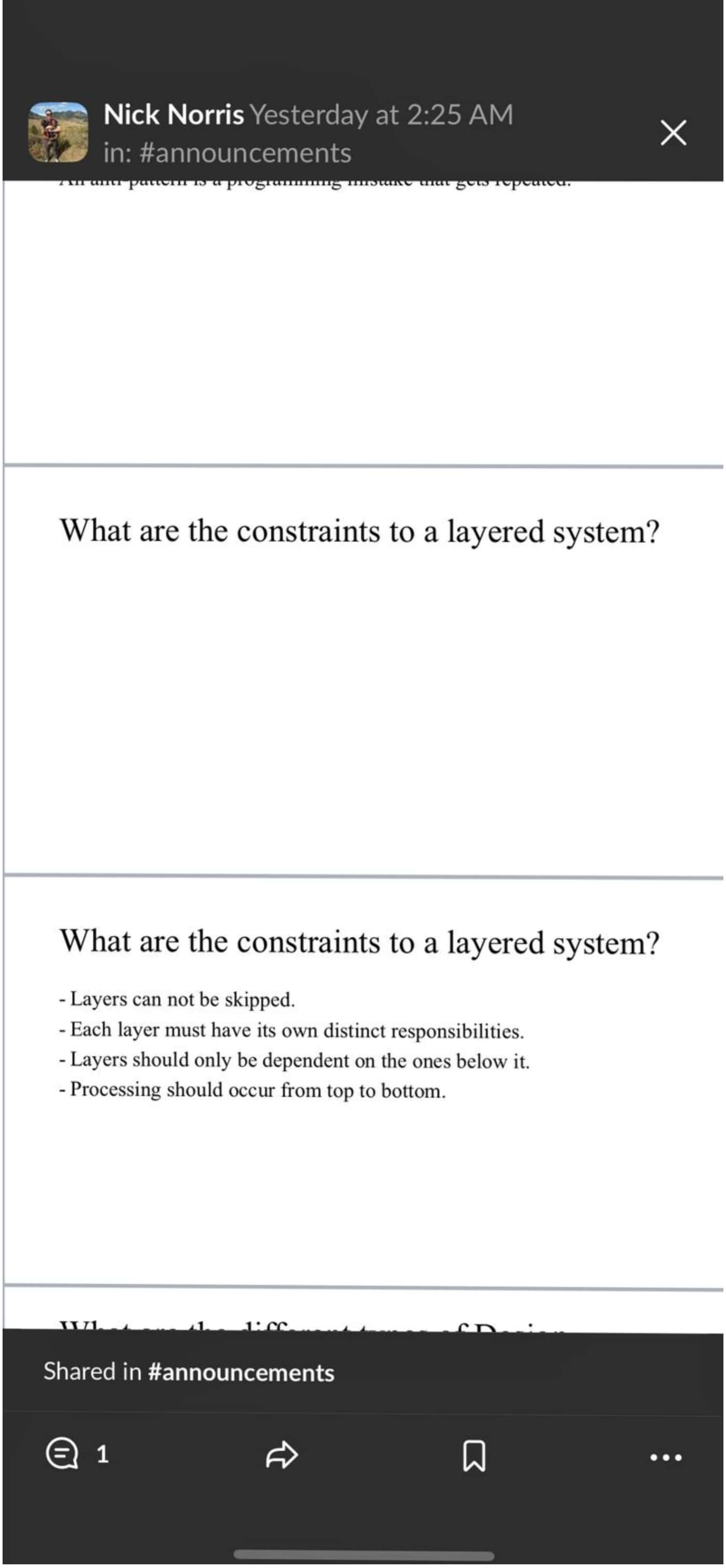

Nick Norris Yesterday at 2:25 AM
in: #announcements

An anti-pattern is a programming mistake that gets repeated.

What are the constraints to a layered system?

What are the constraints to a layered system?

- Layers can not be skipped.
- Each layer must have its own distinct responsibilities.
- Layers should only be dependent on the ones below it.
- Processing should occur from top to bottom.

What are the different types of Design

Shared in **#announcements**

1

What kind of diagram is a Use Case Diagram?

What kind of diagram is a Use Case Diagram?

Use Case Diagrams are **behavior** diagrams

They model the **functionality** of the system from the perspective of the user.

What kind of model is a Sequence Diagram?

What kind of model is a Sequence Diagram?

Sequence Diagrams are **Interaction** models

Sequence Diagrams represent the interaction that occurs via messaging between the system and its environment (outside systems, users, etc.).

