5.2 Module 2513: Structured Practical ICT Tasks

C3.3; N3.2, N3.3; IT3.2	**WO3.2; LP3.1; LP3.2, LP3.3; PS3.1, PS3.2, PS3.3**

This module is designed to develop practical aspects of the subject in a way that is not possible when leading to a formal written examination. A structured approach is possible which is suitable for Advanced Subsidiary GCE candidates.

The following skills are developed:

- Design;
- Software Development;
- Testing;
- Implementation.

This module covers basic knowledge and understanding, as well as skills. No one task tests all three skills. Each task addresses two of the three skills.

- One task may involve the design of a system.
- Another may involve the derivation and production of a testing strategy.
- Some tasks involve the use of hardware for implementation.

Recommended Prior Knowledge

Module 2512 and Section 5.3.1 of Module 2514 should be covered before attempting this module. However, it is envisaged that work on the structured practical tasks will start almost in parallel to work on Module 2514, thus as soon as the relevant material in Module 2514 has been covered it is advisable for candidates to begin the appropriate exercise in Module 2513.

5.2.1 Design

Candidates should be able to specify and document a design, using appropriate algorithms/models. The design specification may include the method of solving a problem, for example:

- General hardware and software requirements;
- Interface design including hardware requirements;
- Data structures/model;
- Special routines/requirements;
- Specify any security measures required.

Learning outcomes

Candidates should be able to:

- specify in general terms the required hardware and software for a given problem (related in particular to sections 5.1.2, 5.1.5);
- specify and document data capture forms and/or screen layouts/displays, report layouts and/or other forms of output (for example, audio output), together with any special input/output devices required for designing a good user interface (related to sections 5.1.2, 5.1.3);
- design and document the data structures necessary to model a given problem (related in particular to section 5.1.4);
- specify and document any required tailoring/routines or any special requirements (related in particular to section 5.1.1 and relevant parts of 5.3);
- specify any security measures required (related to sections 5.1.2, 5.1.4, 5.1.5).

5.2.2 Software Development

Content

- Interpreting a design solution
- Developing a tailored software solution

Learning outcomes

Candidates should be able to:

- interpret a design solution specifying any variables and data structures using an appropriate software tool (related to sections 5.1.1, 5.1.3, 5.1.4);
- interpret a design solution for the interface preparing any interface requirements as a user prototype (related to section 5.1.3);
- develop routines to tailor applications software to meet the design, security and user interface requirements (related to sections 5.1.3, 5.1.4 and relevant parts of 5.3).

5.2.3 Testing

Content

- Test strategy
- Test data

Learning outcomes

Candidates should be able to:

- identify, develop and document a test strategy for a given problem, testing in particular navigational paths and interactivity, as well as functionality (related in particular to section 5.1.1);
- select suitable test data and test actions/responses for a given problem (related in particular to section 5.1.1).

5.2.4 Implementation

Content

- Software testing
- User documentation

Learning outcomes

Candidates should be able to:

- test a software solution, providing documented evidence that the solution works and plan for its implementation (related to section 5.1.1);
- prepare user documentation for the software solution (related to section 5.1.3).