancellation count you can actually check: Count every 7 in this row: 4 7 1 7 7 3 9 7 2 8 7 5 7 6 0 7. Working left to right and tapping once per hit: hits at the second, fourth, fifth, eighth, eleventh, thirteenth and sixteenth positions. That is seven sevens. Two error modes produce nearly all the wrong answers. The first is the adjacent pair - the 7 7 at positions four and five gets counted once, because the eye takes a repeated character as one perceptual object. The second is losing the place after the 9, where the visually similar 9 and 7 force a moment of re-checking and the scan restarts a character early, producing an over-count of eight. The defence for both is a fixed scan path with a physical anchor: a fingertip or cursor moving one character at a time, never jumping back. Re-scanning to check is the thing that creates the double-count, so if you must verify, verify by counting a second time from the RIGHT-hand end and comparing totals, never by re-reading part of the row.

Hits, misses and false alarms: the eager candidate loses: A checking batch contains 300 records, 40 of which are genuinely faulty. Candidate A flags 46 records and 36 of them are truly faulty. Hits 36, misses 40 minus 36 equals 4, false alarms 46 minus 36 equals 10. Candidate B flags 38 and 34 are truly faulty: hits 34, misses 6, false alarms 4. On hit rate alone A wins, 36 out of 40 which is 90 percent against B's 34 out of 40 which is 85 percent. Now apply the scoring rule that most checking tasks actually use, where a false alarm cancels a hit: A scores 36 minus 10 equals 26, B scores 34 minus 4 equals 30. B wins by four despite catching two fewer faults. This is why 'flag anything that looks odd' is bad advice on a scored checking task, and why the first thing to read on a checking item is whether wrong flags are penalised. Your response criterion - how much evidence you demand before flagging - is adjustable, and it should be set from the scoring rule, not from your temperament.

The transposition that passes every gist check: Compare these two lines and decide whether they match. Invoice 4820-7391-06. Invoice 4820-7931-06. They do not: the middle group reads 7391 in the first and 7931 in the second, with the 3 and the 9 swapped. Transpositions are the most-missed error class in record checking for a structural reason - the character SET is identical, the length is identical, the first and last characters of the group are identical, so every fast check the visual system runs comes back clean. Substitutions and omissions change the character inventory and get caught; transpositions do not. Two habits raise the catch rate. Read digits in fixed groups of two rather than as a whole number, so 73-91 against 79-31 becomes a mismatch at the first group instead of a subtle difference somewhere in a four-digit blur. And check groups in a deliberately non-natural order - last group, first group, middle group - because reading left to right lets the confirmation you built at the start carry you through the middle, which is exactly where the swap is usually planted.

Conflict: why reading fights you: The classic demonstration is Stroop's, published in 1935: the word RED printed in blue ink, with the instruction to name the ink colour. Naming takes measurably longer, and errors go up, because reading a familiar word is automatic and cannot be switched off, so the automatic response has to be suppressed before the controlled one can be produced. The same conflict has a numeric version you can test on yourself in a second: how many characters are in the string 4 4 4? The answer is three, and the digit 4 pulls at you the entire way. In an assessment this appears wherever the salient feature and the asked-for feature come apart - a chart where the tallest bar is not the answer to the question, a form where the highlighted field is not the one being verified, a row where the bold total is not what the stem requested. The practical move is to name the target feature out loud before you look - 'ink colour', 'character count', 'the value for March' - because pre-loading the target biases selection before the automatic reading response gets a chance to win.

Where the errors actually appear in a 45-minute block: Errors in a long checking block are not spread evenly. Mackworth's 1948 clock-watching study established the pattern that gives the effect its name: detection declines over a prolonged watch, with the sharpest deterioration early rather than at the very end. Practically, on a 45-minute self-timed checking block, expect your per-minute error rate in minutes 20 to 45 to run visibly above minutes 1 to 20 even though nothing about the material changed, and expect the subjective sense of effort to lag the actual decline, so it will not feel like you are getting worse. Two things work against it. Break the block into three fifteen-minute segments with a ten-second reset between them - look away, unfocus, breathe out - which costs thirty seconds of a 45-minute block, about one percent of the time, and buys back more than that in caught errors. And score your practice by segment rather than as one number, because a single overall accuracy figure hides exactly the information you need: whether your problem is skill, which shows as flat error rate, or endurance, which shows as a rising one.