Some Extra Detail

- Domain Class Models include entities that may be relevant.
  - Limited attributes.
  - Utilizes associations to model the relationship between entities.
  - No methods, types, or roles.
- System Class Models form the layers of the system.
  - Determine system boundary.
  - Derived from DCM with influence from Use Case Diagrams.
  - Breaks the system down into 3 layers: presentation (Boundary classes), application (Control classes), and data layer (Entity Classes).
  - Uses stereotypes to indicate layer membership.
- Implementation Class Model is the closest approximation to the real structure that we can manage.
  - Excludes associations and instead uses real data structures.
  - Includes all real class elements

True or False: Design problems have one solution

False

You will often find that there are multiple possible solutions that can fulfill the needs of the stakeholders. This is because there is a multitude of decisions that are made.

What are the advantages of a formal design process?

What are the advantages of a formal design process?

1. It makes **design decisions explicit**.
2. Allows for greater **documentation.**
3. Reduce the likelihood that **important issues are forgotten or overlooked.**

SER 315 2025 Spring B Final Review

Nicholas (Nick) Norris UGTA

Format

- Identify things for your handwritten notes. 1 page front and back is allowed
- Be prepared to draw a diagram
- Each topic has questions to challenge you and some additional overview at the end.

Week 1

Engineering Design Process and Class Diagrams

True or False: Design problems have one solution

What are Class Diagrams?

What are Class Diagrams?

They are **structural models** that have varying levels of abstraction. Each level aims to refine the architecture of the system in greater detail.

High level

- Design Process – a mostly iterative process whose product is a plan for a system based off a set of requirements.
- Class diagrams are structural models that have varying levels of abstraction. Each level hopes to refine the architecture.
- Domain class model is exploratory and formative. It aids in understanding the world surrounding the system.
- System class model refines your understanding of the architecture
- Implementation class model is a final summary of the structure.

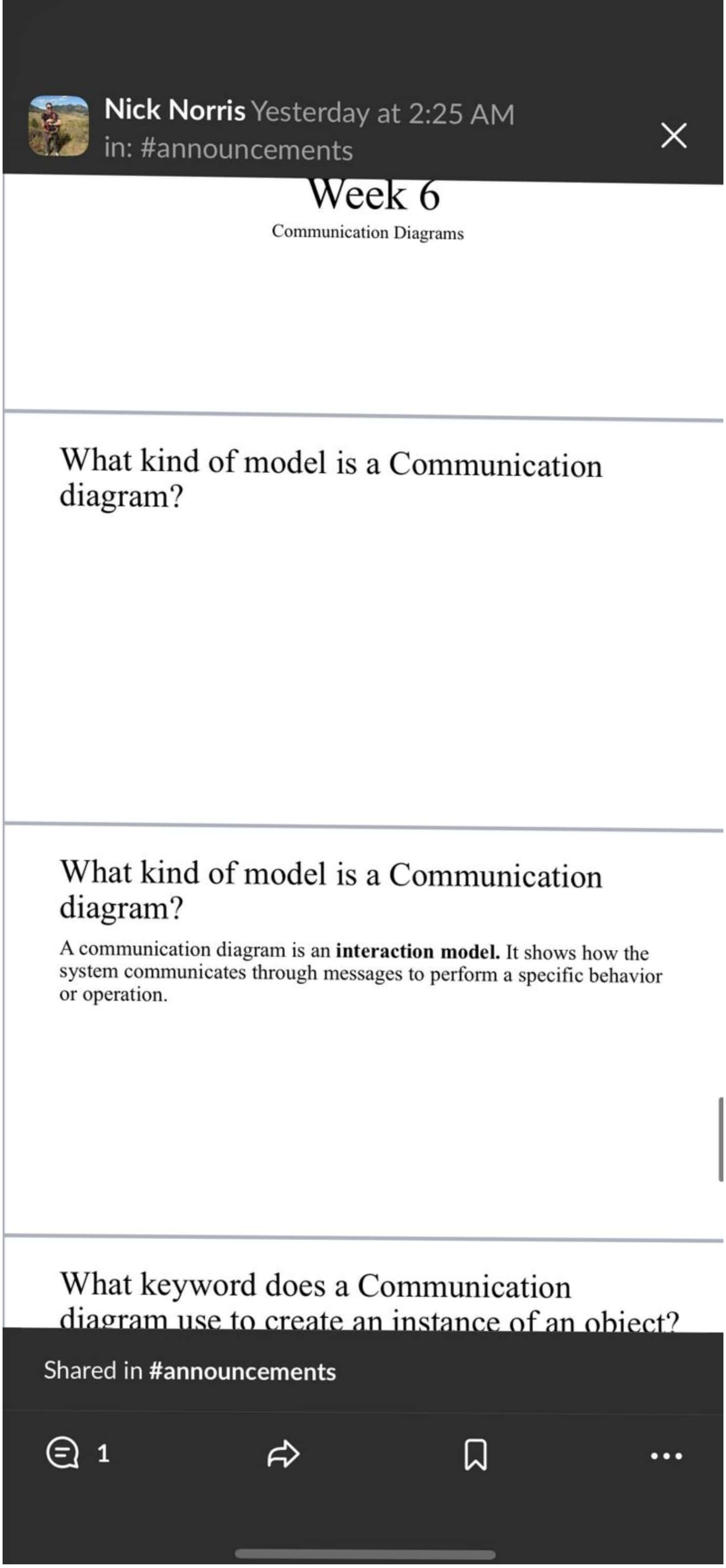
What are the different types of Design Patterns?

