



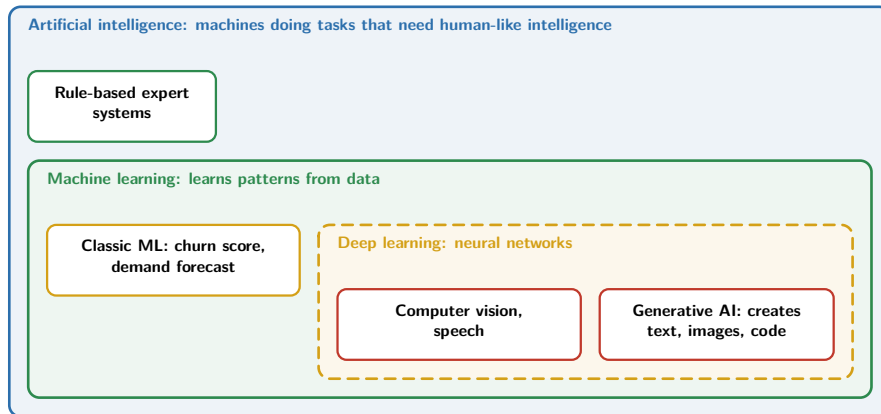
AIB-C01 Domain 1: AI Fundamentals and Literacy

AIB-C01 exam guide Domain 1 (24 %), ISO/IEC 22989 and 23053 vocabulary, as of 2026 | 1.1 AI concepts and data for business readers; 1.2 rules vs AI, agents, drift, shadow-AI classification; 1.3 prompting, tokens and context windows, RAG vs fine-tuning

AIB-C01 Domain 1 AI literacy Agents Shadow AI

1.1 Core AI concepts for business

AI ⊃ ML ⊃ deep learning ⊃ generative AI



Concept	Business reading	Cost and risk angle
Algorithm	the recipe used to learn (e.g. gradient-boosted trees)	choice is the data team's job, out of scope
Model	the learned result you deploy	an asset to version, own and retire
Training	learning from historical labelled examples	one-off or periodic cost; needs representative data
Inference / prediction	using the model on new cases	recurring cost per call or per hour
Label	the known outcome in history (churned yes/no)	missing or biased labels → wrong model
Feature	an input the model uses (tenure, spend)	proxies for protected traits create bias
Foundation model (FM)	large pre-trained GenAI model reused for many tasks	pay per token; little control of training data

Structured data	Unstructured data
tables: ERP, CRM, transactions	contracts, emails, call recordings, images
classic ML: forecast, score, classify	GenAI, NLP, vision, document extraction
quality = complete, accurate, consistent, current	quality = relevant, de-duplicated, permissioned, up to date

EXAM TIP

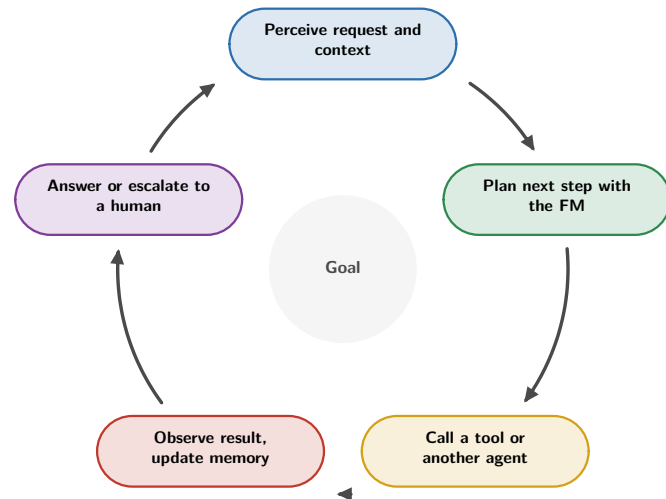
Data quality is the most frequent root cause in Domain 1 scenarios: 'garbage in, garbage out'. Training uses historical data, so a model reproduces past patterns and past bias. Vocabulary standards to name: ISO/IEC 22989 (AI concepts and terminology), ISO/IEC 23053 (framework for ML-based AI systems), ISO/IEC 42001 (AI management system).

1.2 Select the right solution type

Rules, classic ML, GenAI or agent?

