**Chapter# 2**

***2.8***

***1. Name at least one technology that can implement universally invocable connectors***

Answer: Connectors represent interaction among components. They provide the glue for architectural designs. These technologies and framework are used for universal invocability. CORBA, web services, and messaging.

***2. What types of connectors are used in standard four-tiered web architecture?***

Ans: protocol-based connector are used in standard web architecture

***3. Name at least one technology that can implement self-descriptive connectors.***

Answer:Internet, HTTP, Web Applications use self-descriptive connectors

***4. Is class inheritance a type of software architecture connector?***

Answer: Yes, inheritance can be thought of as software architecture connector.

The information flow on a connector can be implemented using various information carriers. If the two incident elements are in the same process, say as two threads, they may use a shared variable to exchange information.

***5. What are the main approaches to agile software architecture design?***

Ans: Agile approach for developing software architectures that maximizes the reuse of architecture, design, and implementation investments. Given a project specification, an abstract high-level software architecture will first be proposed, and attributes will be identified for its elements and connectors.

***6. What are the major types of connectors used in a university's online registration system?***

Ans: Based on the implementation type, a connector may be classified as signature-based or protocol-based, whereas signature-based connectors can only be used to request one type of operation, a protocol-based connector can implement multiple operation types with a single binding signature. Connector active time refers to when an operation request or message is sent over a connector. Connectors may be classified into programmed connectors and event-driven connectors.

**Chapter# 3**

***3.7***

***1.* *List all interaction UML diagrams***

* Sequence diagram
* Collaboration diagram
* Timing diagram

**2.** ***List all structural UML diagrams.***

* [Class Diagram](https://creately.com/blog/diagrams/uml-diagram-types-examples/#ClassDiagram)
* [Component Diagram](https://creately.com/blog/diagrams/uml-diagram-types-examples/#ComponentDiagram)
* [Deployment Diagram](https://creately.com/blog/diagrams/uml-diagram-types-examples/#DeploymentDiagram)
* [Object Diagram](https://creately.com/blog/diagrams/uml-diagram-types-examples/#ObjectDiagram)
* [Package Diagram](https://creately.com/blog/diagrams/uml-diagram-types-examples/#PackageDiagram)
* [Profile Diagram](https://creately.com/blog/diagrams/uml-diagram-types-examples/#ProfileDiagram)
* [Composite Structure Diagram](https://creately.com/blog/diagrams/uml-diagram-types-examples/#CompStrDiagram)

**3. *List all early phase SDLC UML diagrams****.*

Ans. Use case diagram,

***4. List all late phase SDLC UML diagrams.***

**Ans.** Deployment Diagram,

**5. *Describe the relationship between sequence diagrams, communication diagrams, and interaction diagram***

**Ans.** Both diagrams give the same information, but the sequence diagram emphasizes time in its layout and the communication diagram emphasizes the objects that are communicating in its layout.

Time is implicit in sequence diagrams (it is inferred by vertical position), while it is given explicitly in communication diagrams (via numbers).

**While interaction diagrams** are used to represent how one or more objects in the system connect and communicate with each other. **Interaction diagrams** focus on the dynamic behavior of a system. An **interaction diagram** provides us the context of an **interaction** between one or more lifelines in the system

***6****.* ***Enumerate the problem domains suitable to state machine diagrams.***

**State machine diagrams** are usually applied to objects but can be applied to any element that has behavior to other entities such as: actors, **use** cases, methods, subsystems systems and etc. and they are typically **used** in conjunction with interaction **diagrams** (usually sequence **diagrams**)

***7. List problem domains suitable to time diagrams.***

Ans: **Timing diagrams** are UML interaction **diagrams used** to show interactions when a primary purpose of the **diagram** is to reason about **time**. **Timing diagrams** focus on conditions changing within and among lifelines along a linear **time** axis.

***8. In what case is the activity diagram a good choice?***

Ans: **Activity diagram** is used to model business processes and workflows. These **diagrams** are used in software modeling as well as business modeling. Most commonly **activity diagrams** are used to, Model the workflow in a graphical way, which is easily understandable.

***9. What is ADL?***

Ans: ADLs are formal languages for describing the architecture of a software system . Each ADL defines a notation with precise syntax and semantics in which architecture models can be expressed, and provides a corresponding toolkit for working with the language.

ADLs include general purpose languages like xADL and ACME, and [domain-specific languages](https://www.sciencedirect.com/topics/computer-science/domain-specific-language) (DSLs) like Koala , the Architecture Analysis and Design Language , and AUTOSAR . A survey of available ADLs can be found in . Many ADLs are academic research projects.

***10. What is the 4+1 view model?***

Ans: The 4+1 view model was originally introduced by Philippe Kruchten (Kruchten, 1995). The model provides four essential views: the logical view, the process view, the physical view, and the development view. The logical view describes, for example, objects and their interactions; the process view describes system activities, their concurrency and synchronization; the physical view describes the mapping of the software onto the hardware, the server, and the network configuration; and the development view describes the software's static structure within a given development environment.