**Dependable and Distributed Systems**
**Examination Guidance**

Whilst everything taught in the module is examinable, the lists below are intended to help focus your revision on the most essential elements of each topic covered. You should not rely exclusively on this document to plan and structure your revision for the module but it can be helpful to make sure that your revision spans the breath of the module and that you understand exam expectations.

## What can be expected in the examination?

In terms of the focus of questions, you will be required to know the breadth and depth of the content covered in lectures. On top of that, you will be be expected to apply some problem solving in order to answer some question, particularly the more challenging part of the examination that require you to combine knowledge and problem solving.

Typical command words for this type of exam include but are not limited to: Analyse, Argue, Calculate, Compare, Define, Demonstrate, Derive, Develop, Discuss, Distinguish, Enumerate, Justify, Explain, Prove, Quantify, Show, Solve and State. Many of these require you to demonstrate understanding and the ability to apply that understanding, potentially in the context of a previously unseen problem. Try not to let that scare you too much. If you understand the material and think carefully about what you write, you can gain a good amount of credit for taking steps in the right direction and you will often find that taking a few steps in the right direction will lead you to the answer.

## Topic 1 - Introduction

Motivating Dependable and Distributed Systems

## Topic 2 - Dependability Concepts

Dependability Impairments

- Explain Faults
- Explain Errors
- Explain Failures
- Explain The Fundamental Chain

Dependability Means

- Explain Fault Prevention
- Explain Fault Tolerance
- Explain Fault Removal
- Explain Fault Forecasting

Dependability Attributes

- Explain and Calculating Availability
- Explain Confidentiality
- Explain Integrity

Explain and Calculating Maintainability

Explain and Calculating Reliability

Explain Safety

## Topic 3 - Software Fault Tolerance

### Detectors and Correctors

Explain the Structure of Detectors

Explain the Structure of Correctors

Relate Detectors and Correctors to Fault Tolerance Classes

### Fault Tolerance Classes

Relate Fault Tolerance Classes to Detectors and Correctors

Identify Examples of Detectors and Correctors Achieving Fault Tolerance Classes

### Phases of Fault Tolerance

Explain the Phases of Fault Tolerance

Explain Error Recovery Classes

### Checkpointing

Explain Checkpointing Concepts

Develop and Justify Policies for Distributed Checkpointing

### Recovery Blocks

Explain the Motivations for Recovery Blocks

Explain and Justify Recovery Block Design

Develop the General Form of Recovery Blocks

### N-Version Programming (NVP)

Explaining the Motivations for NVP

Analysis and Justify the Application of NVP With and Without Majority Voting

Explain the Theoretical and Practical Implication of the NVP Axiom

Identify Challenges in Applying NVP With and Without Majority Voting

### Fault Injection Analysis

Explain the Motivation and Application of Fault Injection Analysis

Assess Test Case Fault Injection Coverage

### Parameter Estimation

Explain the Motivation and Application of Parameter Estimation

## Topic 4 - Dependability Analysis and Evaluation

Hazards, Risks and Safety

- Define and Explain Hazards
- Define and Explain Risk
- Define and Explain Safety Cases
- Define and Explain Common Forms of Hazard Analysis

Exponential Failure Law

- Derive the Exponential Failure Law

MTTF, MTTR and MTBF

- Calculate and Deriving MTTF
- Calculate and Deriving MTTR

Combinatorial Modelling

- Calculate Reliability for Series Systems
- Calculate Reliability for Parallel Systems
- Calculate Reliability for M-of-N Systems

Cut Set and Tie Sets

- Define Cuts Set and Tie Set Concepts
- Enumerate Cut Sets and Tie Sets

Markov Models

- Derive Continuous Time Markov Chains
- Solve Continuous Time Markov Chains
- Explain and Show How Coverage Concepts Result in Markov Model Adaptation

Triple Modular Redundancy (TMR)

- TMR Reliability Calculation
- TMR Combinatorial Modelling
- TMR Stochastic Modelling

## Topic 5 - Leader Election

Synchronous Network Model

- State, Explain and Justify the Characteristics of the Synchronous Network Model
- Define Synchronous Algorithms Using the Synchronous Network Model

Leader Election Token Ring

- Define the Leader Election Problem

Develop Algorithms to Solve Leader Election

Argue and Prove the Correctness of Algorithms Solving Leader Election

Analyse the Time Complexity of Algorithms Solving Leader Election

Analyse the Communication Complexity of Algorithms Solving Leader Election

Analyse, Identify and Explain Optimisations for Algorithms Solving Leader Election

Leader Election General Network

Define the Leader Election Problem

Develop Algorithms to Solve Leader Election

Argue and Prove the Correctness of Algorithms Solving Leader Election

Analyse the Time Complexity of Algorithms Solving Leader Election

Analyse the Communication Complexity of Algorithms Solving Leader Election

Analyse, Identify and Explain Optimisations for Algorithms Solving Leader Election

