# Filing and coding: study guide

Administration, clerical, records, and office support · suite apt-316-filing-and-coding · generated 2026-09-15T15:02:56.327Z

| Detail | Value |
|---|---|
| Title | Filing and coding: study guide |
| Generated | 2026-09-15T15:02:56.327Z |
| Fixture/version | apt-316-filing-and-coding |
| Sector | Administration, clerical, records, and office support |
| 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 Filing and coding*

| 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-316-filing-and-coding&mode=guided
- Start mini-test: https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-316-filing-and-coding&mode=mini
- Start full simulation: https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-316-filing-and-coding&mode=full

## Skills covered, in full

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

### Coding and decoding

Applying symbol, letter, number, and substitution rules.

Coding and decoding items give you an artificial rule - shift every letter forward by three, replace each word with a nonsense token, code digits through a table with exceptions - and ask you to apply or reverse it, quickly and without slips. The construct is really letter-position arithmetic plus disciplined bookkeeping, which is why it appears in filing and coding batteries, clerical and records screening, signals and communications tests, and any speeded section where accuracy under a clock is the point. The same habits carry into real work: reference-code systems, batch prefixes and case numbering all reward someone who can apply a rule table without drifting. Everything you practise stays on this device unless you export it.

What you should be able to do after this lesson:

1. Convert between letters and positions instantly using anchors (A=1, E=5, J=10, O=15, T=20, Y=25, Z=26) rather than counting from A.
2. Solve forward and backward shift codes as signed arithmetic modulo 26, handling wrap-around at both ends of the alphabet without recounting.
3. Crack word-to-code substitution items by intersecting pairs of lines that share exactly one word, and explain why code word order carries no information unless the item says it does.
4. Apply a conditional coding table with a stated precedence rule, including repeated-digit clauses and mutually exclusive first-and-last conditions.
5. Use the reverse-alphabet complement rule - position plus mirrored position equals 27 - instead of writing out a reversed alphabet.
6. Detect non-constant rules by computing the shift for every letter and looking for a repeating or position-indexed pattern in the difference sequence.

### Worked examples and pitfalls

Shift codes, wrap-around, and doing it as arithmetic: If FLOWER is coded GMPXFS, each letter has moved forward one place: F to G, L to M, O to P, W to X, E to F, R to S. Applying the same rule, GARDEN becomes HBSEFO. Wrap-around is where the counting method breaks: under the same plus-one rule, ZEBRA becomes AFCSB, because after Z the alphabet starts again at A. Backward shifts are the same operation with a negative sign. If TABLE is coded QXYIB, work in positions: T is 20 and Q is 17, so the shift is minus three. Check it against the awkward letters - A is 1, and 1 minus 3 is minus 2, which becomes 24 after adding 26, and the 24th letter is X. B is 2, minus 3 gives minus 1, which becomes 25, which is Y. L is 12, minus 3 is 9, which is I. E is 5, minus 3 is 2, which is B. Every letter confirms minus three. Candidates who count backwards on their fingers get A and B wrong in about a third of attempts, and they get them wrong silently. Convert to numbers, do the arithmetic, add or subtract 26 if you fall outside 1 to 26, convert back.

Anchors, position sums, and the repeated-letter trap: The five-letter anchor set E=5, J=10, O=15, T=20, Y=25 - taught almost everywhere as EJOTY - puts every letter within two or three steps of a known number, and adding A=1, M=13 and Z=26 closes the gaps. Now a position-sum item. If SUN is coded 54, what is MOON? S is 19, U is 21, N is 14, and 19 + 21 + 14 = 54, so the rule is the sum of the letter positions. MOON is M=13, O=15, O=15, N=14, which sums to 57. The repeated O is the trap: under time pressure candidates count each distinct letter once and produce 42, which is a clean-looking wrong answer. There is a second, deeper point about this rule. A position sum is many-to-one, so different words share the same code - NUS, SUN and UNS all give 54, and so does any other arrangement - which means a sum code can be applied but never reversed. If an item asks you to decode a number back to a unique word, the rule cannot be a plain

sum, and you should be looking for something order-sensitive such as concatenated positions or alternating operations.

