Bookkeeping support: study guide

Administration, clerical, records, and office support · suite apt-317-bookkeeping-support · generated 2026-09-15T15:04:50.834Z

Detail	Value
Title	Bookkeeping support: study guide
Generated	2026-09-15T15:04:50.834Z
Fixture/version	apt-317-bookkeeping-support
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 Bookkeeping support

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-317-bookkeeping-support&mode=guided
- Start mini-test: https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-317-bookkeeping-support&mode=mini
- Start full simulation: https://learn.novusstreamsolutions.com/aptitude/practice/new?bank=apt-317-bookkeeping-support&mode=full

Skills covered, in full

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

Numerical reasoning

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

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

What you should be able to do after this lesson:

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

Worked examples and pitfalls

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

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

explicit.

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

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

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

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

How to practise this skill

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

- Keep an error log with four columns (misread the question, wrong operation, unit slip, arithmetic slip), and tally which one you actually commit. Three sessions is usually enough to show that one column dominates, and that is the column to train.
- Work untimed until your method is stable, then add the clock in stages: generous, then realistic, then 20 percent tighter than the real test. Attempts stay on this device, so a slow first pass costs nothing.

Glossary

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

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

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

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

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

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

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

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

Study this next

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

Where this material comes from

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

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

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

Data interpretation

Tables, charts, graphs, dashboards, trends, comparisons, and evidence-based conclusions.

Data interpretation is the discipline of getting a correct number out of a table, chart or dashboard that was not built to make your question easy, and of saying so when the data cannot answer it at all. It dominates graduate and analyst screening, and it is the section where strong arithmetic still fails, because the marks are lost in the header row, the axis scale and the wording of the question. The same skill is the daily work of anyone who reports on a management pack, a clinical audit or a stock ledger. Practice is stored on this device only; there is no account and nothing is uploaded unless you export it.

What you should be able to do after this lesson:

1. Read a value correctly out of a table or chart including its units multiplier, footnotes and any 'excludes' or 'provisional' qualifier attached to the row.
2. Compute share of total, percentage change and percentage-point change from the same pair of cells, identify which of the three a question is asking for, and separate a movement in a rate from a movement in the underlying count.
3. Work with index numbers relative to a base year, including why a change of five index points is almost never a five percent change.
4. Join two tables on a shared key and produce a normalised figure (per head, per unit, per thousand) rather than comparing raw totals.
5. Recognise chart presentation effects (truncated axes, dual axes, cumulative versus periodic series), and answer from the numbers rather than from the visual impression.
6. Apply the 'cannot say' discipline: state precisely which extra fact would be needed before the question becomes answerable.

Worked examples and pitfalls

Read the header: (£000) changes every answer by a factor of a thousand: Table titled 'Regional revenue, year to March (£000)': North 1,240; South 986; East 1,455; West 719. Total = 1,240 + 986 + 1,455 + 719 = 4,400, so the business turned over 4,400 thousand pounds, that is 4.4 million. East's share is 1,455 / 4,400 = 33.1 percent. The whole set: North 28.2 percent, South 22.4, East 33.1, West 16.3, summing to 100. Two things go wrong here. The first is reading East's revenue as 1,455 pounds and then reporting a business with a total turnover of 4,400 pounds, which nobody notices because every option is scaled the same way, until the question asks for revenue in millions and only one option is right. The second is the comparison wording. East's share is (1,455 - 719) / 4,400 = 16.7 percentage points above West's, and East's revenue is (1,455 - 719) / 719 = 102 percent more than West's, that is slightly more than double. 'Sixteen point seven' and 'a hundred and two' both describe the same two cells honestly, and the question decides which one is correct. Note that the percentage-point figure must be computed from the unrounded shares rather than by subtracting the rounded ones, or the last digit will not survive.