## Topic 6 - Consensus

Consensus Problems

Define the Distributed Consensus Problem

Define and Explain the Properties Required of Consensus Algorithms

Distinguish Between Link, Stop / Crash and Byzantine Failure

Consensus with Link Failure

Define and Explain the Concept of Execution

Define and Explain the Concept of Indistinguishable

Argue and Prove the Impossibility of Solving Consensus with Link Failure

Consensus with Stop Failure

Develop Algorithms to Solve Consensus

Argue and Prove the Correctness of Algorithms Solving Consensus

Analyse the Time Complexity of Algorithms Solving Consensus

Analyse the Communication Complexity of Algorithms Solving Consensus

Analyse, Identify and Explain Optimisations for Algorithms Solving Consensus

Analyse and Explain the Implication of Decision Rules for Consensus Algorithms

## Topic 7 - Byzantine Generals Problem

The Byzantine Generals Problem (BGP)

Define BGP

Define and Explain the Properties Required of BGP Algorithms

Explain the Conditions Under Which BGP is Possible to Solve

Argue and Prove the Impossibility of Solving the BGP Under Stated Conditions

Develop Unsigned and Signed Message Algorithms to Solve BGP

Analyse the Time Complexity of Algorithms Solving BGP

Analyse the Communication Complexity of Algorithms Solving BGP

Practical Byzantine Fault Tolerance (PBFT)

Explain the Motivation for PBFT

Develop Algorithms to Implement PBFT

Explain the Advantages and Disadvantages of PBFT

Consensus and Network Models

Explain the Implications of the One-Crash Impossibility for Asynchronous Networks

Multi-Paxos and Variants

Explain and Justify the Assumptions of Paxos and Multi-Paxos

Develop and Explain the Paxos and Multi-Paxos Algorithms

Identify the Advantages and Disadvantages of Paxos and Multi-Paxos

Explain How Paxos and Multi-Paxos is Adapted and Applied in Practical Deployments

F-Tolerance of Consensus for Synchronous, Partially Synchronous and Asynchronous Systems

State and Explain the F-Tolerate of Consensus Protocols Under Varying System Models

## Topic 8 - Clock Synchronisation, Logical Clocks and Vector Clocks

Synchronisation, Drift and Skew

Distinguish Between Physical and Logical Clocks

Explain and Justify the Use of Physical Clocks

Explain and Identify Challenges in the Application of Physical Clocks

Define and Explain the Non-Faulty Clocks

Define and Explain Clock Drift

Define and Explain Clock Skew

Define and Explain Clock Skew

Reliable Time Sources

Define and Explain the Requirements for the Implementation of Reliable Time Sources

Explain the Implementation of the the Berkeley Algorithm for RTS Distribution

Motivate and Explain the Use of Convergence Functions in Satisfying RTS Requirements

Outline Generic Approaches to the Implantation of Clock Synchronisation Protocols

Logical Clocks

Motivate the Development and Application of Logical Clocks in Distributed Systems

Explain and Justify the Use of Logical Clocks

Define the Properties of the ‘Happens Before’ Relation in Logical Clocks

Explain the Implications and Limited of the 'Happens Before' Relation in Logical Clocks

Vector Clocks

Motivate and Explain the Use of Vector Clocks in Distributed Systems

Define the Correctness Conditions for the Implementation of Vector Clocks

Explain and Demonstrate how Vector Clocks Address on the Limitations of Logical Clocks

## Topic 9 - Reliable Links and Failure Detectors

Fair Loss Links

Define and Explain the Properties of Fair Loss Links

Stubborn Links

Define and Explain the Properties of Stubborn Links

Develop and Algorithm for the Implementation of Stubborn Links based on Fair Loss Links

Reliable Links

Define and Explain the Properties of Reliable Links

Develop and Algorithm for the Implementation of Stubborn Links based on Reliable Links

Links Failure Detectors

Define and Explain the Role of Failure Detectors in Asynchronous Systems

Define the Properties and Implications of Applying a Perfect Failure Detector

Develop and Explain Algorithms for the Implementation of a Perfect Failure Detector

Distinguish Between the Properties of Failure Detectors and Perfect Failure Detectors

## Topic 10 - Reliable Broadcast Algorithms

Broadcast Algorithm Abstractions

Characterise and Explain the Purpose of Broadcast Abstractions

Best Efforts Broadcast

Define and Explain the Properties of Best Efforts Broadcast Algorithms

Define the Interface for the Implementation of Best Efforts Broadcast Algorithms

Develop Best Efforts Broadcast Algorithms

Analyse the Communication Complexity of Best Efforts Broadcast Algorithms

Reliable Broadcast

Define and Explain the Properties of Reliable Broadcast Algorithms

Define the Interface for the Implementation of Reliable Broadcast Algorithms

Develop Reliable Broadcast Algorithms

Analyse the Communication Complexity of Reliable Broadcast Algorithms

Uniform Reliable Broadcast

- Define and Explain the Properties of Uniform Reliable Broadcast Algorithms
- Define the Interface for the Implementation of Uniform Reliable Broadcast Algorithms
- Develop Uniform Reliable Broadcast Algorithms
- Analyse the Communication Complexity of Uniform Reliable Broadcast Algorithms

FIFO Broadcast

- Define and Explain the Properties of FIFO Broadcast Algorithms
- Define the Interface for the Implementation of FIFO Broadcast Algorithms
- Develop FIFO Broadcast Algorithms
- Analyse the Communication Complexity of FIFO Broadcast Algorithms

Causal Broadcast

- Define and Explain the Properties of Causal Broadcast Algorithms
- Define the Interface for the Implementation of Causal Broadcast Algorithms
- Develop FIFO Broadcast Algorithms
- Analyse the Communication Complexity of Causal Broadcast Algorithms