Two streams: alternate on a cadence, do not try to merge: A dispatch-style monitoring task: keep a running total of the numbers announced on channel one while watching channel two for the code word AMBER. Genuine simultaneity is not available - the two tasks compete for the same control resource - so the choice is not whether to alternate but whether to alternate deliberately or accidentally. Deliberate looks like this: fix the arithmetic to a rhythm, updating the total only at each announcement and holding a single number between updates, which frees the gaps for channel two. If the total is 34 and the next announcement is 7, you spend under a second reaching 41 and then you are free again. Accidental looks like re-deriving the running total from the beginning because you did not trust it, which locks up the whole window and is when AMBER goes past unheard. The measurable failure of divided attention is almost never a failure to hear the target; it is a failure to have any spare capacity at the moment it arrived. The related phenomenon worth knowing is inattentional blindness, illustrated by Simons and Chabris in 1999: an unexpected and perfectly visible event is missed entirely when attention is committed to a counting task.

How to practise this skill

- Fix your scan path before you start, and never re-scan backwards to double-check. Backward re-scanning is what produces both double-counts and lost places; if you need to verify a count, run it again from the other end and compare.
- Read the scoring rule before the first item. Whether false alarms are penalised changes the correct strategy completely, and a criterion set for the wrong rule costs more marks than slow reading does.
- Check numeric strings in fixed groups of two or three and in a deliberately shuffled group order. Transpositions are the errors that survive whole-string comparison, and they are the ones planted on purpose.
- Time your practice in segments and log accuracy per segment, not per session. A flat error rate means you need speed; a rising one means you need endurance and breaks, and the two problems have opposite fixes.
- Name the target feature out loud before each conflict item. Pre-loading the relevant dimension is the cheapest available defence against the salient-but-wrong answer.
- Practise with an actual distractor present rather than in silence. A test hall has movement, coughing and page-turning, and attention practice in perfect quiet trains a condition you will not meet.

Glossary

Selective attention: Prioritising one source or feature while suppressing competing ones. It is what fails when the visually salient element of a display captures the response instead of the requested one.

Sustained attention (vigilance): Maintaining detection performance over a long, low-event period. It is the capacity that checking, monitoring and inspection tasks are really sampling.

Vigilance decrement: The decline in detection performance over a prolonged watch, typically steepest early in the block. Mackworth's 1948 clock test is the standard public reference.

Divided attention: Handling two streams that compete for control. Performance is best modelled as fast alternation with a switch cost, not as genuine parallelism.

Hit, miss, false alarm, correct rejection: The four outcomes of any detection decision: flagging a real fault, missing one, flagging a clean record, and correctly leaving a clean record alone. Any honest accuracy claim needs at least the first three.

Response criterion: How much evidence you require before flagging. Shifting it trades misses against false alarms without changing your underlying sensitivity, which is why it must be set from the scoring rule.

Stroop interference: The slowing that occurs when an automatic response, such as reading a word, conflicts with the requested response, such as naming its ink colour. Named for Stroop's 1935 experiments.

Inattentional blindness: Failing to notice a fully visible, unexpected event because attention is committed elsewhere - the effect Simons and Chabris demonstrated in 1999 with a counting task.

Study this next

- Attention and concentration skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/attention-concentration
- Cognitive Skills Lab: attention practice: https://learn.novusstreamsolutions.com/cognitive-skills/attention-concentration
- Emergency dispatch and 911 communications suite: https://learn.novusstreamsolutions.com/aptitude/tests/emergency-dispatch-and-911-communications
- Error detection lesson: https://learn.novusstreamsolutions.com/aptitude/skills/error-detection/lesson
- Perceptual speed lesson: https://learn.novusstreamsolutions.com/aptitude/skills/perceptual-speed/lesson
- Attention and concentration lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/attention-concentration/lesson

Where this material comes from

- Character strings, invoice comparisons, batch counts and monitoring scenarios above are written for Novus Learn. All figures are original and no published test item is reproduced.
- Terminology and directions of effect follow widely published work: Mackworth (1948) on the vigilance decrement, Stroop (1935) on response conflict, Simons and Chabris (1999) on inattentional blindness, and the standard hit/miss/false-alarm framing from signal detection theory. Concepts only; no result figures are attributed to those studies.
- 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 not a clinical, diagnostic or attention-disorder screening measure.

