Subject	Year 12 Computer Science	
Title/Topic	**Format**	**Length**
Paper 1 – Computer Systems	Written Paper, exam style questions	90 mins
Paper 2 – Problem Solving and Programming	Written paper with coding-like questions	90 mins

In this Lent assessment I will be asked to show I can…

Paper 1 – Computer Systems

1. Computer Architecture & CPU

- **Von Neumann Architecture**
  - Features of Von Neumann architecture
  - Differences between Von Neumann and modern CPU architectures (e.g., Harvard)
- **Little Man Computer (LMC)**
  - Understanding LMC instructions
  - Interpreting and writing simple LMC programs
- **Contemporary CPU Enhancements**
  - Features that improve performance (e.g., pipelining, cache memory, multi-core processors)
- **CISC vs. RISC Processors**
  - Differences between CISC and RISC
  - Advantages and disadvantages of each

2. Assembly Language & High-Level Languages

- **Differences between Assembly and High-Level Languages**
  - Advantages and disadvantages of assembly language
  - Situations where assembly is used (e.g., embedded systems)
- **Writing High-Level Code vs. Assembly**
  - Translating assembly into high-level language
  - Understanding different programming paradigms

3. CPU Performance & Parallel Processing

- **CPU Registers and Buses**
  - How data moves within a CPU
  - The role of registers (e.g., MAR, MDR, ACC, PC)
  - The function of the address, data, and control buses
- **Parallel Processing**
  - How parallel processing increases performance
  - Uses of parallel processing in computing tasks

4. Operating Systems

- **Memory Management**
  - How an OS manages physical memory (paging, segmentation)
  - Virtual memory and its role
  - Benefits of memory management for users
- **Device Drivers**
  - Definition and purpose
  - Examples of device drivers in a home computer
- **Utility Software**
  - Examples (e.g., antivirus, disk cleanup, backup software)
  - How utility software enhances system performance

5. Scheduling Algorithms

- **Types of Scheduling Algorithms**
  - First Come First Served (FCFS)
  - Round-Robin
  - Other scheduling algorithms (e.g., Shortest Job Next, Multi-level Queue)
- **Purpose of Scheduling in OS**
  - How scheduling optimizes CPU usage
  - Why different algorithms are used for different tasks

6. Encryption & Hashing

- **Encryption**
  - Different types (symmetric vs. asymmetric)
  - How encryption secures communication and stored data
- **Hashing**
  - How hashing is used for data integrity and security
  - Differences between encryption and hashing

7. Databases & SQL

- **Relational Databases**
  - Primary keys and their purpose
  - Foreign keys and how they link tables
- **SQL Queries**
  - Writing basic SQL queries (SELECT, WHERE)
  - Filtering data based on conditions (e.g., returning users who see adverts)
- **Data Types in Databases**
  - Different data types (integer, Boolean, float, etc.)
  - Choosing appropriate data types for fields

8. Networking & Protocols

- **Local Area Networks (LAN)**
  - Advantages of using a LAN
  - Common LAN protocols
- **Network Protocols**
  - Definition of a protocol
  - Examples of protocols used in LANs (e.g., TCP/IP, Ethernet)
- **Protocol Layering**
  - Why protocol layering is used
  - Benefits of layered network models (e.g., OSI model)

Paper 2 – Algorithms and Programming

1. Abstraction & Modelling
- Definition of **abstraction**
- Reasons for using abstraction in system modelling
- Differences between a **virtual model** and real-world systems

2. Programming Concepts
- **Global vs. Local Variables**
  - When and why to use global variables
- **Integrated Development Environments (IDEs)**
  - Features of an IDE and how they assist programmers
- **Algorithms & Pseudocode**
  - Writing structured algorithms for:
    - Validating user input
    - Storing and manipulating data in arrays
    - Outputting calculated values (e.g., average scores)
- **Recursion & Iteration**
  - Identifying recursive functions
  - Difference between **branching** and **iteration**
  - Understanding parameter passing (**by value vs. by reference**)
  - MOD arithmetic operation

3. Object-Oriented Programming (OOP)
- **Classes & Objects**
  - Defining classes with attributes and constructors
  - Writing **getters and setters**
- **Inheritance**
  - Creating subclasses with additional attributes
  - Overriding and extending class functionality
- **Data Structures**
  - Using **arrays to store objects**
  - Searching an array of objects using a function

4. File Handling & Data Processing
- **Reading & Writing to Files**
  - Opening and reading from text files
  - Using **iteration** to process file data
- **Processing Data**
  - Calculating totals, averages, and number of entries

5. Software Development Practices
- **Reusable Components**
  - Benefits of reusable code modules
- **Programming Constructs**
  - Examples of selection, iteration, and functions in a program

6. Validation & User Input Handling
- **Validating Input Length**
  - Comparing while-loops and if-statements for validation
- **Designing User Account Systems**
  - Functions for:
    - User **registration**
    - User **login**
    - Adding items to a shopping basket

7. Parameter Passing
- **By Reference vs. By Value**
  - Differences and when to use each

What should I do to revise and prepare for this assessment?

- Complete practice questions in the lesson booklets
- Re-watch craigndave videos for theory topics
- Complete relevant programming challenges utilising the skills listed above

What useful websites/resources could I use to help me prepare?

Year 12 - De Lisle Lent Term Assessments 2025-2026

All resources and files are available on the Student OneDrive