MIT | 6.S978 AI Engineering

Quiz 1

Wednesday, September 23, 2026 – 25 minutes

Name: _____

Kerberos: _____

Question	Score	Out of
1		4
2		3
3		4
Bonus		+4
Total		**11**

- The quiz is closed book and closed notes.
- Write your answers in the spaces provided. Short answers are fine.

Please do not turn this page until you are told to begin.

1. When to call a larger model (4 points)

You're helping an online service route customer messages to one of 7 support channels. You test a small model and a large model on a **fixed pool of 100 messages**.

The small model produces two structured fields: a predicted `channel` and a Boolean `needs_help` that indicates whether it's not sure. Of the 100 messages, it **marks 20 messages as `needs_help`.**

The online service gave you a correct label for each of the 100 messages. You evaluated both models with greedy decoding, such that their responses are deterministic, and obtained the following results:

	Small model correct	Large model correct
All 100 messages	80	90
The 20 marked *needs help*	8	12

A call to the **small model costs 1 cent**, whereas a call to the **large model costs 10 cents**. Inspired by this tradeoff, you build this program:

```
channel, needs_help = small(message)
if needs_help:
    channel = large(message)
return channel
```

(a) 2 points. How many of the 100 messages does this program route correctly? What's the total cost?

(b) 2 points. The CEO is really impressed by this cost-efficient system but wants to find a way to create a premium version of the service that guarantees 90% accuracy to major customers. He asks you to "just use the large model, since we already saw that its accuracy is 90%". You disagree on at least two key fronts. Explain both.

2. Checking a prescription (3 points)

An online pharmacy lets customers **log in and request medicine** to be shipped to them. The pharmacy's engineers have implemented two functions, which are internally tied to the logged-in customer's account:

- `prescriptions()` returns a *list of dictionaries*, one for each of the current customer's prescriptions. For example, it may return:

  ```
  [{"id": 17, "drug": "Drug A", "valid": True}, {...}, ...]
  ```

- `ship(drug)` ships the given `drug` to this customer.

The pharmacy is required by regulators to ship a drug only if the customer has a valid prescription for it. Given a request from a customer, the program starts by executing `prescriptions()` and storing the returned list in `records`. Then it tries to interpret and act on the customer's request.

Which designs or statements guarantee that the regulator's requirement is met? **Select all that apply.**

(A) The pharmacy conducted advanced RL training on the model to penalize hallucinations, and then commissioned rigorous independent testing which found that their model made no unsafe orders in 1 million adversarial requests. Statistical analyses bound the rate of unsafe orders below 0.001% with 99.9% confidence.

(B) Given the customer's request, the model is only allowed to extract a `prescription_id` (int) and `requested_drug` (str) from the request, but can't execute actions. A simple piece of symbolic code checks that the ID matches a valid record in `records`. If it does, the code ships `requested_drug`.

(C) Given the customer's request, the model is only allowed to extract a `prescription_id` (int) and `requested_drug` (str) from the request, but can't execute actions. A simple piece of symbolic code checks that the ID matches a valid record in `records`. If it does, the code ships the record's `drug`.

(D) Code lists the drugs with valid prescriptions in `records`. The model is forced via constrained decoding to choose from this list or return "do not ship". Code ships the chosen drug, if any.

(E) The full list of `records` is inserted into the model's prompt and the model is forced via constrained decoding to return a string `drug` from `records` and a Boolean `valid`. Code ships that drug if the Boolean is true.

○ A ○ B ○ C ○ D ○ E

3. A course-policy assistant (4 points + 3 bonus)

An instructor maintains a webpage with the course policy. You build a program that reads that page and uses an LLM to answer students' questions. If the program offers a student an extension, the course staff will have to honor it.

(a) 4 points. Treat the LLM as an external component to your design. Your code sends it a prompt and receives free-form text, with no enforced output format.

Label each statement **R**, **A**, or **D**:

- **R**: a requirement about what the program should achieve.
- **A**: an assumption you have about the external world.
- **D**: a design choice about what your system does.

(i) _____ The policy page is correct and current.

(ii) _____ Code includes the policy page in every prompt.

(iii) _____ The model always interprets the student's question correctly.

(iv) _____ Replies must answer the student's question or direct them to the staff.

(b) Bonus: 2 points. A colleague argues you can't use an LLM for this problem and says "LLMs are stochastic. If an LLM helps answer the questions, you have to accept that your system will sometimes make something up, no matter how you design it." Do you agree?

You may redesign the system in any way you wish, as long as (1) it intelligently gives a relevant reply to students as often as you can make it and (2) it NEVER *hallucinates* (makes up) course information. A degenerate system that always replies with the whole course policy or with "ask the staff" is not enough.

○ A design is possible ○ These goals are impossible

If you chose **possible**, describe your design and why it works. Otherwise, explain which goal cannot be met and why. You need no more than 2 or 3 sentences.

3(c). Bonus: course-policy assistant (1 point)

If you chose possible: Do you need the instructor to check anything about the course policy material your design uses, *beyond whether the facts are correct*? Give an example of how your system could mislead a student if this check were skipped.

If you chose impossible: A colleague uses one LLM to write a reply and a second to check its facts against the policy one by one. Code shows only replies the second LLM approves. Does this overcome your objection? Explain how it could still fail OR why it cannot.

4. Bonus: shortening a conversation (1 point)

An agent's conversation is getting long. It asks the model to summarize the conversation so it can use the summary in later calls. This is called **compaction**.

Compare two layouts below (Prompt A vs. B) for the *compaction step itself*. The quality observed from both is equally good. The system instructions and conversation are identical in all three rows.

Previous prompt

`You are a ...` System instructions 1,000 tokens	Conversation 9,000 tokens

Prompt A

`Summarize ...` Compaction request 100 tokens	`You are a ...` System instructions 1,000 tokens	Conversation 9,000 tokens

Prompt B

`You are a ...` System instructions 1,000 tokens	Conversation 9,000 tokens	`Summarize ...` Compaction request 100 tokens

A teammate says, "At the infra level, these two calls have the same input cost. Both send 10,100 tokens through the Transformer." What does this argument miss? Explain in one or two sentences.