Report Writing for COMP1100

Madeleine Stewart

23/08/2023

What’s the point of reports?

- Programs are complex and unique, they need explanation
- Explain your design process
- Justify your program’s correctness

How are reports marked?

Category	Marks
Documentation	4
Reflection	4
Testing	4
Style	3
Total	**15**

Programs are complex and unique, they need explanation	*Documentation*
Explain your design process	*Reflection*
Justify your program’s correctness	*Testing*
Your report should also be easy to read!	*Style*

Structuring a report

Introduction	
Analysis of your program	(Documentation)
Rationale and Reflection	(Reflection)
Testing	(Testing)

Documentation

Show us that you understand your program

Program as a whole

- Explain the program structure
- How do your functions work together to create your program?

Individual functions

- What is the purpose of each function (beyond what's given)?
- Explain how complicated functions work

Common mistakes

- Line-by-line explanation of code
- Copying the assignment specification

For a high mark in this section, we expect the report to demonstrate a conceptual understanding of all relevant functions and depict a holistic view of program structure.

Documentation – examples

A problematic example:

“`shapeToPicture` turns a `Shape` type into a CodeWorld `Picture`.”

A better example:

“To draw shapes to the screen, the `Model` needs to be converted to a `Picture`. The `shapeToPicture` function assists in this by … and it does this by …”

Don’t just repeat the assignment specification, show us that you understand your code!

Reflection

Why is your program the way that it is?

Describe your design process

- How did you solve the problems you encountered?
- Insights into unsolved problems and/or possible improvements

Explain your Reasoning

- Reasons behind design choices
- Assumptions
- Tradeoffs

Common mistakes

- Personal reflection instead of a technical one – we're interested in your code

For a high mark in this section, we expect the report to contain a thorough explanation of the design process, including relevant reasoning and assumptions.

Reflection – examples

A problematic example:

“This assignment included some technical complexities that I struggled to overcome. With more time and a greater understanding of Haskell, I would hope to do better.”

A better example:

“The `toolToLabel` function only changes the `Tool` when it is not in use. To simplify the code, the patterns that result in a change were placed first, allowing use of the wildcard to capture all other cases and return the `Tool` unchanged.”

We’re asking for a technical reflection, not a personal one!

Testing

How do you know that your program works (or not)?

Describe your testing methods

- Whole program
- Individual functions

What do the testing results show?

- Which parts of the program are different tests testing?
- Why are your tests enough?

Common mistakes

- Not including concrete examples
- Not discussing correctness

For a high mark in this section, we expect the report to show evidence of testing of all relevant parts of the program and include a discussion on why these results imply correctness.

Testing – examples

A problematic example:

“I tested my code to ensure that everything worked.”

A better example:

“To test the `shapeToPicture` function, I identified … testing groups. I tested each testing group by … The results of these tests were … and this gave me confidence that this part of the code was correct because …

Include concrete examples of tests, for the whole program and individual functions, and why they show program correctness (if they do)

Style

Make your report easy to read and mark

The better we understand you, the more marks we can give

- Good report structure
- Good formatting
- Good editing

Common mistakes

- There are many!
- Not leaving enough time for the report

For a high mark in this section, we expect the report to be easily readable, with good formatting.

Style – examples

A problematic example:

"when the calculateDistance funciton was included as a helper function in shapeToPicture it was observed that the repetition of code was reduce and as a result it became more convenient to perform isolated testing"

A better example:

"Including `calculateDistance` as a helper function to `shapeToPicture` reduced code repetition and allowed for more convenient, isolated testing."

Please make your report easy and nice to read!

Some pointers

Don’t forget about your report!

- Don’t leave it to the last minute
- Leave enough time to edit
- Take notes as you go

Double check the assignment specifications

Be concise

Make use of course and ANU resources

- Look in resources section of COMP1100 website
- ANU academic skills