One pair of rows, three correct increases: A complaints table: 2023, 120,000 orders, complaint rate 4.0 percent; 2024, 150,000 orders, complaint rate 5.0 percent. Three defensible answers to 'how much did complaints increase?'. The rate rose by 1.0 percentage point. The rate rose by (5.0 - 4.0) / 4.0 = 25 percent

in relative terms. And the count of complaints rose from 0.04 x 120,000 = 4,800 to 0.05 x 150,000 = 7,500, which is (7,500 - 4,800) / 4,800 = 56.25 percent. All three are arithmetically right; only one answers the question in front of you. The pattern to internalise is that a rate and a count move together only when the denominator is fixed, and here it is not. Order volume grew 25 percent as well. If the question is about customer experience, the rate is the honest figure; if it is about how many complaint handlers to hire, the count is. Test items usually ask for the one you would not have chosen.

Index numbers: five points is not five percent: A cost index with 2020 = 100 reads 104 in 2021, 111 in 2022 and 109 in 2023. From 2021 to 2023 the index rose 5 points, but the percentage change is 5 / 104 = 4.8 percent, because the base for the comparison is 104, not 100. From 2022 to 2023 it fell 2 points, which is -2 / 111 = -1.8 percent, and note that costs fell even though the index remains 9 percent above the 2020 base. A level and a change are different claims. The only comparison where points and percent coincide is against the base year itself: 2020 to 2023 is 100 to 109, exactly plus 9 percent. Watch also for a rebased series, where a table switches to 2022 = 100 partway down; the two segments cannot be compared directly without converting one of them, and an item that quietly rebases is testing whether you read the column heading.

Joining two tables: totals and per-head figures disagree on purpose: Table 1, headcount by site: Leeds 84, Derby 47, Bristol 129. Table 2, absence days recorded in the same period: Leeds 630, Derby 300, Bristol 903. 'Which site has the worst absence problem?' On raw totals Bristol is worst at 903 days. Normalise per head and the ranking changes: Leeds 630 / 84 = 7.50 days per employee, Bristol 903 / 129 = 7.00, Derby 300 / 47 = 6.38. Leeds is worst, Bristol is merely biggest. The organisation-wide figure is 1,833 / 260 = 7.05 days per head, which is a useful reference line: Leeds is above it, the other two below. The general rule is that any comparison between units of different size demands a denominator, and the denominator has to come from the other table. Items are built so the raw-total answer and the per-head answer are both on the option list, and so the site with the biggest total is never the site with the highest rate.

The truncated axis: measure the numbers, not the bars: A quarterly satisfaction chart with a y-axis running from 78 to 82 shows bars at 79.2, 79.8, 80.4 and 81.1. Visually the last bar looks several times taller than the first, because only the top 4 points of a 100-point scale are drawn. The actual movement is 81.1 - 79.2 = 1.9 points, which on the score's own scale is a relative rise of 1.9 / 79.2 = 2.4 percent. If the question asks 'by approximately what percentage did satisfaction improve', the answer is about 2 percent, and the distractor built from the bar heights will be something like 40 or 400 percent. Related presentation effects to check before answering: a dual-axis chart where two series use different scales and appear to cross meaningfully when they do not; a cumulative series, where a flattening line still means the total is growing, just more slowly; and a logarithmic axis, where equal vertical distances are equal ratios rather than equal amounts. In every case the defence is the same. Find the printed numbers, or read the gridline values, and compute.

Cannot say: revenue is not profit: A product table shows units sold and total revenue. Product P: 4,200 units, 71,400 pounds. Product Q: 1,800 units, 41,400 pounds. Average selling price is 71,400 / 4,200 = 17.00 for P and 41,400 / 1,800 = 23.00 for Q, so Q earns more per unit while P earns more in total. Now the statement to evaluate: 'P is more profitable than Q.' The correct response is cannot say. Profit needs cost, and the table has no cost column; a product with a 17 pound price and a 16 pound unit cost is less profitable than one priced at 23 with a cost of 9, and nothing here rules that out. Contrast with 'Q generated more revenue per unit than P', which the table fully supports and which is true. The habit worth building is to finish every cannot-say judgement with the missing input named out loud ('cannot say, because unit cost is not given') because that forces you to distinguish a genuinely unanswerable item from one you simply have not worked hard enough on.