What are the different types of Design Patterns?

- Creational – the creation of objects.
- Structural – the composition of classes.
- Behavioral – how classes or objects interact and distribute responsibility.

Week 7: Design Patterns and Architectural Styles

- Design patterns are reusable solutions in a given context.
- Design patterns to focus on: composite, builder, strategy, factory, adapter, decorator, observer, singleton, and facade.
  - Identify their intended purpose.
  - Identify their general structure.
- Understand the following architectural styles: layered systems, blackboard, pipe and filter, and microkernel.
  - Identify advantages/disadvantages.
  - Identify constraints.



Nick Norris Yesterday at 2:25 AM
in: #announcements

Week 6

Communication Diagrams

What kind of model is a Communication diagram?

What kind of model is a Communication diagram?

A communication diagram is an **interaction model.** It shows how the system communicates through messages to perform a specific behavior or operation.

What keyword does a Communication diagram use to create an instance of an object?

Shared in **#announcements**

1

What does the **implicit** keyword mean?

Implicit implies that the existence or uniqueness of an object from the reads or changes is expected.

What does the post condition tell us?

What does the post condition tell us?

How the system state changes after the operation.

Week 6

Communication Diagrams

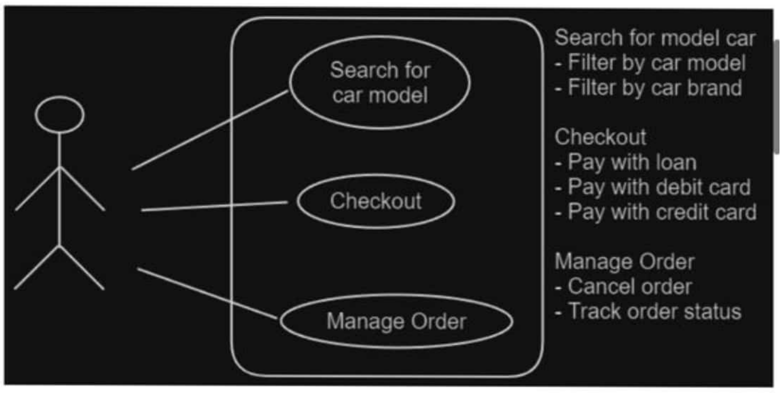
• Lifelines can be constructed and destroyed in abstraction levels where objects can exist.

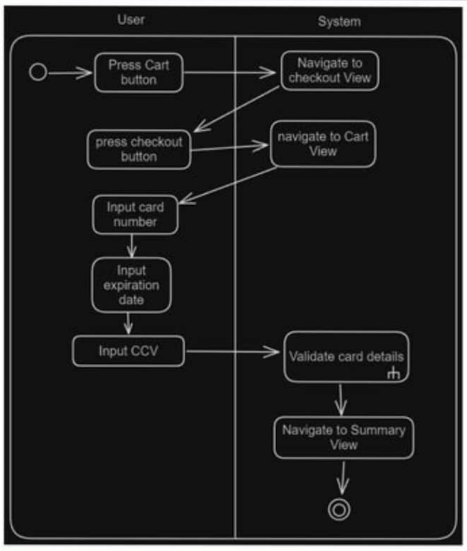
Week 3

Activity Diagrams, UI Design, and System Class Models

Create an Activity Diagram

Create An activity diagram from this use case diagram for the scenario of "Pay with debit card."





