## Design

* Architecture styles (pipe and filter, event-driven, client server, model-view-controller)
* Describe the symbols used in the following diagrams and identify their purposes: network diagram, component diagram, dataflow diagram
* Identify the three types of connectivity in a relational database and give an example of each.
* Cohesion and coupling
* Given a scenario, identify some classes and create a design including UML class diagrams
* Design patterns - what are they and why do we use them?
* Describe Mandel's 3 golden rules of interface design
* Describe **a** design pattern – observer, strategy, factory or singleton – and give an example.

## Implementation

* Refactoring - what is it and why do it? What are some things to look for?
* Why is good layout important? Describe some things that you can do to make your code more readable?
* What do we mean by self-documenting code?
* Know what code tuning is and some common sources of inefficiency.

## Testing

* Verification vs. validation
* What is unit testing and who typically does unit testing?
* Black box testing vs. white box testing
* What is integration testing? Describe one or more integration testing strategies.
* What is regression testing?
* Inspections vs. testing

## Evolution

* Why is software change inevitable?
* What is software maintenance and why is it important?

## Dependability and Security

* Describe the principle dependability properties.
* What is the difference between reliability and availability?
* What is security engineering?
* Give some examples of security requirements.
* What are some guidelines for designing a secure system?
* What is system survivability and what are some strategies for system survivability?