Step	Statement	Go to / result
1a	The logic is stable and can be written as explicit if-then rules	Rule-based automation (workflow, RPA)
1b	The logic must be learned from examples or content	→ go to 2
2a	Output is a number, a class or a ranking from structured data	Classic ML (forecast, score, recommend)
2b	Output is content: text, summary, image, code, answer	→ go to 3
3a	One request, one answer, no actions in other systems	GenAI assistant (prompt, RAG)
3b	Multi-step goal needing tools, APIs, decisions	AI agent (with guardrails and human oversight)

What makes an agent: the agent loop



Agent capability	What it means for the business
Autonomy	acts without a human per step: define what it may do alone and when it must escalate
Tool use	calls APIs, databases, other apps: permissions = business risk
Agent-to-agent communication	specialised agents hand off work (A2A protocol, MCP for tools)
Orchestration	a supervisor agent or a fixed workflow decides the order of steps
Memory	remembers a session or a customer: privacy and retention rules apply

Shadow-AI control: a transparent tool register (task 1.2.4)

Approved	Amazon Quick (enterprise data, SSO) Bedrock apps built in-house Coding assistant with IP indemnity
Under evaluation	Meeting note-taker Image generator for marketing
Blocked	Free consumer chatbots for customer data Browser plug-ins sending pages to unknown APIs

- **Why monitor AI after launch (1.2.3)** data drift (inputs change), concept drift (the relation changes, e.g. new fraud patterns), FM version changes; plan budget and owners for monitoring, retraining and re-evaluation
- **Shadow AI fix** publish the register, give an approved alternative, train staff, monitor; banning without an alternative drives usage underground

⚡ 1.3 Generative AI techniques

- 1 **Role and task** 'You are a claims assistant. Summarise this claim for an adjuster.'
- 2 **Context** paste or retrieve the relevant facts; the model does not know your business
- 3 **Format and constraints** length, tone, bullet list, JSON, language, 'say I don't know if unsure'
- 4 **Examples (few-shot)** one or two good answers to imitate
- 5 **Iterate and save** test on real cases, keep the best prompt as a template

Prompt engineering principles a business user applies

Will it fit in the context window?

FORMULA $T = P + D \times K + H + A$

WHERE

- T = tokens needed in one request (tokens)
- P = system prompt and instructions (tokens)
- D = tokens per retrieved document chunk (tokens)
- K = number of chunks sent (count)
- H = conversation history kept (tokens)
- A = maximum answer length reserved (tokens)

WORKED

P = 1500
D = 800
K = 10
H = 6000
A = 1500
 $T = 1\,500 + 800 \times 10 + 6\,000 + 1\,500$
T = 17 000 tokens

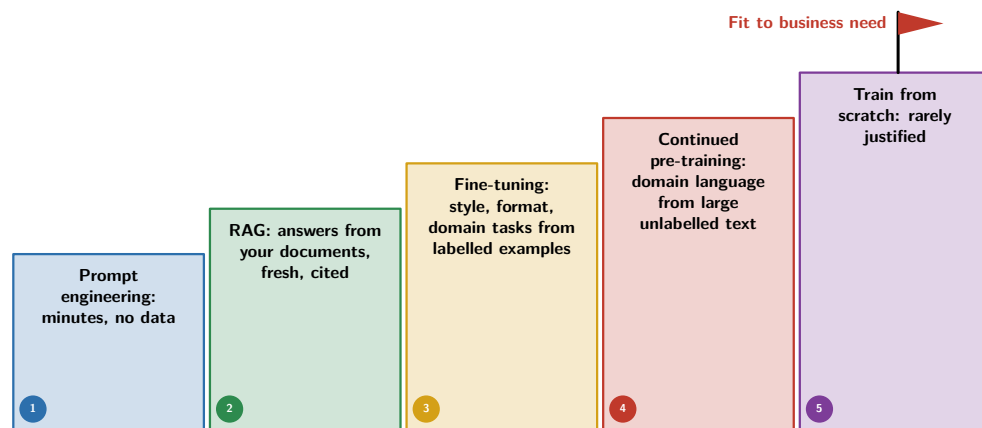
RESULT T = 17 000 tokens

READ IT 17,000 tokens fits a 200,000-token window, but every token is billed: trimming history and K cuts cost and latency.