Word substitution: intersect the lines, ignore the order: Given three coded sentences: 'pit na som' means 'bring the file'; 'som lek dar' means 'file is missing'; 'dar na tuv' means 'missing the envelope'. Work by intersection. Lines one and two share exactly one English word, 'file', and their code sets share exactly one token, 'som', so som means file. Lines two and three share only 'missing', and their codes share only 'dar', so dar means missing. Lines one and three share only 'the', and their codes share only 'na', so na means the. Every remaining token now falls out by elimination: in line one that leaves pit for bring, in line two lek for is, in line three tuv for envelope. So 'bring the envelope' is written with the tokens pit, na and tuv in any order. That last clause is the point most candidates miss: in this format the position of a token carries no information unless the item explicitly states that word order is preserved, so an answer option that lists the right three tokens in a different sequence is correct, and rejecting it is a common way to lose a mark you had already earned. Always work the intersections on paper - three lines of three tokens exceeds what most people can hold reliably in working memory while also doing the elimination.

Conditional coding: read the precedence line first: Rule table: digits 1 to 9 are coded P, Q, R, S, T, U, V, W, X in order, so 1 is P, 5 is T, 9 is X. Three conditions apply. (i) If the first digit is odd and the last is even, code both as A. (ii) If the first and last digits are both odd, code both as B. (iii) Any digit that repeats an earlier digit in the same number is coded Z. Precedence: check (i) and (ii) first - they are mutually exclusive - then apply (iii), judged against the original digits, to the positions not already replaced. Take 35258. First digit 3 is odd, last digit 8 is even, so condition (i) fires and both ends become A. The middle digits are 5, 2, 5: the first 5 is a first occurrence and codes T; the 2 codes Q; the second 5 repeats, so it codes Z. The answer is A T Q Z A. Take 7441. First 7 and last 1 are both odd, so condition (ii) fires and both ends become B. The middle 4s give S then Z. The answer is B S Z B. Take 5359. Both ends odd, so (ii) fires and both are B; the middle 3 codes R, and the middle 5 repeats the original first digit, so it codes Z, giving B R Z B. That last case is the one to study, because whether a repeat is judged against the original digits or against the already-substituted ones changes the answer - and the item's own precedence line, not your instinct, is what decides it.

Reverse alphabet: use 27, never rewrite the alphabet: In a reversed code A stands for Z, B for Y, and so on. Rather than writing out the reversed alphabet under the normal one - which takes twenty seconds and introduces errors - use the complement rule: a letter at position n maps to position 27 minus n, because the pairs always sum to 27. Code MASK: M is 13, so 27 minus 13 is 14, which is N. A is 1, so 26, which is Z. S is 19, so 8, which is H. K is 11, so 16, which is P. MASK becomes NZHP. The same rule answers the position questions that accompany these items. 'Which letter occupies the same position in the reversed alphabet that M occupies in the normal one?' M is 13th, and the 13th letter counted from the end is 27 minus 13, which is 14, so N. 'Which letter is midway between the 5th and the 21st letters of the alphabet?' Take the mean of the positions, (5 + 21) / 2 = 13, which is M - and note that midway questions built on a letter and its own mirror never have an answer, because n and 27 minus n always sum to an odd number. Watch the phrasing too: 'the 7th letter from the right' means position 27 minus 7 = 20, which is T, and misreading left for right is the most common single error in this family.

When the shift is not constant: If CAT is coded DZU, compute each shift separately rather than assuming one rule: C to D is plus one, A to Z is minus one (1 minus 1 is 0, which becomes 26 after adding 26), T to U is plus one. The difference sequence is plus one, minus one, plus one - alternating, not constant. Apply the same alternating rule to BIRD: B plus 1 is C, I minus 1 is H, R plus 1 is S, D minus 1 is C, giving CHSC. A different family indexes the shift to the letter's position in the word. Code FLAME with shifts of plus one, plus two, plus three, plus four, plus five: F is 6 so 7 is G; L is 12 so 14 is N; A is 1 so 4 is D; M is 13 so 17 is Q; E is 5 so 10 is J. FLAME becomes GNDQJ. The general procedure is worth internalising because it never fails: write the numeric difference under every letter pair, then read the difference sequence. Constant means a simple shift; alternating means a two-rule cycle; 1, 2, 3, 4 means a position-indexed shift; and a sequence with no pattern at all usually means the letters have been reordered as well as shifted, which you can confirm

by checking whether the coded word contains the same multiset of shifted letters in a different order.

