# Final

## Note

This is an open-book exam. You may consult any book, paper, note, or on-line resource, excluding paid variants of ChatGPT or the like; however, discussions with others (in person or via a network) are strictly forbidden.

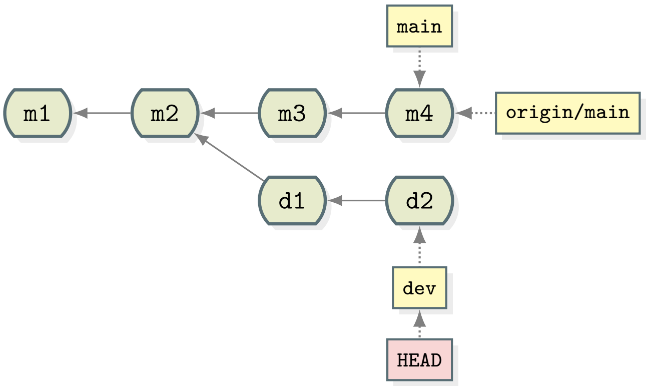
## Problems

1. Two notations are assumed.

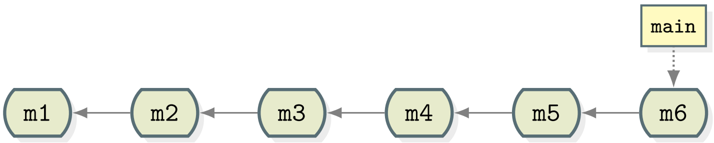
   - A commit $c'$ denotes a commit same as the commit $c$ but with a different parent.
   - A commit $c_0 c_1 \ldots c_n$ exactly consists of all changes in the commits $c_0$, $c_1$, …, $c_n$.

   Answer the following questions.

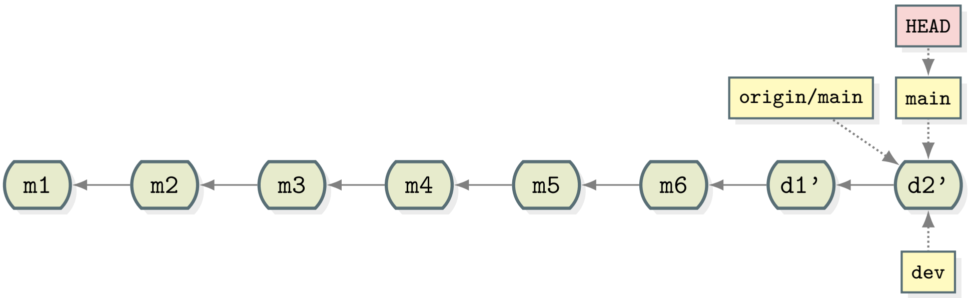
   (a) (10 %) Consider a local git repository with two branches, namely main and dev (pointed to by HEAD).



The upstream of the main branch is origin/main (i.e., the main branch on the remote server origin). The repository on origin is shown below.

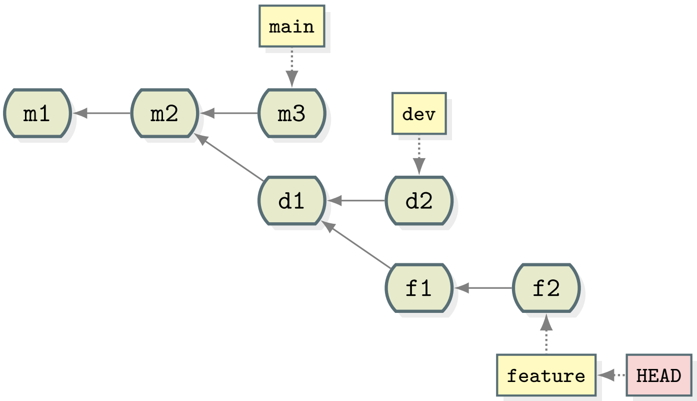


After some git commands, the local repository becomes:

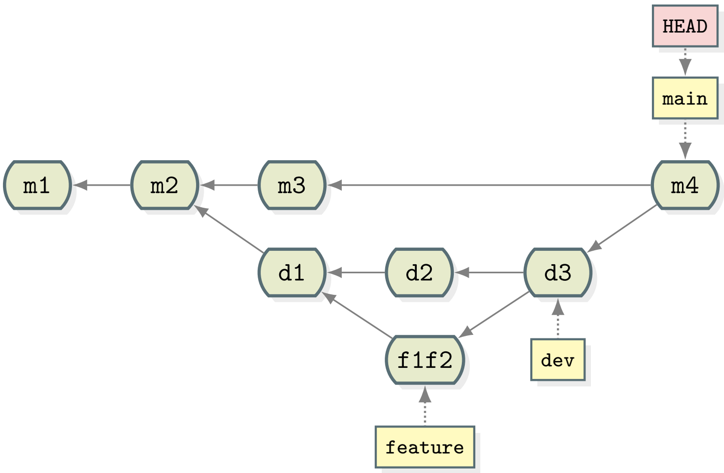


What git commands are executed? Write down the necessary git commands in order.

(b) (10 %) Consider a git repository with three branches, namely main, dev, and feature (pointed to by HEAD).



What are the necessary git commits to make the repository become the following one.



2. You are working on an online service that allows the user to input an image of handwritten questions. The system answers the user's question as speech. The system has the following interfaces and concrete implementation classes of image-to-text, chatbox service and text-to-speech functions:

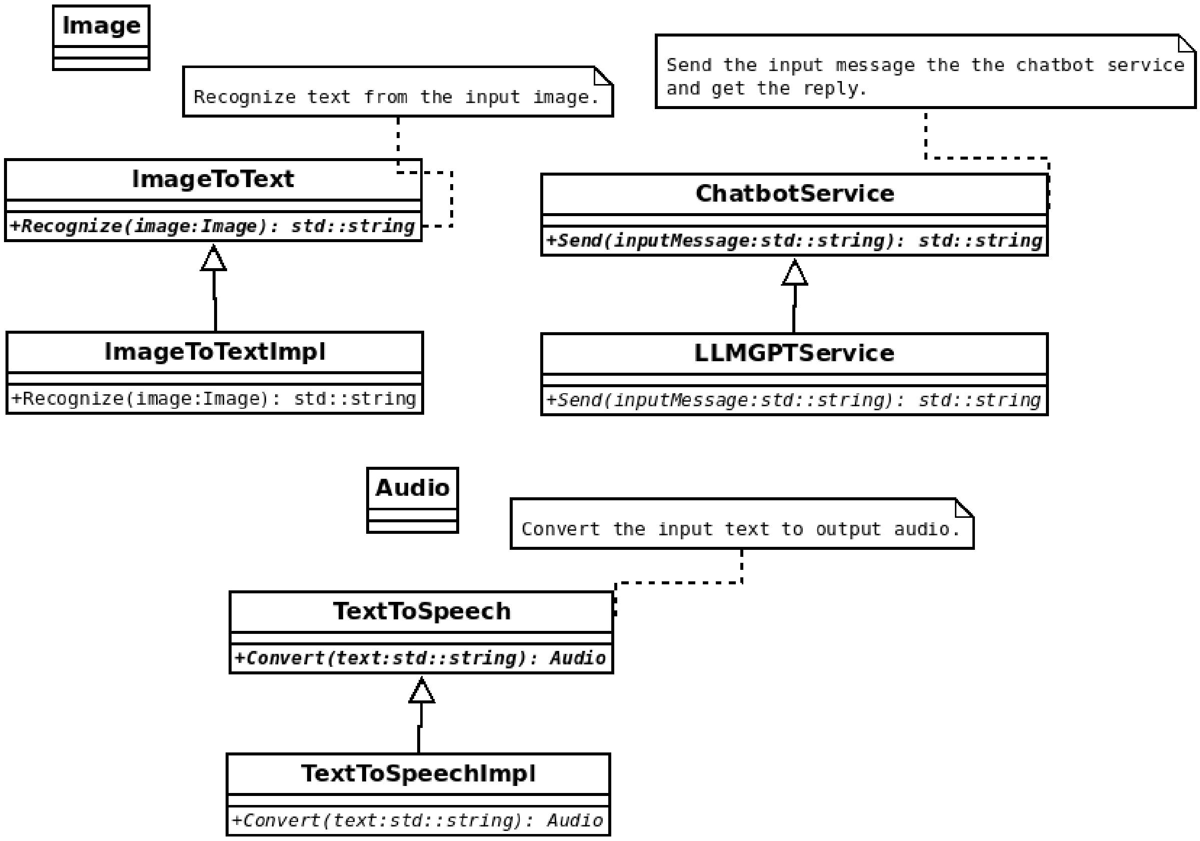


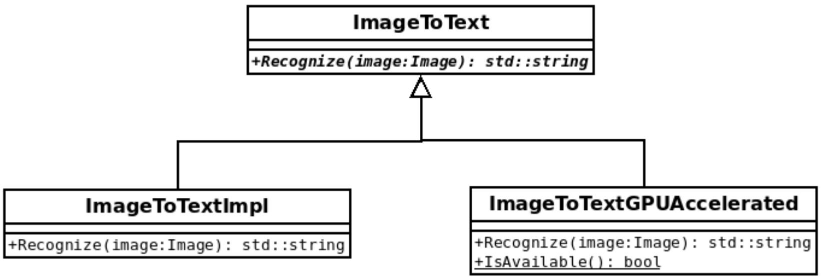
Image
Recognize text from the input image.
ImageToText
+Recognize(image:Image): std::string
ImageToTextImpl
+Recognize(image:Image): std::string
Send the input message the the chatbot service and get the reply.
ChatbotService
+Send(inputMessage:std::string): std::string
LLMGPTService
+Send(inputMessage:std::string): std::string
Audio
Convert the input text to output audio.
TextToSpeech
+Convert(text:std::string): Audio
TextToSpeechImpl
+Convert(text:std::string): Audio

Suppose you want to make it easier for other services to use your image-to-speech answering service. It's a good idea to provide your service as a single interface so that the user doesn't need to interact with the individual classes.

(a) (3 %) How can design patterns help solve this design problem?

(b) (3 %) Please provide your design in a UML class diagram.

3. After your service went online, you found that the pure-software implementation of ImageToTextImpl is too slow and consumes too much CPU time on your servers. You decided to accelerate text recognition using graphics hardware (GPU). The feature is provided by class ImageToTextGPUAccelerated:


ImageToText
+Recognize(image:Image): std::string
ImageToTextImpl
+Recognize(image:Image): std::string
ImageToTextGPUAccelerated
+Recognize(image:Image): std::string
+IsAvailable(): bool

Not all of your servers are equipped with the GPU hardware. Which image-to-text implementation class to use depends on the result of ImageToTextGPUAccelerated::IsAvailable().

In a sprint you managed to ship this improvement on time, but the code creates future maintenance problems. Your code base contains lots of snippets like:

```
ImageToText* imageToTextConverter =
    ImageToTextGPUAccelerated::IsAvailable() ?
        new ImageToTextGPUAccelerated() :
        new ImageToTextImpl();
```

(a) (3 %) What's the problem with the above design? How can design patterns help solve the maintenance problem?

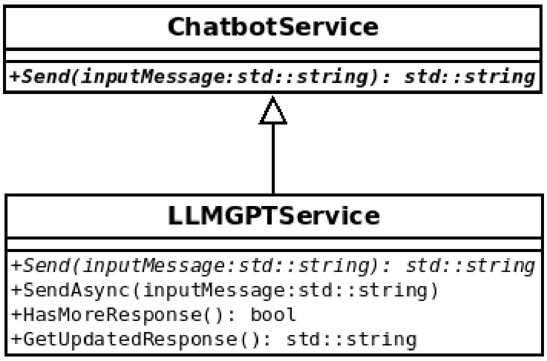
(b) (4 %) Please provide your solution in a class diagram.

