Overall Statistics

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>60</td>
</tr>
<tr>
<td>Max</td>
<td>80</td>
</tr>
<tr>
<td>Min</td>
<td>28</td>
</tr>
<tr>
<td>Mode</td>
<td>58, 69, 75</td>
</tr>
<tr>
<td>Median</td>
<td>61</td>
</tr>
<tr>
<td>StDev</td>
<td>13</td>
</tr>
</tbody>
</table>

There were 3 values for the mode, as the mode count was 3 and we had three 58s, 69s, and 75s.

The following chart shows the average, max, and min score for each question.

For each question at least one person scored the maximum and one person scored the minimum. If you follow the blue bars across the questions, you'll see that there was a balance of difficulties with some rather easy questions (Q3 or Q4 or Q20) and some very hard ones (Q17 or Q23 or Q27).

**Detailed Feedback**

Q24 - Q27 were short answer questions that checked whether students had a good understanding of the basic concepts of this course unit. As expected, they did mostly well on these, though not as good as hoped for:
Q24: most students gave a basically sufficient answer.

Q25: though many students gave a correct basic answer, surprisingly few students were able to describe the two aspects of validating parsers (namely (1) testing validity against a given schema and (2) possibly adding schema information to the document model)

Q26: though many students gave a correct basic answer, surprisingly few students were able to describe the relationship between infoset/DOM tree nodes/edges and XML doc elements/nesting structure in a suitable way.

Q27: though many students gave a correct basic answer, surprisingly few students were able to describe the *relevant* information that is represented in a PSVI and where it comes

Q28: concerns one of the more advanced subjects of the course unit. Some students failed to realise that DTDs correspond to single-type grammars (because they are local!) and thus reasoned 'in a wrong direction'. Almost no student reported correctly that non-single-type schemas (eg in RelaxNG) are perfectly fine to describe/check/ensure constraints on XML documents "as long as we don't want to use this schema for the generation of a PSVI", partly because many students misunderstood the complexity of validation against non-single-type schemas....it is basically as costly as for single-type schemas, as we have seen in class via the algorithms we considered.

Q29 there was no right or wrong answer here, and many students were able to mention a good number of valid points relevant for this question (each of which resulted in 1 mark). There were, however, some students who - despite writing rather many words, only mentioned very few valid points that were relevant for this question.

Q30:

While Schema aware XPath includes XML Schema, that’s not the relevant aspect. The key idea is using queries as tests for which positive answers yield validity. The ultimate exemplar of this approach is Schematron.

XPath is generally more expressive (content tests, far reaching coordination, counting, etc.). It’s not good for "complete" enforcement of a grammar.

Neither XPath or DTD have inherent memory or performance advantages or disadvantages, at least for the "same" functionality. One can certainly write XPath queries that are arbitrarily expensive, but typically that’s because we are doing something we can’t do in DTDs.

Q31: I have to say that this is the absolute best set of answers I’ve ever seen for this sort of question. People have practiced their DTDs! There were amazingly accurate syntactically.

ATTLISTs tend to get a bit messed up...people forgot the type or the required/implied/default, or added everything. Some people made nick an "ID" which is an interesting balance between tightness and flexibility.
<person> caused some trouble. People forgot to declare it altogether. I treated a missing content model as an "empty" declaration.

Q32
Due to a error in the system, some grades were misrecorded. Generally this was to give a higher grade (indeed, the top grade was incorrectly a “6”. Where this benefited the student it was left intact and where not it was corrected).

The biggest problem here was misunderstanding the question. Relax-NG has two syntaxes: An XML one and an “human”/DTDish one. The question is which syntax is more error prone for *published* ontologies. Most arguments need to trade tooling for complexity/verbosity.

Some people got confused about whether an XML format required a schema. Of course, it doesn’t, but it might be more error prone as you can’t check your instances.