## How to practise this skill

- Write the alphabet with its position numbers on your scrap paper before the section starts, once. Every subsequent item becomes arithmetic instead of counting, and the twenty seconds it costs is repaid on the second question.
- Learn the EJOTY anchors so hard that they are not recall but reflex: E=5, J=10, O=15, T=20, Y=25. Candidates who count from A on every letter simply cannot finish a speeded coding section, however accurate they are.
- Treat every shift as a signed number reduced into the range 1 to 26 by adding or subtracting 26. Counting letters near A and near Z is where almost all silent errors in this construct are made.
- On substitution items, write the intersections down. Holding three code lines and three English lines in your head while doing set intersection is exactly the load that produces confident wrong answers.
- Check the repeated-letter or repeated-digit case explicitly on every item, as a separate final step. It is the single most common careless error in the construct, and it costs a mark on an item you had otherwise solved correctly.
- Because these sections are speeded, train accuracy first and speed second: run untimed sets until the alphabet arithmetic is automatic, then time yourself and watch which error type reappears under the clock. Attempts are recorded on this device only, so a messy first pass costs nothing.

## Glossary

Substitution cipher: A code in which each letter, word or symbol is replaced by a fixed counterpart. The mapping is the whole content of the code, so recovering the mapping solves every item built on it.

Caesar shift: A substitution in which every letter moves the same fixed number of places along the alphabet. Fully described by one signed number, which is why finding the shift for one letter pair solves the item.

Wrap-around: The rule that the alphabet is circular: past Z you return to A, and before A you return to Z. Handled by adding or subtracting 26 to bring a position back into the range 1 to 26.

EJOTY: The standard mnemonic for the anchor positions E=5, J=10, O=15, T=20, Y=25. Every letter is within two places of an anchor, which turns letter-to-number conversion into a one-step adjustment.

Complement rule: In a reversed alphabet, a letter at position n maps to position 27 minus n, since the mirrored pairs always sum to 27. It replaces writing out the reversed alphabet by hand.

Conditional coding: A code where a base table is overridden by exception clauses tied to the first digit, the last digit, or repetition. The precedence statement governing which clause wins is the part of the item that must be read first.

Many-to-one code: A rule such as summing letter positions, under which different inputs produce identical codes. Such a code can be applied but not reversed, so any decode question implies a different, order-sensitive rule.

Difference sequence: The list of signed shifts between each plaintext letter and its coded counterpart. Constant means a simple shift, alternating means a repeating cycle, and 1, 2, 3, 4 means a position-indexed rule.

## Study this next

- Coding and decoding skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/coding-decoding
- Practise the coding and decoding bank: https://learn.novusstreamsolutions.com/cognitive-skills/coding-decoding
- Filing and coding suite: https://learn.novusstreamsolutions.com/aptitude/tests/filing-and-coding

- Processing speed lesson: https://learn.novusstreamsolutions.com/aptitude/skills/processing-speed/lesson
- Diagrammatic reasoning lesson: https://learn.novusstreamsolutions.com/aptitude/skills/diagrammatic-reasoning/lesson
- Coding and decoding lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/coding-decoding/lesson

### Where this material comes from

- Worked items written for Novus Learn. Every code, substitution table, rule set and worked answer above is original and invented for this lesson; no published or copyrighted test item is reproduced.
- The EJOTY anchor mnemonic and the reverse-alphabet complement rule are long-standing public study aids in wide general circulation, reproduced here as method rather than as anyone's proprietary material.
- Terminology follows standard, widely published usage in elementary cryptography - substitution cipher, Caesar shift, modular wrap-around.
- 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.

### Data checking and clerical accuracy

Matching, filing, coding, record checking, and identifying discrepancies.

Clerical accuracy is the ability to handle a large volume of records correctly and consistently: filing them in the right order, coding them against a key, spotting duplicates that do not look like duplicates, and holding that standard on the two-hundredth record as well as the second. Where error detection asks whether two things match, clerical accuracy asks whether you can apply a rule reliably at volume. It is assessed in administrative, records, clinical admin, school office, evidence-handling and insurance operations selection, and it is exactly the skill an employer is buying when they hire for a data-heavy back-office role. 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. Apply an alphabetical filing rule consistently, and state whether a given system files letter-by-letter or word-by-word. The two produce different, equally correct orders.
2. Sort alphanumeric references correctly under both string ordering and natural numeric ordering, and explain why record systems zero-pad.
3. Code records against a written key, handling every boundary condition (under, up to, and over, inclusive ranges) exactly as written rather than as intuited.
4. Identify duplicate records that differ only in formatting (case, whitespace, date order, punctuation inside a reference), and route genuinely ambiguous ones to exceptions instead of resolving them by guess.
5. Measure your own accuracy decay across a long batch and use the measurement to set a realistic attempt rate.
6. Convert a stated scoring rule into a target items-per-minute figure, and show why the optimum rate moves when the penalty multiplier changes.

