A. From Ian Sommerville, _Software Engineering, 9th Edition_:

1. Explain what a _software process_ is and briefly describe the typical phases of a software process (specification, design, etc.).

2. Explain the differences between _plan-driven_ and _agile_ software processes.

3. Describe the _waterfall_ software engineering process model. Also, indicate its advantages, disadvantages, and applicability.

4. Describe the _reuse-oriented_ software engineering process model. Also, indicate its advantages, disadvantages, and applicability.

5. Describe the _incremental delivery_ software engineering process model. Also, indicate its advantages, disadvantages, and applicability.

6. Describe the _Boehm’s spiral model_ software engineering process model. Also, indicate its advantages, disadvantages, and applicability.

7. Describe _the Rational Unified_ software engineering process model. Also, indicate its advantages, disadvantages, and applicability.

8. Explain what is mean by _software specification_ and describe the activities involved in the requirements engineering process.

9. Explain what is meant by _software design_ and describe the activities involved in the software design.

10. Describe the _testing phases_ in a plan-driven software process.

11. Explain what is meant by _software evolution_ and describe the activities involved in the software/system evolution.

12. Explain what is meant by _software prototyping_ and indicate its benefits. Also, explain why _throw-away prototypes_ should be discarded.

13. Indicate what is meant by a _requirement_ and explain the difference between _user_ and _system_ requirements. Give an example for each user and system requirements.

14. Explain the difference between _functional_ and _non-functional_ requirements and give two examples for each.

15. Provide 5 types of _metrics (properties) for non-functional requirements_ and indicate their measuring units.
16. Describe and discuss 3 ways of writing a system’s requirements specification. Indicate your opinions on each of them.

17. Describe three types of requirements validation techniques.

18. Explain what is meant by a sequence diagram and give an example of a non-trivial such diagram (at least 3 objects involved).

19. Explain what is meant by a class diagram and give an example of a non-trivial such diagram (at least 5 classes in the diagram).

20. Explain what is meant by generalization and aggregation in object-oriented modeling and give an example for each.

21. Indicate what is meant by model-driven engineering and present its pros and cons.

22. Discuss the advantages of developing explicit software architecture.

23. Discuss how decisions on software architecture influence system characteristics (or non-functional properties) such as security, safety, performance, availability, and maintainability.

24. Describe the 4+1 view model of software architecture.

25. Describe the model-view-controller (MVC) architectural pattern. Also, indicate its advantages, disadvantages, and applicability.

26. Describe the layered architecture pattern. Also, indicate its advantages, disadvantages, and applicability.

27. Describe the repository architecture pattern. Also, indicate its advantages, disadvantages, and applicability.

28. Describe the client-server architecture pattern. Also, indicate its advantages, disadvantages, and applicability.

29. Describe the pipe-and-filter architecture pattern. Also, indicate its advantages, disadvantages, and applicability.

30. Briefly compare and contrast validation testing and defect testing.

31. Explain what is meant by software inspection and discuss its advantages.

32. Describe what is meant by development testing.

33. Describe interface testing and present at least 4 interface testing guidelines.

34. Explain what is meant by test-driven development and discuss its benefits.

35. Describe what is meant by release testing.

36. Describe what is meant by user testing.

37. Briefly describe the three major types of software maintenance.
38. Describe and discuss four maintenance cost factors.

39. Explain what is meant by software reengineering and describe the reengineering process activities.

40. Explain what is meant by refactoring and indicate how it differs from reengineering.

41. Give 5 examples of “bad smells” in program code.


1. Provide a UML metamodel for software requirements specification.

2. Explain why requirements are important and describe ways or sources for finding requirements.

3. Give an example of use case diagram with at least 6 use cases (briefly describe each of them), and provide a UML template-based description for one of the use cases.

4. Explain what is meant by a project glossary and why it is important in software development.

5. Give an example of a non-trivial UML activity diagram with at least 6 action nodes. Briefly describe the diagram.

6. Briefly describe the 4 types of action nodes used in UML activity diagrams.

7. Briefly describe the 7 types of control nodes used in UML activity diagrams. Give an example of activity diagram that includes all of them.

8. Indicate what is meant by provided and required interfaces and give a UML example for each of them.

9. Explain why interfaces are useful in component-based software engineering and discuss their benefits and limitations.

10. Describe the layering pattern used in software architecture.

11. Provide the UML notation for states and transitions (used in behavioral state diagrams). Give an example for each of them.

12. Give an example of a non-trivial UML state diagram with at least 5 states. Briefly describe the diagram.

13. Explain the difference between shallow and deep history used in UML state machines. Provide an example of state diagram that illustrates both of them.

C. From Presentations by Students (Focus areas).

1. What are the common issues in game development [Chris]?

2. What are the arguments for and against Model Driven Development [Ivan]?
3. Name and comment on three important areas for research in Software Engineering for mobile application development [Vamsi].

4. When it is a good idea to use MVC? When is it not a good idea? [John]

5. Describe the main concepts and ideas behind Aspect Oriented Software Development (AOSD) [Javier].

6. What are the goals and advantages of software visualization? [Likhitha]