Examination report – June 2011 series

2730-025 Software engineering

Section 1 – Areas of good performance
Syllabus reference: 1.4, 1.5, 1.7 – The need for software engineering/software process. Solutions to this question were very good. Syllabus reference: 1.10, 1.11 – Software specification. This question on the whole, was answered very well. Some candidates still confuse systems modelling with prototyping. Syllabus reference: 1.16 – Programming practice. State diagrams were often inaccurate with some states left out. Syllabus reference: 1.23 – Programming practice and software tools. Solutions to this question were very good. Syllabus reference: 1.28, 1.29, 1.34 – Programming practice and software tools. This question on the whole, was answered very well. Very few candidates understood the meaning of 'pragmatics'. Syllabus reference: 1.39, 1.40 – Software validation/testing. On the whole answered satisfactorily. Not all candidates had a good knowledge of equivalence partitioning. Syllabus reference: 1.47 – Programming Languages. Solutions to this question were very good. Syllabus reference: 1.49 – Programming languages. On the whole answered satisfactorily. The majority of candidates chose C++ and Java.
Section 2 – Areas for development
Syllabus reference: 1.22 – Software design and implementation/design methodology. Solutions to this question were very poor. Very few candidates were able to design a suitable pseudo-code program. Syllabus reference: 1.61 – Programming languages. Very few candidates knew the meaning of a Unix metacharacter and therefore were unable to obtain marks.
Section 3 – Recommendations
Overall, this year shows an improvement in candidate performance. Candidates do need to develop a better knowledge of pseudo-code programming. As a large part of the syllabus is dedicated to Unix, it would make sense for candidates to apportion more effort to this topic.