### Worked examples and pitfalls

Word-by-word or letter-by-letter: two correct answers, one rule: File these four names: Van Dyke, Vandenberg, Van Horn, Vance. Under word-by-word filing, each space-separated unit is compared in turn and 'nothing files before something', so the first unit 'Van' sorts ahead of both 'Vance' and 'Vandenberg'. The

order is Van Dyke, Van Horn, Vance, Vandenberg. Under letter-by-letter filing, spaces are ignored entirely, so the keys are VANCE, VANDENBERG, VANDYKE, VANHORN, and the order is Vance, Vandenberg, Van Dyke, Van Horn. Both orders are correct filing; only one is correct for the system you are working in. Test items state the convention in the instructions, and the candidates who lose marks are almost always the ones applying whichever convention their previous employer used. The same fork appears with prefixes and punctuation: whether Mc and Mac interfile, whether St is treated as Saint, whether a hyphen counts as a space. Read the rule, restate it to yourself in one sentence, then apply it mechanically and do not let a name that 'obviously' belongs somewhere override it.

Alphanumeric codes: A-102 at the front or the back of the drawer: Sort the references A-7, A-12, A-70, A-102, B-3. Under natural numeric ordering, the way a person reads them, the answer is A-7, A-12, A-70, A-102, B-3. Under plain string ordering, the way most software sorts by default, each character is compared in turn, so '1' comes before '7' and the answer is A-102, A-12, A-7, A-70, B-3, with A-102 first rather than last. Both are defensible; a filing item is testing which one the stated system uses. This is also why serious record systems zero-pad their references: rewrite the set as A-007, A-012, A-070, A-102 and the string sort and the numeric sort agree, permanently. If you are ever asked to design or clean a reference scheme, pad to a fixed width and the whole class of problem disappears. In a test, the giveaway is a set that deliberately mixes one-, two- and three-digit suffixes; that mixture exists only to separate candidates who apply the stated rule from candidates who apply the intuitive one.

Coding to a key: every mark is at a boundary: A claims routing key: under 500 pounds with no injury reported goes to CS1; under 500 with injury goes to CI2; 500 to 4,999 with no injury goes to CS3; 500 to 4,999 with injury goes to CI4; 5,000 and over goes to CE5 regardless of injury. Now code five records. R-118 at 499.99, no injury: CS1, since 499.99 is under 500. R-119 at exactly 500.00, no injury: CS3, because 'under 500' excludes 500 itself and the second band starts there. R-120 at 4,999.00 with injury: CI4, since the band is inclusive at its top. R-121 at exactly 5,000.00 with no injury: CE5, because the top band is 'and over' and its 'regardless of injury' clause overrides the injury split that governs the lower bands. R-122 at 86.40 with injury: CI2. Four of those five decisions turn on a boundary, and that is not an accident, coding items are written so that the interior cases are trivial and every discriminating mark sits on an edge. Before coding anything, underline the boundary words: under, up to, and over, between, inclusive. Then decide, once, what each one does to the endpoint, and apply that decision to every record in the batch.

Duplicates that are not textually identical: Three rows arrive in a merge. Row 1: SMITH, JANE | 07/04/1988 | AB123456C. Row 2: Smith, Jane | 1988-04-07 | AB 123456 C. Row 3: SMITH, JANE | 04/07/1988 | AB123456C. Rows 1 and 2 are the same person: normalise case, strip the spaces from the reference, and convert the ISO date and they match exactly. Row 3 is the genuinely hard one. If the file is in day/month order it is a different date of birth and possibly a different person; if that row came from a system using month/day order it is the same record again. You cannot tell from the row itself, so the correct action is to send it to the exception queue with the ambiguity noted, not to merge it and not to discard it. This is the judgement that separates competent records work from the appearance of it: the goal is not to make every row disappear, it is to make every decision defensible. In test form the item usually asks 'how many distinct individuals are represented', and the answer is often given as a range or accompanied by a 'cannot determine' option for exactly this reason.

