Part 1

1. Software architecture is often compared to the architecture of buildings as a conceptual analogy. What are the strong points of that analogy? What are the weaknesses of the analogy?
2. How an architecture serves as a basis for analysis? What about decision-making? What kinds of decision-making does an architecture empower?
3. What’s architecture’s role in project risk reduction?
4. In the chapter 2 there were described 13 reasons why architecture is important, write more than 3 of them which you think is best explains the importance of an architecture. Explain your choice.
5. Suppose you want to introduce architecture-centric practices to your organization. Your management is open to the idea, but wants to know the ROI for doing so. How would you respond?
6. In the 3rd there were given 4 dominant software development processes. Explain each of them.
7. Chapter 3 describes 7 activities required to be done in creating software architecture, using that architecture to realize a complete design, and then implementing or managing the evolution of a target system or application. Describe these activities.
8. Name and explain 4 different contexts in which architecture exists.
9. No matter the source, all requirements encompass the 3 categories. Name these categories and give an explanation to each of them and how an architecture responds to them.
10. Name and give an explanation to quality attribute requirements.
11. Draw and explain the general scenario for availability.
12. In Chapter 4 we covered 7 categories of design decisions. Name and explain each category.
13. Consider the choice between synchronous and asynchronous communication (a choice in the coordination mechanism category). What quality attribute requirements might lead you to choose one over the other?
14. What is Hazard analysis? Name and explain categories of hazards.
15. Recover-from-faults tactics are refined into preparation-and-repair tactics and reintroduction tactics. Explain each tactic and give examples.
16. Write a concrete availability scenario for the software for an (hypothetical) online hotel and room reservation.
17. How does availability trade off against modifiability? How would you make a change to a system that is required to have "24/7'' availability (no scheduled or unscheduled downtime, ever)?
18. If you are a technology producer, what are the advantages and disadvantages of adhering to interoperability standards? Why would a producer not adhere to a standard?
19. Consider a system of automatic teller machine. With what other systems will an automatic teller machine need to interoperate? How would you change your automatic teller system design to accommodate these other systems?
20. What are the 4 main tactics of Modifiability? Name and explain each tactic.
21. In a certain metropolitan subway system, the ticket machines accept cash but do not give change. There is a separate machine that dispenses change but does not sell tickets. In an average station there are six or eight ticket machines for every change machine. What modifiability tactics do you see at work in this arrangement? What can you say about availability?
22. Once an intermediary has been introduced into an architecture, some modules may attempt to circumvent it, either inadvertently (because they are not aware of the intermediary) or intentionally (for performance, for convenience, or out of habit). Discuss some architectural means to prevent inadvertent circumvention of an intermediary.
23. What are main Performance tactics? Name them and give explanation to each of them.
24. Web-based systems often use proxy servers, which are the first element of the system to receive a request from a client (such as your browser). Proxy servers are able to serve up often-requested web pages, such as a company' s home page, without bothering the real application servers that carry out transactions. There may be many proxy servers, and they are often located geographically close to large user communities, to decrease response time for routine requests. What performance tactics do you see at work here?
25. A fundamental difference between coordination mechanisms is whether interaction is synchronous or asynchronous. Discuss the advantages and disadvantages of each with respect to each of the performance responses: latency, deadline, throughput, jitter, miss rate, data loss, or any other required performance-related response you may be used to.
26. User interface frameworks typically are single-threaded. Why is this so and what are the performance implications of this single-threading?

Part 2

1. What is virtual environment? What are the benefits of using virtual environments? How do you create virtual environment and activate it (write a command)?
2. What is Web Server Gateway Interface (WSGI)?
3. What is route and view functions and how are they defined (some code)?
4. What is the Request-Response cycle? Which modules are used to work with requests and responses in Flask?
5. What are templates? Explain how you render templates, what are Variables Control Structures in Jinja2?
6. What is the Web Form? Explain Cross-Site Request Forgery (CSRF) and protection against them.
7. How do you handle forms in view functions?
8. What is Database? What are the factors you should consider while choosing database framework?
9. While working with databases, what is the Model? What are the most common SQLAlchemy column options?
10. How do you define relationships between rows of different tables? Give an example of how the one-to-many relationship is represented.