4. You then find that the synchronous interface of ChatbotService isn't good for user experience: it may take 30 seconds or longer for the implementation class, LLMGPTService, to respond to a question. This means the user starts to hear the answer only after the full response is returned from the interface.

   You notice that LLMGPTService provides an interface that allows you to get partial results. For example, after you send a message "What is virtual memory?", it can provide partial results like:

   "Virtual memory is a technique used in computer operating system " after 1 second,
   "that allows a process to think " after 2 seconds
   "that it can use the full contiguous address space " after 3 seconds and so forth until there is no more update.

   It provides this feature in the following interface:


ChatbotService
+Send(inputMessage:std::string): std::string
LLMGPTService
+Send(inputMessage:std::string): std::string
+SendAsync(inputMessage:std::string)
+HasMoreResponse(): bool
+GetUpdatedResponse(): std::string

(a) (3 %) How can design patterns be used to allow the user of the ChatbotService to get updates on new responses from the remote chatbot service?

(b) (4 %) Please apply the pattern to LLMGPTService in code or a UML diagram

5. (10 %) Why do we need both verification and validation in an adequate software development process? Please explain the reasons using concrete examples from YOUR TERM PROJECT.

6. (20 %) Construct an abstract data model for the new sales-and-services system of a major-appliance chain store that has to meet the following requirements:

   - The chain store has tens of locations (which may change over time).
   - It sells major appliances such as refrigerators, washers, ovens, air conditioners, etc. of various brands and models.
   - A customer may place an order (of purchase or after-sale service) with any location of the store.
   - To provide better after-sale services, the purchase order of a machine by a customer should be kept along with the customer's contact information such as phone number and home address (which may change over time).
   - It should be convenient to view all the purchases and the after-sale services ordered by a particular customer.

   You should avoid many-to-many relationships; otherwise, you must provide a verbal explanation for including such relationships. Please use the UML as much as possible when describing the model. State the assumptions, if any, you make for your construction.

7. (10 %) We have considered in class an example (two signs at the entrance of an escalator) showing ambiguity of natural language (English). Please give another example in English or Chinese and show how logic formulae may help clarify/reveal the ambiguity. Your example must have two sentences with the same syntactic structure but different semantical structures and meanings. Is the logic you use typed or untyped? Why?

8. (20 %) Please examine the following C function sumofMM and give a suitable function contract, namely the pre and post-conditions, for sumofMM, using either ACSL or the conventional logic notation. Be careful that the function has made an implicit assumption about the size of the input array. You may omit the condition concerning proper memory allocation for the input array and focus on what the function does. Let us assume that, for this problem, the type int is the same as the set $\mathbb{Z}$ of integers. The formulae you will write are intended for the semantic structure $\mathcal{Z} = (\mathbb{Z}, \{a[\,], +, -, *, 0, 1, 2, <\})$; "=" is implicitly assumed to be a binary predicate as usual.

```
int sumofMM (int* a, int n)
{ int min, max, i;

  // Initialize min and max.

  if (a[0] < a[1]) {
    min = a[0];
    max = a[1];
  }
  else {
    min = a[1];
    max = a[0];
  }

  // Divide the rest into pairs.
  // Compare the smaller with min and the larger with max.

  for (i=2; i<n; i=i+2)
    if (a[i] < a[i+1]) {
      if (a[i] < min)
        min = a[i];
      if (a[i+1] > max)
        max = a[i+1];
    }
    else {
      if (a[i+1] < min)
        min = a[i+1];
      if (a[i] > max)
        max = a[i];
    }

  return min+max;
}
```