Where accuracy actually decays on a long batch: Run a self-measurement rather than trusting a number from anyone: take 200 record pairs, split them into four blocks of 50, and log errors per block. Most people find block 1 slightly worse than block 2, a warm-up cost, and then a rise across blocks 3 and 4 as vigilance falls. The reason to measure it is that the arithmetic of small percentages is brutal at volume. Checking 200 pairs at 98 percent accuracy passes 4 bad records; at 99.5 percent it passes 1. Scale that to a realistic month of 20,000 records and the same two accuracy rates mean 400 defects against 100: a four-fold difference in downstream rework from a 1.5 point difference that would look like noise on a single test. Once you know where your own curve turns, the intervention is cheap: a deliberate twenty-second break at that

point, or splitting the batch so the hardest records fall in your strongest block. Practice sessions on this site are recorded on your own device, so building a block-by-block picture across several sessions costs nothing but the logging.

Setting an attempt rate from the scoring rule: A 120-item checking test with a 10-minute limit, scored as correct minus incorrect. Suppose practice has told you that you hold 92 percent at ten items a minute and 96 percent at seven and a half. Fast: 10 x 10 = 100 attempted, 92 correct and 8 wrong, score 84. Careful: 10 x 7.5 = 75 attempted, 72 correct and 3 wrong, score 69. Fast wins by 15. Now change the rule to correct minus three times incorrect: fast scores 92 - 24 = 68, careful scores 72 - 9 = 63, and the 15-point gap has shrunk to 5. Now suppose your real accuracy at ten a minute is 80 percent rather than 92. A gap most people do not discover until they measure it. Fast now gets 80 right and 20 wrong: under simple correct-minus-incorrect that is 60, already behind the careful strategy's 69, and under the triple penalty it is 80 - 60 = 20 against 63. Same test, same person, opposite advice, and the two deciding variables are the penalty multiplier, which the instructions hand you, and your own accuracy-at-speed, which only measurement gives you. Do not pick a pace from temperament. Measure two rates in practice, write both accuracy figures down, and do this arithmetic before the test rather than during it.

## How to practise this skill

- Before the first record of any batch, write the filing or coding rule at the top of the page in your own words. The single largest source of clerical error is applying a remembered rule from a previous system.
- Underline the boundary words in a coding key (under, up to, and over, inclusive), and resolve each endpoint once. Interior cases carry almost no marks; the edges carry nearly all of them.
- Normalise before you compare: mentally strip case, spaces and punctuation from references, and restate dates in one fixed order. Half of apparent duplicates are formatting, and half of apparent non-duplicates are too.
- When a record is genuinely ambiguous, mark it and move on. Time spent resolving one unresolvable row is taken from thirty rows you could have done correctly, and a guessed merge is worse than a flagged one.
- Measure your accuracy at two different speeds in practice and write both numbers down. You cannot choose an attempt rate rationally without them, and the fast rate is almost never as accurate as it feels.
- Practise on batches long enough to hit your own fatigue point, not on ten-item samples. The construct is specifically about sustained accuracy, and a ten-item drill measures the part of the curve that was never in doubt.

## Glossary

Word-by-word filing: An alphabetical convention that compares space-separated units in turn, with a shorter first unit filing before a longer one. Under it, Van Horn files before Vance.

Letter-by-letter filing: An alphabetical convention that ignores spaces and punctuation entirely, comparing the run of letters. Under it, Vance files before Van Dyke.

Natural sort: Ordering that reads embedded digit runs as numbers, so A-7 precedes A-12. Contrasted with string ordering, which compares characters one at a time and puts A-102 first.

Zero padding: Writing references to a fixed width by adding leading zeros (A-007 rather than A-7) so that string ordering and numeric ordering produce the same result. The standard structural fix for alphanumeric filing errors.

Primary and secondary sort key: The field sorted on first and the field used to break ties within it. A batch sorted by site then by surname will look wrong if the two keys are applied in the opposite order.

Canonicalisation: Converting records to a single standard form (one case, no stray whitespace, one date order, punctuation stripped from references) before any comparison, so that formatting differences stop

masquerading as data differences.

Exception queue: The destination for records that cannot be resolved from the information available. Routing an ambiguous record here is a correct outcome; guessing it into a merge is not.