Week 3: Activity Diagrams

- Activity Diagrams are behavioral models that refine the activities that must be completed and their general flow.
  - Actions are the steps of the workflow.
  - Transitional arrows are used to indicate the next step.
  - Splits and merges are where decisions are made about the next step.
  - Forks and joins are used to indicate concurrency.
  - Swim lanes show the division of responsibilities between actors.
  - Objects can be shown as squares in UML, and are usually data that need

What kind of model is a Sequence Diagram?

Sequence Diagrams are **Interaction** models

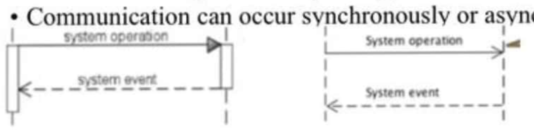
Sequence Diagrams represent the interaction that occurs via messaging between the system and its environment (outside systems, users, etc.).

Week 2: Use Case and Sequence Diagrams

- Use Case Diagrams are a behavioral model of the system's functional requirements based on requirements.
  - Main components consist of use cases, actors, scenarios, and associations.
  - Actors are people and systems outside of the system we are creating.
  - Use cases are the behaviors and functions that are available to the actors.
  - Scenarios are a lower classification of behavior that is used when utilizing a high level of abstraction (such as in your project).
  - Associations have multiple classes; inclusions (use cases that must also be utilized), extensions (specialized/optional use case), inheritance (generalization/specialization)
  - Includes a system boundary.

Week 2: Use Cases and Sequence Diagrams

- Sequence diagrams are an interaction model that models system operations and system events that occur between the actor and the system.
  - Each actor and system or object is given a lifeline.
  - Communication can occur synchronously or asynchronously



  - Alternatives are like if/else statements, while Options are like if statements
  - Repetition can occur with loops.
  - Lifelines can be constructed and destroyed in abstraction levels where objects can exist.

Week 3

Activity Diagrams, UI Design, and System Class Models

• Excludes associations and instead uses real data structures.
• Includes all real class elements

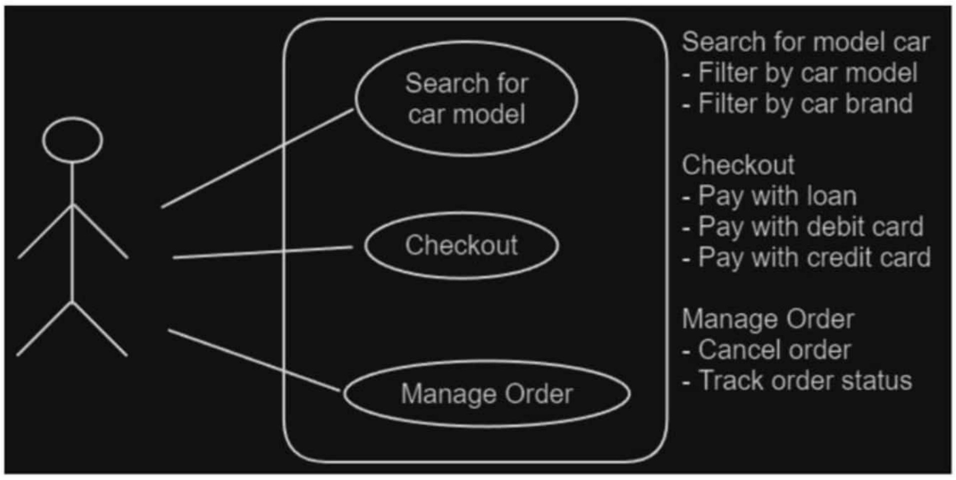
Week 2

Use Case Models and Sequence Diagrams

Write a Use Case diagram for the following requirements

You are asked to create a system for Acme Model Car Manufacturing. You have the following list of requirements:

1. Users should be able to filter their search with a car model.
2. Users should be able to filter their search with the brand of car.
3. Users should be able to purchase their cars with buy-now pay-later loans, debit cards, or credit cards.
4. Users should be able to cancel orders prior to them being shipped.
5. Users should be able to track the status of their online order.



Search for model car
- Filter by car model
- Filter by car brand

Checkout
- Pay with loan
- Pay with debit card
- Pay with credit card

Manage Order
- Cancel order
- Track order status

What kind of diagram is a Use Case Diagram?

Week 4: Formalization and Z-Notation

- Design by contract states that if the preconditions are met then this component should fulfill the post conditions.
  - Sets describe the membership of elements to a grouping.
  - Predicates describe true/false statements given a few conditions.
- Formal methods are vital in safety critical systems.
  - Identifies uncertainty about the state of the system at a point.
  - Requires precision in the logic.
- Completed early in the design process.

