CITS5501 Short Exercises Assessment Rubric

2021-03-28

Version: 0.1.0

For the "short exercise" assessments, you will be given some short questions and/or a scenario to analyse and asked to demonstrate that you can:

- distinguish the relevant from irrelevant facts in the scenario
- identify what topics we have covered which apply in this case, and how
- come up with a clear recommendation or answer
- justify that recommendation (logically, or via evidence covered in lectures etc.)

"Identifying relevant facts/topics" and "justifying answers" are the key things we are looking for in workshop answers.

Each question in an exercise will be marked out of 5 (2 + 2 + 1). There are 2 marks for identifying relevant facts/topics, 2 marks for justifying answers, and 1 mark for clarity/legibility.

Demonstrated level	Identifying relevant facts/topics	Justifying answers
Proficient (2 marks)	All (or nearly all) relevant facts/topics identified, with at most minor omissions/errors	Nuanced justification and reasons for answers
Satisfactory (1 mark)	Some relevant facts and topics identified but some major omissions/one-sided arguments	Some attempt at justification but major omissions or errors e.g. fails to justify major points or identify key assumptions
Not yet satisfactory (0 marks)	Little or no evidence of ability to identify relevant facts/topics	Little or no evidence of ability to justify answers

Finally, 1 mark is awarded for clarity and concision: 1 mark if the submission is clearly and concisely expressed, 0 if it is not.

Example

As an example, consider the following scenario:

You and a colleague are reviewing a unit test for the address-book application you are working on.

The unit test currently:

- creates an in-memory database of people and their contact details
- calls an `addPerson()` method, which takes as argument the full name of a person to be added to the address-book
- checks the database to ensure the new person was correctly added.

Your colleague says this test follows best practices for unit tests, and recommends it be kept in the test suite. Do you agree or not? Briefly explain, justifying your answer.

A sample proficient answer:

> My colleague is incorrect – this test does not follow best practice for unit tests. The reason is that it as part of its setup, it creates an in-memory database; in general, *unit* tests should not depend on other components, but should rely on *mock* objects to mimic their behaviour, if needed. That is what should be done here. Reasons for this are that:
>
> - if the test does not pass, it may not be clear whether the error is in the code being tested, or the database. Using mocks helps isolate the problem.
> - unit tests should run quickly, and creating and populating a database – even an in-memory one – is likely to be a slow operation.

This answer identifies relevant information (namely, best practices for unit tests), clearly states a conclusion (the colleague is incorrect), and justifies the conclusion (based on properties we want unit tests to have).