The University of Melbourne
SWEN90010: High Integrity Systems Engineering
Learning Outcomes

By the end of this subject, a student should be able to do the following (categorised by chapters in the course notes):

Chapter 1 — An Introduction to HISE

- Define the term "high-integrity system"
- Define the different classes of high-integrity system

Chapter 2 — Ada

- Describe the features of Ada that make it suitable for high-integrity software
- Read and modify basic Ada programs

Chapter 3 — Safety engineering

- Explain the role of safety engineering in the system engineering lifecycle.
- Discuss the role of accidents and incidents in the safety analysis process
- Perform a preliminary hazard analsyis using the HAZOP method
- Apply the fault-tree analysis method to a system for a given fault

Chapter 4 — Model-based specification

- Explain the advantages and disadvantages of formal model-based specification in software engineering
- Apply basic logic, set/relational theory concepts to software-based problems
- Model a domain using the Alloy language
- Define and check assertions using the Alloy language and tool
- Model and reason about (properties of) execution sequences in Alloy

Chapter 5 — Fault-tolerant design

- Explain the concept of fault tolerance in systems engineering
- Compare hardware and software fault tolerance
- Design, analyse, and critique a fault-tolerant hardware design
- Design, analyse, and critique a fault-tolerant software design
- Implement algorithms for majority voting, median voting, and k-plurality voting

- Compare and contrast the different voting algorithms, and evaluate their use in specific systems
- Apply the concepts of duplication, parity coding and checksums to small information redundancy problems, and explain the situations in which cryptographic mechanisms are necessary instead.
- Analyse and explain the lower bounds for Byzantine Agreement in the authenticated and unauthenticated models, and the protocols that achieve those bounds.

Chapter 6 — SPARK

- Describe the features of SPARK that make it suitable for high-integrity software

Chapter 8 — Reasoning about program correctness

- Explain the advantages and disadvantages of program proof compared to other program verification techniques.
- Explain the meaning of Hoare logic statements: $\{P\}\ S\ \{Q\}$
- Devise appropriate loop invariants for reasoning about loops using Hoare logic
- By hand, prove the correctness or otherwise of small programs using Hoare logic

Chapter X — Security and cryptography

- Informally describe the main security properties of digital signatures, message authentication codes, and cryptographic hash functions.
- Understand and analyse the logical trust structure of digital certificates or the properties of public key encryption, and synthesise a model of it in Alloy.
- Recall and explain some specific recent examples of security problems caused by the use of weak or poorly implemented cryptography.
- Explain at a high level how the Bitcoin protocol works and how certain properties are achieved.