Lecture 12, Part 2:
Final Exam
Jennifer Campbell
CSC340 - Winter 2007

Course Outline
- Lecture 1
  - What is Requirements Engineering?
  - What are Requirements?
- Lecture 2
  - Science, Engineering and Systems
  - Requirements Specifications
- Lecture 3
  - Formal Inspections
  - Feasibility Studies
- Lecture 4
  - Stakeholders Goals
  - Elicitation Techniques
- Lecture 5
  - Risk
  - Intro to Requirements Modelling
- Lecture 6
  - Modelling Enterprises
  - Modelling Objects
- Lecture 7
  - Modelling Relationships
  - Modelling State
- Lecture 8
  - Modelling Events
- Lecture 9
  - Modelling Interactions
  - Non-functional Requirements
- Lecture 10
  - Verification and Validation
  - Prioritizing Requirements
- Lecture 11
  - Software Evolution
  - Moving into Design
- Lecture 12
  - Software Architectures

Modelling: UML
- Activity diagrams
  - capture business processes involving concurrency and synchronization
  - good for analyzing dependencies between tasks
- Class Diagrams
  - capture the structure of the information used by the system
  - good for analysing the relationships between data items used by the system
  - good for helping you identify a modular structure for the system
- Statecharts
  - capture all possible responses of an object to all uses cases in which it is involved
  - good for modeling the dynamic behavior of a class of objects
  - good for analyzing event ordering, reachability, deadlock, etc.
- Use Cases
  - capture the view of the system from the view of its users
  - good starting point for specification of functionality
  - good visual overview of the main functional requirements
- Sequence Diagrams
  - capture an individual scenario (one path through a use case)
  - good for modelling dialog structure for a user interface or a business process
  - good for identifying which objects (classes) participate in each use case
  - helps you check that you identified all the necessary classes and operations

Exam Format
- Length: 2 hours
- Closed book
- Format is similar to previous exams
- Types of questions:
  - Short answer questions
    - Written questions, calculations, etc
  - Modelling questions
    - Create a model according to a description
    - Three modelling notations are definitely NOT on the final:
      - Fault trees, robustness diagrams, and package diagrams
Modelling: Non-UML Notations

- **Goal Models**
  - Capture strategic goals of stakeholders
  - Good for exploring ‘how’ and ‘why’ questions with stakeholders
  - Good for analysing trade-offs, especially over design choices

- **Entity-Relationship Models**
  - Capture the relational structure of information to be stored
  - Good for understanding constraints and assumptions about the subject domain
  - Good basis for database design

- **Mode Class Tables, Event Tables and Condition Tables (SCR)**
  - Capture the dynamic behaviour of a real-time reactive system
  - Good for representing functional mapping of inputs to outputs
  - Good for making behavioural models precise, for automated reasoning