OOAD (Object-Oriented Analysis and Design)

Short question answers:

1. What is Analysis and Design?

Ans: **Analysis** emphasizes an investigation of the problem and requirements, rather than a solution.

Design emphasizes a conceptual solution that fulfills the requirements, rather than its implementation.

2. What Is Object-Oriented Analysis and Design?

Ans: **Object-oriented analysis,** there is an emphasis on finding and describing the objects—or concepts—in the problem domain.

Object-oriented design (or simply object design) there is an emphasis on defining software objects and how they collaborate to fulfill the requirements.

3. Define Use Cases?

Ans: Requirements analysis may include stories or scenarios of how people use the application; these can be written as use cases.

4. Define a Domain Model?

Ans: Object-oriented analysis is concerned with creating a description of the domain from the perspective of objects. There is an identification of the concepts, attributes, and associations that are considered noteworthy.

The result can be expressed in a domain model that shows the noteworthy domain concepts or objects.

5. What is UML?

Ans: The Unified Modeling Language (UML) is a visual language for specifying, constructing, and documenting the artifacts of software systems,

6. Ways to apply UML?

Ans: Three ways to apply UML

- **UML as sketch** informal and incomplete diagrams (often hand sketched on whiteboards) created to explore difficult parts of the problem or solution space, exploiting the power of visual languages.
- **UML as blueprint** relatively detailed design diagrams used either for 1)reverse engineering to visualize and better understanding existing code in UML diagrams, or for 2)code generation(forward engineering)
- **UML as programming language** complete executable specification of a software system in UML)

7. Three perspectives to apply UML?

- Conceptual perspective
- Specification perspective
- Implementation perspective

9. Benefits of Iterative Development?

Ans: Benefits of iterative development include:

• Early rather than late mitigation of high risks (technical, requirements, objectives, usability, and so forth)

• Early visible progress

• Early feedback, user engagement, and adaptation, leading to a refined system that more closely meets the real needs of the stakeholders

• managed complexity; the team is not overwhelmed by "analysis paralysis" or very long and complex steps

• The learning within an iteration can be methodically used to improve the development process itself, iteration by iteration.

10. What is the UP?

Ans: The Unified process has emerged as a popular iterative software development process for building object.is also called rational unified process or RUP.

11. UP phase?

Ans: **1. Inception**— approximate vision, business case, scope, vague estimates.

2. Elaboration—refined vision, iterative implementation of the core architect true, resolution of high risks, identification of most requirements and scope, more realistic estimates.

3. Construction—iterative implementation of the remaining lower risk and easier elements, and preparation for deployment.

4. Transition—beta tests, deployment.

12. UP Disciplines?

Ans: **1. Business modeling—**when developing a single application, this includes domain object modeling.

2. Requirements—Requirements analysis for an application, such as writing use cases and identifying non-functional requirements.

3. Design—All aspects of design, including the overall architecture, objects, databases, networking,

14. Types of Requirements

Ans: **+FURPS?**

- **Functional** feature, capabilities, security
- **Usability** human factors, help, documentation
- **Reliability** frequency of failure, recoverability, predictability.
- **Performance** response time, throughput, accuracy, availability, resource usage.
- **Supportability** adaptability, maintainability, internationalization configurability.
- **+ in FURPS indicates ancillary and sub factors such as**
- **Implementation** resource limitations, languages and tools hardware
- **Interface** constraints imposed by interfacing with external systems
- **Operations** system management in its operational setting
- **Packaging** for example a physical box
- **Legal** licensing and so forth

15. What is user interface?

Ans: **User Interface** - graphical interface; windows.

A component is a physical and replaceable part of the system that conforms to and provides the realization of a set of **interfaces**. It represents the physical packaging of elements like classes and **interfaces** In UML diagrams, a component is represented by a rectangle.

16. What is logic application and domain object?

Ans: **Application Logic and Domain Objects—**software objects representing domain concepts (for example, a software class named Sale) that fulfill application requirements.

17. What is use case?

Ans: Use cases are text stories, widely used to discover and record requirements. They influence many aspects of a project. Use case is a collection of related success and failure scenario.
Use cases are not diagram, they are text.

18. Scenario?

Ans: A scenario is a specific sequence of actions and interactions between actors and the system.it is also called a use case instance.

19. Use case and use case model?

Ans: Use cases are text documents, not diagrams, and modeling is primarily an act of writing text, not drawing diagrams.

20. Use case Writing style?

Ans: **Brief—**terse one-paragraph summary, usually of the main success scenario.
The prior Process Sale example was brief.
Casual—informal paragraph format. Multiple paragraphs that cover various scenarios. The prior Handle Returns example was casual.
Fully dressed—the most elaborate. All steps and variations are written in detail, and there are supporting sections, such as preconditions and success guarantees.