Coursework commentary 2017–2018

CO3325 Data compression

Coursework assignment 1

General remarks

A good approach to attempting this coursework assignment would be to simply follow the given instructions and guidelines. For example doing exercises to understand the relevant concepts before attempting the assignment tasks, reviewing the Software Engineering approach (see Subject guide CO2226 Software engineering, algorithm design and analysis – Volume 2) to complete a software development journey as a professional. The milestone stages include:

1. Working on the given problem specifications →designing rough solutions in block diagrams.
2. Designing fine solutions in class diagrams →finalising the design with general analysis → class diagrams or pseudocodes.
3. Testing.
4. Evaluation of your software.

A successful coursework assignment often reflects good efforts. For example, a dedicated piece of work should be presented, starting with a considerate design, offering comprehensive details to serve clear purposes. Be sure to make wise choices concerning data structures and sensible decisions for efficient approaches to problem solving. The report should be clearly structured, well-written with insightful discussions on technical issues and demonstrate personal experience of attempting the coursework assignment. The programming style should be neat with sufficient comments. Screenshots should be used to highlight the execution process of the programs.

Comments on specific questions

It is a great pleasure to see more outstanding coursework assignments this year. In addition to having all the required components completed to a high quality, extra efforts had obviously been made to extend some aspects of the assignment. For example, studies and implementations of new algorithms, applied linked lists or other advanced data structures, extending the algorithms to apply to an alphabet of any size, using extra functions such as NumberFormat, long 64 bits float computation, implementation from scratch, and automatic demonstration.

Most students implemented the algorithm for the smallest alphabet as the binary form of (a,b). In fact, this can easily be extended. For example, even an attempt to implement the extension algorithm could gain extra marks.

Many students seem to have been disadvantaged by having difficulties in Java programming instead of those in Data Compression. They described themselves as “having limited Java skills”. Some even said that it was the first time for them to develop a program from scratch. This reflects a certain weakness in their knowledge of other foundation modules.

However, almost all the students who attempted both coursework assignments for CO3325 said that they learnt a lot, both in understanding the compression algorithms and in software development.

Some students said they found the coursework assignment 1 a little bit vague with no explicit questions nor specified format requirement. This, however, is quite common in real-world practice. The customers often do not know what they require precisely, and it is part of a programmer's job to help them design a specific use case and requirements for the project system.

Coursework commentary 2017–2018

CO3325 Data compression

Coursework assignment 2

General remarks

A good approach to attempting this coursework assignment would be to simply follow the given instructions and guidelines. For example, doing exercises to understand the relevant concepts before attempting the assignment tasks, reviewing the Software Engineering approach (see Subject guide CO2226 Software engineering, algorithm design and analysis – Volume 2) to complete a software development journey as a professional. The milestone stages include:

1. Working on the given problem specifications → designing rough solutions in block diagrams.

2. Designing fine solutions in class diagrams →finalising the design with general analysis → class diagrams or pseudocodes.

3. Testing.

4. Evaluation of your software.

A successful coursework assignment often reflects good efforts. For example, a dedicated piece of work should be presented, starting with a considerate design, offering comprehensive details to serve clear purposes. Be sure to make wise choices concerning data structures and sensible decisions for efficient approaches to problem solving. The report should be clearly structured, well-written with insightful discussions on technical issues and demonstrate personal experience of attempting the coursework assignment. The programming style should be neat with sufficient comments. The screenshots highlight execution process of the programs.

Comments on specific questions

More students seemed to have spent more time on Coursework 2 than they did on Coursework 1, showing the interests, efforts and dedications.

A good starting point was to decide which compression algorithm to use. First use your own compression algorithm that you developed in Coursework 1, then if time permits, you may implement another compression algorithm for your new program.

Next, you should decide what the input and output would be. Most students used the input.txt and output.txt auxiliary files for their compression system. This is a good choice due to their flexibility.

Quite a lot of students were very creative and applied the data compression ideas nicely on various compression problems. Some focussed demonstration programs as a proof of concept.

Students who attempted the second coursework assignment generally felt their programming skills had been improved and again almost all the students said that they learnt a lot from completing theassignment, both in understanding the compression algorithms and in software development.