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

Multitasking and monitoring

Tracking multiple streams, priorities, alerts, and changing conditions.

Practise monitoring several information streams without treating every change as equally urgent. Define normal ranges and action thresholds, keep a reliable scan, and protect handoffs when attention must switch.

What you should be able to do after this lesson:

1. Distinguish a routine update, a trend requiring closer observation, and an alert requiring action.
2. Use a consistent scan cycle and external notes so unattended streams are not forgotten.
3. Prioritize by consequence and threshold while communicating handoffs during interruptions.

Worked examples and pitfalls

Worked scenario: alert, call, and routine log: A dashboard crosses a defined critical threshold while a caller asks for a non-urgent update and a routine log is due later. Acknowledge the critical alert, stabilize or escalate it under the procedure, tell the caller when ownership will return, and leave a time-stamped note before resuming. Finishing the easy log first feels productive but ignores consequence; trying to answer everything at once weakens all three tasks.

Scan–detect–decide–record: Set a repeatable scan order, compare each stream with its expected range, act only when a threshold or trend requires it, and record the state before switching. This reduces memory load and makes the next handoff recoverable.

How to practise this skill

- Learn the difference between an alert threshold and normal variation before adding time pressure.
- After an interruption, restart from the last recorded state instead of trusting memory alone.
- Escalate early when concurrent critical demands exceed one person's capacity; overload is itself a risk signal.

Glossary

Situation awareness: Understanding the current state, meaningful changes, and likely near-term consequences across monitored streams.

Threshold: A defined value or condition that changes the required level of attention or action.

Handoff: A deliberate transfer of current state, outstanding risk, and next responsibility to another person or future self.

Study this next

- Multitasking: https://learn.novusstreamsolutions.com/search?q=Multitasking
- Situation awareness: https://learn.novusstreamsolutions.com/search?q=Situation%20awareness
- Monitoring: https://learn.novusstreamsolutions.com/search?q=Monitoring
- Multitasking and monitoring lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/multitasking-monitoring/lesson

Where this material comes from

- Novus Learn original monitoring, dispatch, control-room, and contact-centre suite scenarios.
- Novus educational framework: scan, detect, prioritize by threshold, act, record, and hand off.

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

Perceptual speed

Rapid visual comparison, symbol matching, and identification.

Perceptual speed is how quickly you can decide whether two visual strings, symbols or records are the same, and how quickly you can find a target among visually similar distractors. It is measured with deliberately trivial content (the items are easy, the clock is not), and it appears in clerical checking, numeric keypad, dispatch, security screening and several of the short commercial cognitive batteries. Nothing here rewards thinking harder, which is why candidates who are strong at reasoning often score below their expectation: the winning behaviours are chunking, a fixed scan path and knowing when to stop. Practice is stored on this device only, with no account and no upload unless you export it.

What you should be able to do after this lesson:

1. Tell a perceptual speed item apart from a reasoning item within the first two seconds, and switch off the

analytical approach that costs time here.
2. Compare long strings in fixed chunks of three or four characters rather than as a single run, and stop at the first mismatch when the instruction permits it.
3. Identify homoglyph pairs (0 and O, 1 and I and l, 5 and S, 2 and Z) and slow down deliberately on mixed alphanumeric strings where they appear.
4. Memorise a small lookup key in the opening seconds of a speeded section so the item rate is not limited by re-reading the key.
5. Scan a grid or column in a consistent raster order so no cell is visited twice and none is skipped.
6. Read a speeded score as correct-per-minute rather than percent-correct, and work out which of two performances wins under the section's actual scoring rule.

