Examination report – June 2012 series

2730-025 Software Engineering

Section 1 – Areas of good performance
Syllabus reference: 1.5, 1.6 –The need for software engineering – software process. This question on the whole, was answered very well. Syllabus reference: 1.13 – Software specification / Requirements analysis. The majority of the candidates were unable to provide a drawback. Syllabus reference: 1.15 – Software specification / Requirements analysis. Overall solutions were good although few candidates could state reasons why finite machines are used. Syllabus reference: 1.24, 1.25 –Programming practice and software tools. Solutions lacked in detail. Syllabus reference: 1.44, 1.46 – Programming languages. Generally answered satisfactorily. Syllabus reference: 1.64 – Programming languages. Generally answered satisfactorily.
Section 2 – Areas for development
Syllabus reference: 1.22 – Software design and implementation / Design methodology. This was the least popular section on the paper. Candidates showed a complete lack of understanding regarding the use of Jackson Structured Programming (JSP). Syllabus reference: 1.35 –Software validation / Testing. Candidates had trouble distinguishing between validation and verification. Few candidates were able to describe suitable methods used for software verification. Syllabus reference: 1.49 – Programming languages. This section asked for a description of the structure, semantics and syntax of an OOP language specific to one that the candidate knew. Most candidates just explained the meaning of the terms structure, semantics and syntax. Syllabus reference: 1.51 – Programming languages. The candidates knowledge of Unified Modelling Languages (UML) was very poor. Few candidates could give two benefits of using a UML. Candidates were unable to describe the functions of the UML diagrams.
Section 3 – Recommendations
The results for this examination were disappointing. Candidates need to develop a better knowledge of creating state diagrams and using Jackson procedural notation. Other topics requiring attention are programming language details (structure, syntax, semantics), software testing and UML.