Week 4: Formalization and Z-Notation

- Z-notation is a formalization language used in Operational patterns.
  - Types give context to variables; therefore, variables must be typed.
  - A cartesian product is the set of all possible pairs of two sets.
  - Predicates establish true/false statements.
  - Sets require members to satisfy a predicate.

Week 5

Operational Patterns

Which part of the documentation might we pull these components of an OP from?

- Description
- Input
- Reads/Changes
- Sends
- Pre conditions
- Post conditions

Explain the meaning for the following Z Notation operators

•

- # is the cardinality operato
  number of elements in a s
- ∈ is the “element of” opera
- U is the union operator.

What is the purpose of formal specification in Software Engineering?

What is the purpose of formal specification in Software Engineering?

Formal specification removes all ambiguity that may exist in natural language descriptions. Formal specification allows for the developer to verify correctness, consistency, and completeness. They are utilized in safety critical systems.

Week 4: Formalization and Z-Notation

- Design by contract states that if the preconditions are met then this component should fulfill the post conditions.
  - Sets describe the membership of elements to a grouping.
  - Predicates describe true/false statements given a few conditions.
- Formal methods are vital in safety critical systems.
  - Identifies uncertainty about the state of the system at a point.
  - Requires precision in the logic.
- Completed early in the design process.

Week 3: System Class Models

- System class model refines your understanding of the architecture
  - Forms the layers of the system.
    - Determine system boundary.
    - Derived from DCM with influence from Use Case Diagrams.
    - Breaks the system down into 3 layers: presentation (Boundary classes), application (Control classes), and data layer (Entity Classes).
    - Uses stereotypes to indicate layer membership.

Week 4

Formalization and Z-Notation

Explain the meaning for the following Z Notation operators

-
  - #
  - ∈
  - ∪
  - ∩

Explain the meaning for the following Z Notation operators

-
  - # is the cardinality operato number of elements in a s
  - ∈ is the "element of" opera

Week 3: Activity Diagrams

- Activity Diagrams are behavioral models that refine the activities that must be completed and their general flow.
  - Actions are the steps of the workflow.
  - Transitional arrows are used to indicate the next step.
  - Splits and merges are where decisions are made about the next step.
  - Forks and joins are used to indicate concurrency.
  - Swim lanes show the division of responsibilities between actors.
  - Objects can be shown as squares in UML, and are usually data that need shared between actors.
  - Heavily influenced by use case diagrams, 1 AD per use case, 1 AD per Use case actor interaction.

What are the 3 layers of the System Class Model?

What are the 3 layers of the System Class Model?

- **Data Layers or Entity layer/class** – holds the system's data
- **Application Layer or Control layer/class-** encapsulates the system's business logic
- **Presentation Layer or Boundary layer/class** – represents the outward facing interface to the user.

Week 3: System Class Models

- System class model refines your understanding of the architecture
  - Forms the layers of the system.
    - Determine system boundary.
    - Derived from DCM with influence from Use Case Diagrams.
    - Breaks the system down into 3 layers: presentation (Boundary classes), application (Control classes), and data layer (Entity Classes).
    - Uses stereotypes to indicate layer membership.

Week 7

Design Patterns

What Design pattern would you use if you had a resource, and you wanted to ensure only one instance of it existed?

What Design pattern would you use if you had a resource, and you wanted to ensure only one instance of it existed?

The Singleton pattern ensures that only one instance of a class exists. It does this by making the constructor private and using a static method to create the object on the first call with subsequent calls only returning the created instance.

What is an anti-pattern?

What keyword does a Communication diagram use to create an instance of an object?

What keyword does a Communication diagram use to create an instance of an object?

O = create(parameters)

Week 6: Communication Diagrams

- Communication diagrams are interaction models that describe the communication (primarily function calls) of objects within the context of a feature/scenario.
- Focus is on objects not generic classes, include object names with class type.
- Utilize guards to account for conditions that must be checked.
- Communication diagrams are based off the system class model (for class names, and structure), sequence diagrams (for the operation, and messages), and operational pattern (for guards, and reads/changes).

Week 5

Operational Patterns

Which part of the documentation might we pull these components of an OP from?

- Description
- Input
- Reads/Changes
- Sends
- Pre conditions
- Post conditions

Which part of the documentation might we pull these components of an OP from?

- Description – requirements
- Input – Sequence Diagrams and System Class Model
- Reads/Changes – System Class Model
- Sends – Sequence Diagrams
- Pre conditions – assumptions gathered from the requirements
- Post conditions – assumptions about the state after the operation, which can be pulled from System Class Model and Sequence Diagrams

What does the **implicit** keyword mean?