Examination report – June 2014 series

2730-025 Software Engineering

Section 1 – Areas of good performance
Syllabus reference: 1.6 / 1.7 – Describe the waterfall model of software development in terms of comprising; specification/requirements analysis, design, implementation, testing and maintenance. State the importance of documentation. On the whole, this section was answered very well. Syllabus reference: 1.13 – List and describe the benefits of formal specification. Very few candidates were able to describe two benefits of a formal specification. Syllabus reference: 1.16 – Draw a simple state diagram. Overall, solutions to this section were good. State diagrams were sometimes inaccurate and poorly labelled. Syllabus reference: 1.18 – Define the terms cohesion and coupling. Some candidates did not know the difference between cohesion and coupling. Syllabus reference: 1.32 – Describe, with the aid of a diagram, the structure of an IPSE. Solutions to this section were often vague and inaccurate. Syllabus reference: 1.37 – Describe the relative merits of top-down and bottom-up testing. On the whole, this section was answered satisfactorily. Several candidates answered this section in relation to 'program development' and not 'testing'. Syllabus reference: 1.44 / 1.45 / 1.46 – Differentiate between scripting and system languages. List scripting languages. List system languages. Overall this section was answered satisfactorily. Solutions to how a major security issue may occur on a computer when using a scripted language to access an external source of data were often incorrect. Syllabus reference: 1.47 / 1.48 / 1.49 – Identify and describe at least one programming language from each of the programming paradigms; object-orientated, functional, logical, visual and scripted. Recognise and describe the function of OOP components. Identify and discuss the merits of OOP languages. This section was answered satisfactorily although candidates rarely stated the main benefits of using OOP (reusability, maintainability, dealing with complex problems etc.).
Section 2 – Areas for development
Syllabus reference: 1.57 – Describe the fundamental differences between given SGML sub-sets. The relationship between SGML, HTML and XML was not generally known by the candidates. Candidates were unable to produce five differences between HTML and XML language coding. Syllabus reference: 1.60 / 1.66 – Describe a method of programming in the Unix environment. Give examples of the fundamental Unix file access primitives including 'open', 'create', 'close', 'read' and explain their functions in writing Unix programs. Solutions to this section were very poor. Very few candidates could describe four system calls associated with Unix file access.

Section 3 - Recommendations
Candidates need to improve their knowledge in the following areas: • SGML, HTML and XML mark-up languages • Unix environment – System Calls.