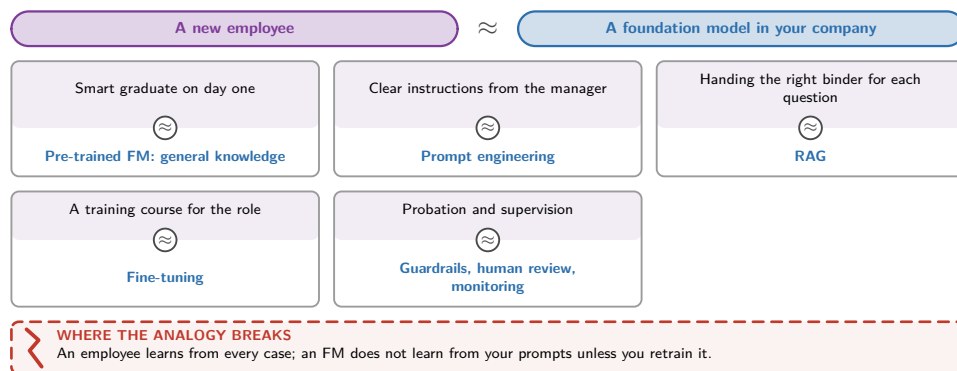
MISTAKES

- A 300-page manual (~150,000 tokens) in every prompt is slow and costly: use RAG
- Long contexts can bury facts in the middle: retrieve less, better

Adapting a model, from cheapest to costliest



RAG	Fine-tuning
adds knowledge at question time	changes model behaviour or style
data changes daily → re-index only	data changes → retrain
citations to sources, easier to audit	no citations, knowledge baked in
Bedrock Knowledge Bases	Bedrock model customisation, SageMaker AI



Recap and self-test

YOU MAY HEAR "AI is always smarter than rules" BUSTED ✓ ACTUALLY Stable, explainable logic is cheaper and safer as rules Tax rate lookups need rules, not a model.	YOU MAY HEAR "A model learns from every conversation" BUSTED ✓ ACTUALLY Inference does not change the model Only retraining or fine-tuning changes it; RAG changes what it reads.
YOU MAY HEAR "Once deployed, a model keeps its accuracy" BUSTED ✓ ACTUALLY Drift degrades it Budget monitoring and retraining from day one.	YOU MAY HEAR "Banning ChatGPT-style tools ends shadow AI" BUSTED ✓ ACTUALLY Usage moves underground Offer approved tools and a clear register.

Tick a box only when you could explain the point to someone else

- ☐ **AI ⊃ ML ⊃ DL ⊃ GenAI**
pick the narrowest that solves the problem
- ☐ **Rules when logic is stable**
agents when goals need tools and steps
- ☐ **Prompt → RAG → fine-tune**
climb only when needed
- ☐ **Data quality first**
historical data carries past bias
- ☐ **Tokens = cost and limits**
context window holds prompt + docs + history + answer

Q1 Difference between training and inference in cost terms?

A Training is a one-off or periodic cost to build the model; inference is paid on every use.

Q2 When should you prefer rule-based automation?

A When the logic is stable, explicit and must be fully explainable (fixed thresholds, regulations).

Q3 Four core capabilities of an AI agent?

A Autonomy, tool use, agent-to-agent communication and orchestration (plus memory).

Q4 Three classes of a shadow-AI tool register?

A Approved, under evaluation, blocked.

Q5 Company policies change weekly; the chatbot must cite them. RAG or fine-tuning?

A RAG: fresh documents at question time with citations, no retraining.

Q6 The model must always answer in the brand's tone and format. Technique?

A Prompt templates first; fine-tuning if prompts are not enough.

Q7 Why does a context window matter to a manager?

A It caps how much text fits in a request, and every token in it is billed and adds latency.

Q8 What is model drift?

A Performance decay because data or behaviour changed after training; detected by monitoring.

Q9 Which ISO standard certifies an AI management system?

A ISO/IEC 42001.

Q10 Unstructured data examples?

A Emails, contracts, PDFs, images, call audio, video.