Boundary condition: The endpoint of a coded band, where wording such as under, up to or and over decides which side a value falls. Coding tests concentrate their discriminating items here.

### Study this next

- Data checking and clerical accuracy skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/clerical-accuracy
- Clerical accuracy practice in the Cognitive Skills Lab: https://learn.novusstreamsolutions.com/cognitive-skills/clerical-accuracy
- Error detection lesson: https://learn.novusstreamsolutions.com/aptitude/skills/error-detection/lesson
- Administrative and clerical entry suite: https://learn.novusstreamsolutions.com/aptitude/tests/administrative-and-clerical-entry
- Clerical checking suite: https://learn.novusstreamsolutions.com/aptitude/tests/clerical-checking
- Data checking and clerical accuracy lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/clerical-accuracy/lesson

### Where this material comes from

- All filing sets, reference codes, routing keys and records above were invented for this lesson. The two filing orders, the two sort orders and every score calculation in the attempt-rate example were worked through and checked by recomputation.
- Filing and sorting conventions cross-checked against standard public descriptions such as the Wikipedia articles 'Alphabetical order', 'Natural sort order' and 'Collation'. Terminology only; the examples are original.
- 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.

### 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.

## Error detection

Comparing strings, records, transactions, forms, and data for mistakes.

Error detection is the skill of holding a source and a copy side by side and finding the one character, digit or field that moved, and of knowing which errors a given check will and will not catch. It is assessed directly in clerical checking, banking operations, records, proofreading and dispatch selection, and it underpins quality control anywhere a record is rekeyed or transcribed. The useful part of the skill is not staring harder; it is a

repeatable scan procedure plus a set of arithmetic checks that catch what the eye misses. Everything you practise 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. Name the four transcription error families (substitution, transposition, omission, duplication), and state which of them a plain digit-sum check will fail to detect.
2. Run a fixed field-order comparison between a source record and a copy, reporting the specific field and character position that differs rather than a general impression.
3. Compute a modulus 11 check digit (ISBN-10 style) by hand and use it to decide whether a single record is internally consistent.
4. Apply the Luhn algorithm to a candidate account number, and state the one transposition case Luhn provably cannot catch.
5. Work out, from a stated scoring rule, whether guessing on an uncertain item has positive, zero or negative expected value.
6. Reconcile a document at batch level (line totals, subtotal, tax, grand total) and explain why an internally consistent document can still be wrong.

## Worked examples and pitfalls

The four transcription error families: Source reference: INV-2024-08871. Four corrupted copies, one of each family. INV-2024-08571 is a substitution: a single character changed, 8 to 5. INV-2024-0871 is an omission: one character dropped, and the string is now a character short. INV-2024-088711 is a duplication: a character repeated, one character long. INV-2024-08817 is a transposition: the final 7 and 1 have swapped places, and crucially the string is the same length and contains exactly the same characters. That last property is why transposition is the hardest of the four to see and the most dangerous in practice. Anything that checks length catches omission and duplication. Anything that compares character sets or sums the digits catches substitution but not transposition, because 0+8+8+7+1 and 0+8+8+1+7 both come to 24. Reading the reference aloud catches substitution reliably and transposition poorly, because the ear is comparing sounds and both versions sound similar at speed. The only defences against transposition are a positional comparison and a weighted check digit.

A fixed scan order finds the one field that moved: Source record: Name Priyanka Ramaswamy / Account 4471-9026-3358 / Date of birth 14/03/1991 / Postcode SW1A 2AA / Phone 0161 496 0872. Copy: Name Priyanka Ramaswamy / Account 4471-9026-3358 / Date of birth 14/03/1991 / Postcode SW1A 2AA / Phone 0161 469 0872. The difference is in the phone field, where 496 has become 469: a transposition inside the middle block. Most candidates find it eventually; the ones who find it in eight seconds are the ones running a procedure. Always compare in the same field order, top to bottom, never skipping a field because it 'looks fine'. Compare long strings in the printed blocks rather than as a single run (4471, then 9026, then 3358) because short-term memory holds about four items and a twelve-digit run does not fit. Say the block silently, look, compare, move on. On a same-or-different item you may stop at the first mismatch; on a 'how many fields differ' item you must complete every field, and the instruction wording is what tells you which regime you are in. Getting this wrong in either direction costs marks: stopping early on a count item, or exhaustively checking a same-or-different item you had already resolved.

