Fable 5 vs Opus 4.8 — Head-to-Head Test Kit

AIThinkerLab internal protocol · June 2026 · Run before June 22 (free window)

This kit contains everything for the 4-step protocol: two test briefs (one per Draft Economics mode), standardized revision prompts, the blind-scoring rubric, and the recording sheet. Paste prompts **identically** into both models — change nothing, not even punctuation.

⚙️ Test Conditions (control these or the numbers are meaningless)

1. **Fresh conversation** for each model — no prior context, no memory, no custom styles/preferences active.
2. **Default effort settings** on Fable 5 — do not touch the effort parameter.
3. **Web search OFF** for both models — both briefs include all source facts, so search would contaminate the comparison.
4. **Same session window** — run both within the same hour (server load varies by time of day).
5. **Start a stopwatch the moment you press send.** Lap 1 = first visible word. Lap 2 = generation complete.
6. **Record token counts** from the API console (or claude.ai usage panel) immediately after each response.

📝 TEST BRIEF #1 — Interactive Mode (the drafting test)

This is the ~800-word brief. Copy everything between the lines into both models.

You are a senior editorial writer for AIThinkerLab.com, an AI-tools publication read by content creators, indie developers, and technically-minded marketers. Write a complete, publication-ready blog article from the brief below. Follow every instruction exactly.

ARTICLE TOPIC: Prompt Caching: The 90% API Discount Most Claude Users Never Turn On

TARGET LENGTH: 1,400–1,600 words.

READER: A solo creator or small-team developer who uses the Claude API for content or product features, spends $50–$500/month, and has never configured prompt caching. They are technical enough to edit a JSON request but are not ML engineers.

SEARCH INTENT: Informational with commercial undertones — they want to understand it, then implement it today.

VOICE REQUIREMENTS: Second person. Conversational but substantive. Contractions. Vary sentence rhythm — alternate short punchy sentences with longer detailed ones. Take a clear stance where the facts support one. No hype words ("game-changer," "revolutionary," "unlock"). Never open consecutive paragraphs with the same word.

REQUIRED STRUCTURE:

1. TL;DR — 4 bullets, each containing a specific number or verdict (no teasers).
2. Introduction (100–130 words) — open with a concrete cost scenario, not a definition.
3. Six H2 sections, each 180–280 words, each opening with a direct answer to the implied question in its heading:
    - What prompt caching actually does (mechanism, not marketing)
    - The real math: what 90% off input tokens means at your spend level
    - The 5-minute setup (include one short code block showing the cache_control parameter)
    - The three mistakes that silently break caching
    - When caching does NOT help (be honest — list the genuine cases)
    - Caching strategy for content workflows specifically (system prompts, style guides, source documents)
4. FAQ — 4 questions, 40–70 word answers each, self-contained.
5. Closing section titled "The five-minute decision" (90–120 words) — one takeaway, one action, forward-looking last line.

FACT SHEET — use ONLY these facts for all claims. Do not invent statistics. If you need a fact not listed here, write [FACT NEEDED] instead of guessing:

- Prompt caching discount: cached input tokens cost 10% of the base input price (90% discount).
- Cache write cost: 25% premium over base input price on the first write.
- Claude Opus 4.8 base pricing: $5 per million input tokens, $25 per million output tokens.
- Claude Fable 5 base pricing: $10 per million input tokens, $50 per million output tokens.
- Cache reads on Fable 5: $1 per million tokens.
- Default cache lifetime: 5 minutes, refreshed on each use.
- Extended cache option: 1 hour, at a higher write cost.
- Minimum cacheable prompt length: 1,024 tokens on most Claude models.
- Cache breakpoints available per request: up to 4.
- A cache miss occurs if ANY content before the breakpoint changes, including one character.
- Caching works on: system prompts, tool definitions, conversation history, and document content.

- Caching does not reduce output token costs at all.

INTERNAL LINKS TO PLACE (descriptive anchors, weave naturally):

- /claude-fable-5-vs-opus-4-8-writers-hidden-costs (anchor about Fable 5's cost structure)
- /run-ai-models-locally-offline-privacy-guide (anchor about the zero-API-cost alternative)

HARD CONSTRAINTS:

- Every H2 section must be comprehensible if read alone.
- Exactly one code block in the entire article, inside the setup section, under 15 lines.
- No bullet lists longer than 4 items anywhere.
- The phrase "prompt caching" must appear in the first 50 words and in at least 2 H2 headings.
- Do not use em-dashes more than 4 times total.
- End the article with a single-sentence paragraph.

Write the complete article now. Do not ask clarifying questions. Do not include any preamble before the TL;DR or commentary after the final line.

🔁 STANDARDIZED REVISION PROMPTS (Brief #1)

Use these verbatim, in order, in the same conversation. A model "reaches publishable" when you would post its output with under 5 minutes of human edits. Record how many rounds each model needs.

Revision Round 1 (always run):

> Revise the article with these changes: (1) The introduction's cost scenario must use a specific monthly dollar figure and show the before/after caching numbers. (2) Cut total length by 10% without removing any H2 section — tighten, don't delete. (3) The "three mistakes" section must rank the mistakes from most to least common and say how to detect each one in the API response. Keep everything else unchanged. Return the full revised article.

Revision Round 2 (run only if Round 1 output still isn't publishable):

> Two problems remain. First, the FAQ answers read like summaries of the article — rewrite all four so each adds information not stated elsewhere in the piece. Second, find every sentence over 30 words and split or tighten it. Return the full revised article.

