- Patterns – what do we need to know for Half-Exam 1?
    - Code: Be able to write short snippets of code for a pattern, given the context and pattern you are working on. You do NOT need to refactor code to use the pattern.
    - UML: Be able to interpret UML diagrams and the significance of the pattern structures
    - Advantages and disadvantages: Be able to discuss the advantages and disadvantages of each pattern, given examples
- Domain-level diagrams
    - Domain-level: objects/classes related to the problem that won't change with changing implementations
        - RIBS – what each letter stands for
    - Solution-level: objects/classes that are need for the solution
- Labs
    - Lab 1:
        - Write short code snippets similar to the code you wrote in the lab
        - Perform UML design similar to design work you were asked to do in the lab
- UML
    - Relationship Arrows
        - IS-A
        - HAS-A
        - Make sure to draw them in the right direction
    - Multiplicity
        - 1 – Simply omit 1; it is implied
        - 0..* – Use just * instead
        - 0..1
        - 1..*
        - 0..2 – Use just * instead usually
        - 1..2 – Use just 1..* instead usually
    - Underline: static
    - Italics: abstract
    - {readOnly}: Final
    - Access levels: +/-/# public/private/protected