Coursework commentaries 2015–16

CO3325 Data compression

Coursework assignment 2

General remarks

Most students did a good job for this coursework assignment, although a minority submitted incomplete work. Again, many students enjoyed the assignment and reported a valuable learning experience. Examiners noted that weak coursework seemed to be a result of poor time management or limited problem-solving skills.

Strong coursework was characterised by careful design, with reports that provided precise accounts of the approach taken, together with insightful discussion. Weak coursework tended to be incomplete, with limited design and coding, and reports that were vague and provided only superficial discussion (or no discussion at all).

Some students did not follow the advice to do the preparatory exercises before attempting the coursework. Instead, their starting point was to search for an existing solution to solve the given problem (sometimes without even properly studying the problem).

Some students were uneasy about making their own decisions, and would have preferred to be told what to do for predefined cases. However, in many real-world situations, customers have only vague ideas about what they want from the software system to be developed. An important initial step for any software development is to specify the requirements and identify the algorithmic problems. Reasonable assumptions also need to be made sometimes to simplify computational problems. This assignment mimics this situation to help prepare you for real-world problem solving in software development.

Good submissions focused on design before implementation. Some students only realised that they should have done this after completion of the whole coursework assignment – an extremely valuable learning experience for all future development work.

Comments on specific questions

Question 1

This coursework assignment requires the development of a prototype Java program. It is important to note that this is Level 3/6 coursework, preparing you for the real world of software development in which you will have to learn something without explicitly being taught.

As with Coursework assignment 1, an efficient way to complete this assignment is simply to follow the basic Software Engineering approach (outlined above). Start the process by analysing the given problem. You could then apply the ‘divide and conquer’ method to break a big problem into a number of smaller subproblems. Here, you may use the block diagrams to help you.

Unlike Coursework assignment 1, you need to design your own Pyramid algorithms. You will need to understand what is required from the given examples. You may also need to make decisions to handle any situations that emerge but are outside the given examples.

A good program would first need to meet all the given requirements, correctly and efficiently transforming the corresponding output data from the input data across a reasonably wide range. An easy way to realise this is to use variables instead of fixed data examples. Next, your algorithm or program should be reasonably efficient, so some analysis of time complexity would be helpful. As this is a preprocessing algorithm, the best submissions would also exploit compression implications either in conjunction with some learnt compression techniques, or by experimenting using different parameters; for example, in the selection of different ‘representatives’.

Comments on report structure follow those for Coursework assignment 1 above.