Revision Round 3 (run only if needed — failing to publish by Round 3 is itself a result):

> Final pass: fix the single weakest section of this article. Identify which section it is, state in one line why it's the weakest, then return the full article with only that section

> rewritten.

📚 TEST BRIEF #2 — Delegable Mode (the research test)

This tests the mode where Fable 5 should win. Prepare: copy the full text of THREE of your own published AIThinkerLab articles (suggested: the offline/local AI models pillar, the GraphRAG vs RAG guide, and the Qwen vs Gemma benchmark). Paste them where marked.

You are a research analyst for AIThinkerLab.com. Below are three published articles from our site, separated by ===ARTICLE=== markers. Your task has four parts. Complete all four in one response. Work carefully — accuracy against the source text matters more than speed.

PART 1 — STRUCTURED EXTRACTION: Build a table of every verifiable factual claim across all three articles. Columns: Claim · Article # · Whether the claim is dated/version-specific (Yes/No) · Risk of being outdated by December 2026 (Low/Medium/High with one-line reasoning).

PART 2 — CONTRADICTION AND OVERLAP AUDIT: Identify (a) any claims that contradict each other across the three articles, (b) any sections covering substantially the same ground in two or more articles, quoting the overlapping passages' locations by section heading.

PART 3 — SYNTHESIS: Write a 600-word executive briefing titled "What our local-AI coverage actually argues" that synthesizes the three articles' combined position into one coherent argument — including any tension between them. This must be a genuine synthesis: at least 4 sentences must connect ideas across two or more articles in ways no single article states.

PART 4 — PILLAR OUTLINE: Produce a section-by-section outline for one consolidated pillar post that would replace all three articles, marking for each section which article(s) it draws from and what new content would be required. Flag which existing article should become the canonical URL.

===ARTICLE 1=== [PASTE YOUR FIRST ARTICLE FULL TEXT HERE] ===ARTICLE 2=== [PASTE YOUR SECOND ARTICLE FULL TEXT HERE] ===ARTICLE 3=== [PASTE YOUR THIRD ARTICLE FULL TEXT HERE]

No revision rounds for Brief #2 — it's a walk-away task. Score the single response. This brief also stress-tests context handling: three full articles ≈ 15K–25K tokens of source material.

🎯 BLIND SCORING RUBRIC (for your second person — Janki)

Prepare: save each model's final output as "Draft A" and "Draft B" (flip a coin for labels; record the key privately). The scorer must not see model names, token counts, or timing.

Score each draft 1–10 on three axes:

STRUCTURE (1–10)

- Does every section open by answering its heading's implied question?
- Could each H2 section stand alone and still make sense?
- Did it follow the required structure exactly (count the bullets, check the constraints)?
- Deduct 1 point per violated hard constraint (Brief #1) or incomplete part (Brief #2).

VOICE (1–10)

- Read 3 random paragraphs aloud. Does it sound like a sharp colleague or a textbook?
- Sentence rhythm: actually varied, or uniform 20-word sentences?
- Any banned-phrase energy ("it's worth noting," "dive in," hype adjectives)?
- Would you recognize this as AIThinkerLab's voice next to a published article?

FACTUAL GROUNDING (1–10)

- Brief #1: check every number against the fact sheet. Deduct 2 points per invented statistic, 1 per misused fact. A [FACT NEEDED] marker where appropriate GAINS half a point (honesty signal).
- Brief #2: spot-check 8 random extracted claims against the source articles. Deduct 2 per claim not actually in the source.

Tiebreaker question (answer in one line): "Which draft would need less of your time before publishing, and why?"

📊 RECORDING SHEET

Brief #1 — Interactive Mode

Metric	Fable 5	Opus 4.8
Seconds to first word (initial)		
Minutes to complete draft (initial)		
Seconds to first word (Rev 1)		
Revision rounds to publishable (1–3, or DNF)		
Total input tokens (all rounds)		

Metric	Fable 5	Opus 4.8
Total output tokens (all rounds)		
Total cost (tokens × rate)		
Cost per finished draft = total cost (it IS one draft)		
Time per finished draft (sum all waits + generation)		
Blind score — Structure		
Blind score — Voice		
Blind score — Factual grounding		
Blind total /30		

Brief #2 — Delegable Mode

Metric	Fable 5	Opus 4.8
Seconds to first word		
Minutes to complete		
Total tokens (in / out)		
Total cost		
Parts fully completed (of 4)		
Claims spot-check accuracy (x/8)		
Blind total /30		

Rates for cost math (June 2026)

- Fable 5: input $10/MTok · output $50/MTok
- Opus 4.8: input $5/MTok · output $25/MTok
- (Thinking tokens bill as output on Fable 5 — they'll show in your console totals automatically.)

✍️ What goes back into the article

Map results straight to the three placeholders:

- **[INSERT YOUR TEST RESULT: time-to-first-word]** → Brief #1 row 1, both models
- **[INSERT YOUR TEST RESULT: tokens & cost per finished draft]** → Brief #1 cost rows
- **[INSERT YOUR TEST RESULT: blind ranking + rationale]** → Blind totals + Janki's tiebreaker line, quoted

The headline finding to watch for: if Opus 4.8 wins Brief #1 on time and cost while Fable 5 wins Brief #2 on completeness and accuracy, your Draft Economics framework is validated by your own data — that's the strongest possible originality signal for the article and for AdSense review.