Modulus 11: why a weighted check digit catches a transposition: The ISBN-10 scheme multiplies the ten digits by descending weights 10, 9, 8 down to 1 and requires the total to be divisible by 11. Take 0-306-40615-2. Working left to right: 10x0 = 0, 9x3 = 27, 8x0 = 0, 7x6 = 42, 6x4 = 24, 5x0 = 0, 4x6 = 24, 3x1 = 3, 2x5 = 10, and the check digit 1x2 = 2. The sum is 0 + 27 + 0 + 42 + 24 + 0 + 24 + 3 + 10 + 2 = 132, and 132 = 12 x 11 exactly, so the number is valid. Now transpose the 1 and the 5 to give 0-306-40651-2. The digits are identical, so a plain digit sum is unchanged at 27 either way and would report no problem. The weighted sum, however, becomes 0 + 27 + 0 + 42 + 24 + 0 + 24 + 15 + 2 + 2 = 136, and 136 - 132 = 4, so it is not a multiple of 11 and the record is rejected. That is the entire reason check digits are weighted rather

than plain: position has to matter, or the commonest human error passes straight through.

Luhn: the number that fails, and the one case it misses: The Luhn check, used on payment and many membership numbers, doubles every second digit counting from the right, subtracts 9 from any doubled result above 9, sums everything, and requires a total ending in zero. Take 4539 1488 0343 6467. The doubled positions contribute 3, 3, 8, 0, 7, 2, 6 and 8, totalling 37; the undoubled positions contribute 7, 4, 3, 3, 8, 4, 9 and 5, totalling 43. The grand total is 80, which ends in zero, so the number passes. Now transpose the last two digits to 4539 1488 0343 6476. The doubled contributions become 5, 3, 8, 0, 7, 2, 6, 8 = 39 and the undoubled become 6, 4, 3, 3, 8, 4, 9, 5 = 42, giving 81. Not a multiple of ten, so the mistyped number is rejected at the point of entry. Luhn catches every single-digit substitution and almost every adjacent transposition: with exactly one blind spot. Swapping an adjacent 0 and 9 leaves the total unchanged, because doubling 0 gives 0 and doubling 9 gives 18 which reduces to 9, so both digits contribute the same amount whether doubled or not. A form that validates with Luhn will happily accept 90 where you typed 09. Knowing the blind spot is the point: a check digit narrows the space of undetected errors, it never closes it.

Negative marking: 44 right can beat 46 right: Many clerical checking sections score correct minus incorrect, with omissions neutral. Under that rule, guessing between two remaining options has an expected value of 0.5 x (+1) + 0.5 x (-1) = 0, exactly neutral, and guessing blindly among four options has an expected value of 0.25 - 0.75 = -0.5, clearly negative. Concretely: on a 60-item section you attempt 48, get 44 right and 4 wrong, and score 40. A colleague attempts all 60, gets 46 right and 14 wrong, and scores 32. More correct answers, a lower score. Reverse the scoring rule to plain raw-correct with no penalty and the ordering flips: 46 beats 44 and leaving a blank becomes strictly irrational. Neither strategy is universally right, which is why the instruction screen is not optional reading. Find the sentence that says whether wrong answers are penalised, and if it is absent, assume raw scoring and answer everything.

Reconcile the batch: consistent is not the same as correct: An invoice lists three lines: 12 at 14.25 = 171.00; 7 at 33.60 = 235.20; 3 at 128.00 = 384.00. The stated subtotal is 709.20, VAT at 20 percent is stated as 141.84, and the total is stated as 851.04. Every downstream figure checks out against the one above it: 709.20 x 0.20 = 141.84 and 709.20 + 141.84 = 851.04. Nothing in the tax or total column is inconsistent. But recompute the lines: 171.00 + 235.20 + 384.00 = 790.20, not 709.20. The subtotal is a transposition of the correct figure, and because every later number was calculated from the wrong subtotal, the document is perfectly self-consistent and perfectly wrong. The true VAT should be 158.04 and the true total 948.24, a shortfall of 97.20. This is the single most valuable habit in the construct: never verify a total against the number printed above it, always recompute it from the underlying items. Internal consistency proves that one person did the arithmetic carefully; it proves nothing about whether they started from the right number.

