Examination report – June 2014 series

2730-018 Programming Principles

Section 1 – Areas of good performance
Syllabus reference: 1.5 / 1.8 – Describe the main computer components crucial to program execution and discuss data handling by the CPU and the role of software to deal with data entities. The purpose of each of the computer components; disk storage, Random Access Memory (RAM), Central Processing Unit (CPU) and Input/Output (I/O), crucial to the execution of a program were described satisfactorily. However, very few candidates gave suitable reasons why data handling software is necessary. For example code conversion, buffering, error detection / correction. Syllabus reference: 1.19 – Identify data types, values and their representation. Generally answered satisfactorily although in part b) the majority of candidates thought real numbers were integers. Syllabus reference: 1.25 / 1.26 – List components of a program specification and explain detailed specifications. Solutions to this section, on the whole, were good. Syllabus reference: 1.27 / 1.28 – Function and use of algorithms and a flowchart for a given process showing process boxes, decision boxes, input/output boxes, using a simple routine. The flowcharts in this sections were sometimes lacking detail. Syllabus reference: 1.29 – Describe a top down design for a given process. Solutions to this section were satisfactory. Syllabus reference: 1.42 / 1.44 – Explain databases, query languages and fourth Generation Languages (4GL) and define Standard Query Language (SQL) and 4GL and their relations to relational database systems. Very few candidates could state suitable reasons for using a 4GL. For example programmer productivity, shorter development time, natural language etc.
Section 2 – Areas for development
Syllabus reference: 1.12 / 1.13 – Examples of code for a given low-level language and analyse program code for a small program and explain the command lines and their function. Very few candidates could provide suitable instructions from the types of instructions; arithmetic, logical and branch/jump from an assembly language. Candidates were unaware of memory mapped I/O for why some CPUs do not have dedicated input/output instructions. Syllabus reference: 1.15 / 1.16 – Identify examples of coding for a high-level language and obtain and analyse a program code for a high-level language. Solutions in this section were very poor. Candidates failed to identify one of the most popular high level languages. The explanations to the purpose of the program and its operation were very vague. Syllabus reference: 1.23 – Describe concepts and usage of subprograms. Very few candidates could provide suitable benefits of using functions as an aid to program structure (code reduction, improved layout etc.). The functions produced, that would add three integer values, were often inaccurate, incomplete or lacking detail.

Syllabus reference: 1.45 / 1.46 – Identify display formats and explain the difference between display coding (tagging) and computer languages. Produce coding for a simple HTML Web page using a plain text editor ie without the use of software, which outputs as HTML. Disappointing results for this section. Candidates were required to use FORM and INPUT code tags.
Section 3 - Recommendations
Candidates need to improve their knowledge in the following: • Knowledge of assembly language instructions • High level languages • HTML coding.