How to practise this skill

- Read the title, the units line, the row and column headers and any footnote before you look at a single value. Roughly the first fifteen seconds of an item should contain no arithmetic at all, and that fifteen seconds is what prevents the thousand-fold and percentage-point errors.

- For every item, write down which of the three quantities is wanted (share of total, relative change, or change in percentage points), before computing. Most wrong answers on this construct are correct arithmetic applied to the wrong quantity.
- Whenever two groups differ in size, ask what the denominator should be. If a question compares sites, teams, countries or periods of unequal length and you have not divided by something, you are almost certainly answering the wrong question.
- Practise the cannot-say items separately and force yourself to name the missing variable each time. Candidates who train only on computational items reliably over-answer inference statements under time pressure.
- Do not redraw or re-scale charts in your head. Locate the gridline values or the data labels, and if neither exists, interpolate between two labelled gridlines and state the bound rather than guessing a point value.
- Time yourself per item rather than per section. Data interpretation sets share a stimulus, so the first item costs the reading time and the rest should be fast; if item four takes as long as item one, you did not build a mental map of the table.

Glossary

Units multiplier: A scaling note in a table title or column header, such as (£000), (millions) or (per 1,000 population). It applies to every value in scope and is the single most common source of order-of-magnitude errors.

Index number: A series rescaled so a chosen base period equals 100. Changes between two non-base periods must be divided by the earlier value, so a movement in index points is not a percentage change except when measured from the base.

Rebasing: Restating an index against a new base period. Segments of a series with different bases cannot be compared directly, and a table that rebases partway down is testing whether you read the headings.

Truncated axis: A chart whose value axis does not start at zero, which exaggerates the apparent size of differences between bars or points. Legitimate for showing small movements in a large quantity, misleading if read as area or height.

Cumulative series: A line showing a running total rather than each period's value. It can only go up or stay flat, so a flattening cumulative line means the periodic figure is falling, not that the total is.

Weighted average: An average in which each value is multiplied by the size of the group it represents. Averaging two group percentages directly is only correct when the groups are the same size, which in these tables they rarely are.

Normalisation: Dividing a raw figure by an exposure measure (headcount, units sold, population, days open), so groups of different size can be compared. The denominator usually lives in a second table.

Cannot say: The verdict when a statement is neither supported nor contradicted by the data supplied. A correct cannot-say answer can always be defended by naming the specific missing variable.

Study this next

- Data interpretation skill hub: https://learn.novusstreamsolutions.com/aptitude/skills/data-interpretation
- Data interpretation practice in the Cognitive Skills Lab: https://learn.novusstreamsolutions.com/cognitive-skills/data-interpretation
- Numerical reasoning lesson: https://learn.novusstreamsolutions.com/aptitude/skills/numerical-reasoning/lesson
- Financial data interpretation suite: https://learn.novusstreamsolutions.com/aptitude/tests/financial-data-interpretation
- Scientific data interpretation suite: https://learn.novusstreamsolutions.com/aptitude/tests/scientific-data-interpretation

- Data interpretation lesson on Novus Learn: https://learn.novusstreamsolutions.com/aptitude/skills/data-interpretation/lesson

Where this material comes from

- Every table, index series and chart described above was constructed for Novus Learn, and each figure was verified by recomputing the totals and the reverse calculation.
- Definitions of index numbers, rebasing and weighted averages cross-checked against standard public references such as the Wikipedia articles 'Index (economics)' and 'Weighted arithmetic mean'. Terminology only; no data or item text is taken from any source.
- 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.

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.

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

- Build numerical reasoning: https://learn.novusstreamsolutions.com/aptitude/skills/numerical-reasoning/lesson
- Practice data interpretation: https://learn.novusstreamsolutions.com/aptitude/skills/data-interpretation/lesson

- Strengthen clerical accuracy: https://learn.novusstreamsolutions.com/aptitude/skills/clerical-accuracy/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/bookkeeping-support/study-outline