Worked examples and pitfalls

Chunk the string, stop at the first mismatch: Compare 4820 6193 5507 with 4820 6913 5507. Read them as three blocks, not as twelve digits. Block 1 matches, 4820 against 4820. Block 2 does not: 6193 against 6913, where the 1 and the 9 have swapped. You are done. On a same-or-different item there is no reason to check block 3, and checking it anyway costs you about a second on every item, which over eighty items is more than a minute of pure waste. The reason chunking works is capacity: a twelve-digit run exceeds what most people can hold for a comparison, so reading it whole forces you back to the source repeatedly, while a four-digit block fits in one go. The instruction wording decides whether early stopping is allowed. 'Are these identical?' permits it. 'How many of the five fields differ?' does not, and stopping early there converts a correct three into an incorrect one. Read the instruction once, classify the section, then run the matching procedure without re-deciding it per item.

Homoglyphs: 0 and O are where the time goes: Three pairs to judge. Nguyen, Thi-Hoa against Nguyen, Thi-Hoa: identical. Muller, Jorg against Mueller, Jorg: different, and the difference is in the surname spelling rather than anywhere in the given name, which is where most candidates look first. RN0O1I5 against RNO01I5: different, because positions three and four hold zero-then-letter-O in the first string and letter-O-then-zero in the second. That last pair is the killer, and it is deliberate. In most screen and print faces, a zero and a capital O differ by a few pixels, and the digit one, capital I and lower-case L can be visually identical. Production systems avoid these characters in generated references for exactly this reason; test writers include them for exactly the opposite one. The practical rule: whenever a string mixes letters and digits, drop your rate for that item, and use context to disambiguate. If a field is described as numeric, an O in it is a zero, and if it is a postcode you can often tell from the format which positions are letters. Guessing on a homoglyph is not a speed decision, it is a coin flip you can avoid by spending half a second.

Memorise the key, then never look at it again: A routing section supplies two small tables. Region: NW = 12, NE = 14, SW = 21, SE = 23, CE = 31. Priority: A = 1, B = 4, C = 7. Each item names a region and a priority and asks for the combined code, so 'SE, priority B' is 23-4 and 'NE, priority C' is 14-7. There are only eight mappings. If the section holds forty items in three minutes, you have 180 / 40 = 4.5 seconds per item, and a candidate who glances back at the key twice per item (once for the region, once for the priority) spends around two seconds of that on lookup alone, which is roughly eighty seconds of the section, or nearly half of it. Twenty seconds spent memorising eight pairs before the first item pays for itself about ten items in and then keeps paying. The general principle for speeded sections: any fixed information that appears in every item should be moved into memory as early as the format allows, because per-item costs multiply by the item count and one-off costs do not.

Cancellation drills and mirror-image distractors: Build your own cancellation drill rather than reusing a published one. Write twenty rows of forty characters drawn only from b, d, p and q, each carrying zero, one or two tick marks above or below. Define the target as 'b with exactly two marks' and cross out every occurrence, timed. What the drill teaches is feature overlap: a p with two marks is a vertical mirror of the target and a d with two marks is a horizontal mirror, and both take measurably longer to reject than an obviously different character would, because they share every feature except orientation. The same effect is why 6 and 9, and M and W, are expensive in real checking work. Two things improve with practice on this drill and neither is eyesight. The first is that you stop re-deriving the target on each row and start rejecting non-targets pre-attentively. The second is that you learn your own decay point, the row at which misses start appearing, which is the information you actually need in order to pace a real section.

Raster order on a scanning grid: A ten-by-ten grid of two-digit numbers, and the task is to mark every value ending in 7. Scan strictly left to right, top to bottom, one row at a time, keeping a finger or the cursor on the current row. The reason is not neatness. Free-roaming the eye over a hundred cells means some cells get visited twice and others not at all, and there is no way to know which is which, so you finish with no idea whether you are done. If you re-visit even fifteen percent of the grid, that is fifteen wasted fixations, and at roughly a quarter of a second each that is close to four seconds on one grid: often the difference between completing the section and leaving items blank. On a two-column layout, treat each column as its own raster and finish it before moving; jumping between columns doubles the re-visit rate. The same discipline transfers directly to real work: scanning a picking list, a passenger manifest or a stock count row by row is slower per glance and faster per document.

