

21 Lecture - CS504

Important Subjective

Q: What is the purpose of a Synchronous message in a Sequence Diagram? **A:** A Synchronous message indicates a method call where the sender waits for a response from the receiver before continuing.

Q: How is an Asynchronous message different from a Synchronous message in a Sequence Diagram? **A:** An Asynchronous message represents a method call where the sender continues its execution without waiting for a response from the receiver.

Q: What does a Return message indicate in a Sequence Diagram? **A:** A Return message shows the response from the called object to the calling object after a method call is completed.

Q: How are Synchronous messages represented in a Sequence Diagram? **A:** Synchronous messages are depicted with a solid line connecting the lifelines of the objects involved in the communication.

Q: Describe the appearance of an Asynchronous message in a Sequence Diagram. **A:** An Asynchronous message is represented with a dashed line connecting the lifelines of the objects involved in the communication.

Q: When is a Return message used in a Sequence Diagram? **A:** A Return message is used to show the response from the called object to the calling object after a method call is finished.

Q: What is the significance of the arrowhead in a Return message? **A:** The arrowhead in a Return message points back to the calling object, indicating the direction of the response.

Q: How are loops represented in a Sequence Diagram? **A:** Loops are depicted using a dashed line with a loop arrow, indicating repetitive message exchanges between objects.

Q: What information is conveyed by the timing of messages in a Sequence Diagram? **A:** The timing of messages in a Sequence Diagram shows the order and sequence of interactions between objects during runtime.

Q: How do message types in a Sequence Diagram help in understanding object interactions? **A:** Message types specify the communication behavior, such as synchronous or asynchronous, providing insights into how objects interact and exchange messages in the system.