**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?***

*Answer:* 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?***

*Answer:* The main approach to agile software architecture design is an abstract high-level software architecture will first be proposed, and attributes will be identified for its elements and connectors.

This abstract software architecture will generally be free of deployment considerations. The architecture will then go through multiple refinement processes to support particular deployment constraints.

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

*Answer:* Many types of connectors can be used in university online registration system. For example

• one-initiator connector can be used to send sms to university students and inform them about different event.

• signature-based connectors and protocol based connectors can be used to send information between different of university

• event-driven connectors will be used in GUI of the web application.

• Inside the university online software program connectors can be use to share data.

**Chapter# 3**

***3.7***

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

Ans. UML provides several modeling diagrams that can be grouped into two major categories: structural (static) and behavioral (dynamic).

**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 cases, Sequence diagrams, Class diagram, Communication.

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

**Ans.** Deployment and package diagram.

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

**Ans.** They all capture dynamic behavior of the system. In UML, interaction diagrams can take one of two forms: sequence diagrams or communication diagrams.

Interaction diagrams describe patterns of communication among a set of interacting objects. Sequence diagram is a special form of interaction diagram. Sequence diagrams represent the objects participating in the interaction horizontally and time vertically. Communication diagrams depict the same information as sequence diagrams.

Communication diagrams represent the sequence of messages by numbering the interactions.

--------------

***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.