Correct per minute, not percent correct: Two candidates, four minutes each. A attempts 62, gets 60 right: 15.0 correct per minute at 96.8 percent accuracy. B attempts 96, gets 84 right: 21.0 correct per minute at 87.5 percent accuracy. Under raw scoring, count the correct answers, B wins, 84 to 60. Under correct minus incorrect, A scores 60 - 2 = 58 and B scores 84 - 12 = 72, so B still wins. Under correct minus three times incorrect, A scores 60 - 6 = 54 and B scores 84 - 36 = 48, and A wins. The same two performances rank three different ways depending on a single line in the instructions. This is why percent-correct is close to useless as a self-assessment on a speeded test: it ignores the rate, and the rate is the whole construct. Log both numbers after every practice session (attempts and correct, from which rate and accuracy both fall out), and compare sessions on rate. A session where accuracy improved but the rate fell is not necessarily progress.

How to practise this skill

- Practise in short, intense blocks of two to four minutes rather than long sessions. The construct is a rate under pressure, and a relaxed twenty-minute session trains a different behaviour from the one being measured.
- Never sub-vocalise a string you are comparing. Saying '4820' in your head takes about as long as saying it aloud and caps your rate; comparing the shapes of the blocks does not.
- Deliberately vary the drill format between name checking, number checking, symbol cancellation and table lookup. Speed built on one format transfers less than candidates expect, and real sections mix them.
- Slow down on purpose for mixed alphanumeric strings and speed back up on pure numeric ones. A uniform pace is the wrong strategy when the items have unequal difficulty.
- Fix your scan path before starting (chunk size, direction, whether you finish a column before moving), and do not change it mid-section. Changing procedure under time pressure costs more than the procedure was ever going to save.
- Record attempts and correct answers separately for every session so you can plot rate over time. Scores stay on this device unless you export them, so the log is yours and is worth keeping across weeks rather than days.

Glossary

Speeded test: A test whose items are easy enough that nearly everyone could answer them given unlimited time, so the score reflects how many were completed. Contrasted with a power test, where difficulty rather than the clock is the limiting factor.

Cancellation task: A format in which a target character or symbol must be marked wherever it appears in a dense field of similar non-targets, scored on hits, misses and false marks within a fixed time.

Distractor (perceptual): A non-target that shares visual features with the target. The more features shared, the longer the rejection takes, which is why mirror-image pairs such as b and d are the most expensive distractors of all.

Homoglyph: A character that looks nearly identical to another in a given typeface: 0 and O, 1 and I and l, 5 and S. The commonest single cause of unavoidable errors in alphanumeric checking.

Chunking: Grouping a long string into units of three or four characters so each comparison fits within working memory. It is the difference between one pass over a reference number and several.

Raster scan: Covering a field in a fixed order, row by row in one direction, so that every cell is visited exactly once. Its value is not tidiness but the guarantee that the search is complete.

Rate measure: Performance expressed per unit of time, such as correct answers per minute. It is the only fair basis for comparing two speeded sessions of different lengths.

Ceiling effect: What happens when a test is easy enough that many candidates reach the maximum score, destroying its ability to discriminate. Speeded sections are deliberately long enough that almost nobody finishes, which is why an unfinished section is normal rather than a failure.

Study this next

- Perceptual speed skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/perceptual-speed
- Perceptual speed practice in the Cognitive Skills Lab: https://learn.novusstreamsolutions.com/cognitive-skills/perceptual-speed
- Processing speed lesson: https://learn.novusstreamsolutions.com/aptitude/skills/processing-speed/lesson
- Clerical checking suite: https://learn.novusstreamsolutions.com/aptitude/tests/clerical-checking
- Numeric keypad suite: https://learn.novusstreamsolutions.com/aptitude/tests/numeric-keypad
- Perceptual speed lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/perceptual-speed/lesson

Where this material comes from

- All strings, code tables, grids and drills above were written for this lesson. The rate and score comparisons were computed and checked, including the three different orderings produced by the three scoring rules.
- Timing figures given for fixations, lookups and per-item costs are illustrative arithmetic showing the shape of the trade-off, not measured research findings; use your own logged times in their place.
- Terminology (speeded versus power tests, cancellation tasks, chunking, homoglyphs) cross-checked against standard public references such as the Wikipedia articles 'Speed test (psychometrics)', 'Chunking (psychology)' and 'Homoglyph'.
- 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.

Recommended preparation

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

- Strengthen attention and concentration: https://learn.novusstreamsolutions.com/aptitude/skills/attention-concentration/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/game-based-assessment-familiarization/study-outline