## How to practise this skill

- Drill transposition specifically. Generate pairs where the only difference is two adjacent characters swapped, because that is the family your eye is worst at and the family a naive check will not catch.
- Fix a scan order and never vary it: field by field, top to bottom, and within a long value block by block. Free-roaming comparison feels faster and reliably misses one field per record.
- Learn one check-digit scheme by hand, modulus 11 is the easiest, until you can run it in about twenty seconds. It converts a whole class of 'does this record look right' items from judgement into arithmetic.
- Read the scoring rule before the first item and decide your guessing policy then, not at minute seven when you are behind. Under penalty scoring, an omission is a legitimate answer; under raw scoring it is a wasted mark.
- When you are checking financial or tabular documents, recompute the lowest-level figures first and work upward. Errors introduced at a subtotal propagate perfectly and are invisible from below.
- Log the errors you miss, not the ones you find. A miss log after four sessions usually shows a personal signature (always the middle of long numeric strings, or always the second occurrence of a repeated field), and that signature is what to drill.

## Glossary

Transposition error: Two adjacent characters exchanged, as in 496 typed for 469. Length and character content are unchanged, so length checks and plain digit sums both pass it; only positional comparison or a weighted check digit detects it.

Substitution error: One character replaced by another, as in 08571 for 08871. It is the family most reliably caught by reading a value aloud against the source, and by any check digit scheme.

Check digit: An extra digit computed from the others so that a corrupted record fails an arithmetic test. It detects errors; it never corrects them and never proves the record refers to the right person or item.

Modulus 11: A weighted check scheme in which digits are multiplied by descending position weights and the total must divide by 11. Used in ISBN-10 and several banking and national identifier formats.

Luhn algorithm: A modulus 10 check that doubles alternate digits from the right and requires the sum to end in zero. It catches all single-digit errors and most adjacent transpositions except an adjacent 0 and 9.

Miss and false alarm: A miss is a genuine discrepancy passed as correct; a false alarm is a difference flagged where none exists, usually a formatting variant such as a trailing space or a differently written date. Misses are the costlier of the two in records work, which is why procedures normalise formatting first and then bias toward escalating doubt.

Correction for guessing: A scoring rule that subtracts a fraction or multiple of the wrong answers from the correct ones, making blind guessing negative in expectation and turning omission into a rational choice.

Reconciliation: Recomputing an aggregate from its components rather than accepting the printed figure. It is the only check that catches an error introduced at summary level, where everything below and above it still agrees.

## Study this next

- Error detection skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/error-detection
- Error detection practice in the Cognitive Skills Lab: https://learn.novusstreamsolutions.com/cognitive-skills/error-detection
- Data checking and clerical accuracy lesson: https://learn.novusstreamsolutions.com/aptitude/skills/clerical-accuracy/lesson
- Clerical checking suite: https://learn.novusstreamsolutions.com/aptitude/tests/clerical-checking
- Copyediting and proofreading suite: https://learn.novusstreamsolutions.com/aptitude/tests/copyediting-and-proofreading
- Error detection lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/error-detection/lesson

## Where this material comes from

- The ISBN-10 modulus 11 worked example (0-306-40615-2, weighted sum 132) and the Luhn worked example (4539 1488 0343 6467, sum 80) were computed by hand for this lesson and each was re-verified digit by digit, including the corrupted variants.
- Algorithm definitions cross-checked against the public specifications described in the Wikipedia articles 'International Standard Book Number' and 'Luhn algorithm', including the documented 09/90 transposition blind spot. Records, invoices and reference numbers above are invented.
- 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.

## Situational judgment

Evaluating workplace responses against role-relevant principles.

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

What you should be able to do after this lesson:

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

## Worked examples and pitfalls

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

Method: facts, duties, risks, response: Write four short notes before ranking options: what is known, who may be affected, which duty or boundary applies, and what safe next step is available now. Prefer an action that addresses the immediate issue and creates useful follow-through. Do not reward an option merely because it sounds decisive.

## How to practise this skill

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

## Glossary

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

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

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

## Study this next

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

### Where this material comes from

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

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

## Recommended preparation

- Practice coding and decoding: https://learn.novusstreamsolutions.com/aptitude/skills/coding-decoding/lesson
- Strengthen clerical accuracy: https://learn.novusstreamsolutions.com/aptitude/skills/clerical-accuracy/lesson
- Strengthen error detection: https://learn.novusstreamsolutions.com/aptitude/skills/error-detection/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/filing-and-coding/study-outline*