

How To: Write a Programming Report

Felix Palludan Hargreaves

December 1, 2011

Contents

1	Introduction	3
2	Getting started	3
3	Sections	4
4	Language	5
4.1	Figures	5
5	Programming	5

1 Introduction

This paper contains a few important points on how to write a scientific report on a programming project. Everything here is subjective and representative of the authors opinion only. Use at own discretion.

2 Getting started

Even though the project description was read before working on the given problems, it should be re-read before the report is written. Special attention should be paid to requirements, both explicit and implicit. Make sure to take note of requirements to the format of the report.

3 Sections

The report should contain the following sections:

- **Specification**
 - Describe the task at hand. Make sure to distinguish between the given problem and your own additional goals (if any). Possibly refer to the project description. Be *very brief* and *concise*.
- **Design**
 - Describe the algorithm(s) that was(were) designed. Avoid describing how the design was produced, rather, show different versions of the design if relevant. **Use pseudo code to avoid ambiguity.**
- **Implementation**
 - Show the implementation. The goal of this section is to show and explain the most important parts of the code. **Listing the code with highlighting and possibly line numbering is essential.** Explain the code by referring to line numbers, function calls and variable names. Leave out trivial parts (initialization, parameter-tuning, etc...).
- **Testing**
 - Show tests of final code version. Tests include correct input, incorrect input, special cases and stress tests. Method of testing will vary with the problem at hand. This section is **very important**. Display the tests with this format:
 - * Input
 - * Expected output
 - * Actual output
 - * Comments
- **Conclusion**
 - Describe to which degree the product meets the requirements of the project. If relevant, describe how the performance of the product compares to your expectations (or to state of the art).
- **Appendices**
 - Full source code listing
 - Extensive test output (if relevant)
 - Make scripts, auxilliary scripts, etc... (if relevant)

4 Language

A scientific report should be written in a scientific language. English is preferable. Passive form is also preferable.

- **Passive form (better):** The program was written with performance in mind
- **Active form (worse):** I wrote the program with performance in mind.

Passive form prevents reports with list-like language. Example: "Then I did this...then I did that...Then I wanted to..."

In general, the process is not as relevant as the product. Explain and motivate the design, try to avoid explaining things that didn't work or didn't make it to the final product. It can be beneficial to show different versions of the design, especially when they have different strong points.

Humor is not forbidden, but should be limited to places where it does not add unnecessary text or confuses the reader.

4.1 Figures

A rule of thumb: **More figures, less text!** Figures are extremely helpful in preventing ambiguity and keeping the report *concise and precise*. Keep figures relatively close to their references.

5 Programming

It can be beneficial to simplify the code and to revisit the program design in order to make code presentation easier. Furthermore, this will add to your understanding of the problem. Understanding your solution better will aid you in describing the implementation in a clear and concise language.