# Repetition Questions (dt. Wiederholungsfragen) Lesson 4

## Topics and Concepts (Link to Lecture, via Fact Sheets)

The lesson of the lecture today covered the following concepts (see fact sheets in script folder):

1. Component Modeling (Identification, Specification)[1]
2. Layered Reference Architectures[2]

In the corresponding exercise[3], we worked with these concepts.

## Questions

### Topic/Concept: Component Modeling

1. What are the four component modeling steps (according to the lecture)? Which of them were covered in exercise 2?
2. What is the relationship between component modeling and the C4 model?
3. What are general component realization options/alternatives? List at least two.

### Topic/Concept: CRC Cards

1. What does CRC stand for?
2. What is the purpose and role of CRC cards in terms of level of abstraction, role, architecting phase?
3. What is their relationship to architectural patterns?

### Topic/Concept: Container Reference Architectures

1. What is the difference between a project-level software architecture and a reference architecture (purpose, level of detail)?
2. What is an application server?
3. Which architectural styles were mentioned in the lecture and in the exercise? List at least three.

### Topic/Concept: Java Enterprise Edition (JEE)

1. What are the main architectural elements shown in the architecture overview diagram for JEE shown in the lecture?
2. Name at least one JEE API per layer: presentation, business logic, data access.
3. Which architectural styles or patterns are realized by the JEE reference architecture? List at least two.

[1] ../2-lecture-script/lesson4/ZIO-ComponentModelingStepsTechniquesNotations.pdf
[2] ../2-lecture-script/lesson4/ZIO-ContainerReferenceArchitecturesFactSheet.pdf
[3] ../3-exercises-solutions/ZIO-AppArch-ExerciseWeek4.md

## Answers

### Topic/Concept: Component Modeling (Identification, Specification)

1. *Component identification, specification, realization, composition (the first two were covered).*
2. *C4 diagrams can serve as input (for instance, the context diagram; they can also represent and visualize its output (container and component diagram).*
3. *Buy, build; use operating system feature, use/enhance open source software*

### Topic/Concept: CRC Cards

1. *Components, Responsibilities, Collaborators; you also find Classes, Responsibilities, Collaborations or Candidates, Responsibilities, Collaborations*
2. *They are used in architectural synthesis (solution strategy) as a bridge between requirements and code to learn about and specify component interfaces and their usage*
3. *Patterns can be specified with CRC cards (see POSA 1); patterns can replace the need for CRC cards (no new design work is required if a pattern is chosen); they can also appear in CRC cards to keep the text short (but this has to be handled with care, as not every reader of the cards might know the pattern)*

### Topic/Concept: Container Reference Architectures

1. *Class-instance relationship; reference architecture is more abstract, much like a pattern, is a knowledge sharing and marketing instrument; project-level architecture must be more detailed because it steers development (specification) and informs administrators and maintainers (documentation)*
2. *Middleware that implements inversion of control and dependency injection and resides in mid tier of 3-tier architecture (or backend tier in 2-tier architecture).*
3. *Client/Server (lecture), Centralized Control Style (responsibility-driven design tutorial by R. Wirfs-Brock, page 28 onwards), Pipes-and-Filter (G. Starke, page 102 onwards)*

### Topic/Concept: Java Enterprise Edition (JEE)

1. *Web Container/Tier, EJB Container/Tier (note that the name "tier" is not used as defined in the lecture, the two JEE containers are not necessarily physically distributed; in practice, they are typically co-located)*
2. *Servlet API, JSF; EJBs, Tx; JPA, JDBC*
3. *Logical Layers, Client/Server*