Project on a topic in Computer Science

Candidate no. 0000000

Word count: 0000

Part B - MCompSci Computer Science

Trinity 2024

Acknowledgements

You might want to thank your supervisor or anyone else who supported your project, your friends or family who stayed up at night to keep you company, or the guy at your favourite coffee shop who you saw more often than you saw your own feet.

I'd like to thank my supervisors for encouraging me to explore the power of LaTeX and John McManigle for providing such an effective and helpful starting point for this template.

Abstract

OxCSProject is a LaTeX template intended to encourage you to become familiar with using LaTeX, give you a helpful starting point when it comes to structuring your project report, and ensure that you have easy and consistent formatting for your project. OxCSProject is an adaptation of very helpful work done by those intending to serve the same purpose as OxCSProject in different aspects of Oxford academic life, and would not exist without them.

OxCSProject is also designed to be an introduction into the basic tools that I expect you'll find helpful throughout the process of writing your report, as well as introduce you to other tools that you may appreciate being aware of. Lastly, ways for the value of OxCSProject to be enhanced are laid out, so that you can contribute any additional development for future students to enjoy.

Contents

1 Introduction **2**

1.1 Motivation 2

1.2 Structure of the Document 2

2 Installation **4**

2.1 OverLeaf 4

2.2 Offline 5

3 Background **6**

3.1 Benefits of using LaTeX 6

3.2 Challenges of working in conventional word processors 6

4 Usage **8**

4.1 Using Images 8

4.2 Using Tables 9

4.3 Other features 9

5 Other tools **10**

5.1 Matplotlib 10

5.2 Zotero 10

5.3 Draw.io 11

6 Future Work **12**

6.1 Things to add 12

6.2 How to contribute 12

Appendices **13**

A Code listing **14**

References **16**

1 Introduction

OxCSProject is a (hopefully) easy to use LaTeX template that you can use to develop and format your 3rd or 4th year Computer Science (CS) project at the University of Oxford. This should allow you to avoid all of that silly formatting nonsense and stick to the thing that you're really here for, your project.

This template comes as the full package, and even compiles to its own user manual, which can also be found alongside the template on GitHub [1] once you inevitably start writing up your own work. However, that doesn't mean you shouldn't modify, extend, and augment the class to fit your needs. That might mean a project for a degree that *isn't* Computer Science, updating it to match new formatting requirements, or using your own set of packages and tools to fit your project.

If you do adapt this template to suit another purpose, such as a project for another Oxford degree, I encourage you to publish and advertise it just like I do here.

1.1 Motivation

I wanted to make this template because navigating LaTeX and trying to adapt and modify John's template while I was conducting my project was one more thing that I needed to do. It was certainly worth it, it's something I realised that I could do for your project too, as well as ahead of my own Part C project. I'm hoping that making it easy to use LaTeX in your project will mean you choose to do so, making your project better than it otherwise would have been and helping you develop yet another skill for your future endeavours.

1.2 Structure of the Document

The rest of this document is structured as follows:

Section 2 describes how to install this template to OverLeaf [2], an easy to use online LaTeX editor. It also includes how to install this template so that you can

develop your project writeup offline if you prefer.

Section 3 discusses the reasons for working with LaTeX over methods of developing your project writeup, such as Microsoft Word or Google Docs.

Section 4 demonstrates how to use some of the features of LaTeX you may find helpful to be familiar with while preparing your writeup.

Section 5 introduces some of the other tools that it's worth becoming acquainted with to make your project as good as it can be.

Section 6 explains how you, dear reader, can add to this template for the benefit of anyone who might embark on a project in future.

2 Installation

For the purpose of your project, I strongly encourage you to use OverLeaf [2] while you become familiar with this new and exciting technology. Once you feel a desire to learn more about LaTeX, such as in the summer after your project of choice, it is worth considering writing something in a code editor or IDE such as Visual Studio Code (VS Code), as I am for this template.

2.1 OverLeaf

OverLeaf (`www.overleaf.com`) is an online LaTeX editor that makes it easy to get started with using LaTeX avoiding having to install a TeX distribution such as TeX Live [3] onto your PC and letting you share your writeup with your supervisor to make getting feedback as easy as possible.

As a disclaimer, these steps may not be perfectly accurate, since I'm writing them from memory and they are, of course, subject to change.

1. Navigate to `www.overleaf.com` and create an account using your Oxford login. Either your college (e.g., your.name@college.ox.ac.uk) or your department (e.g., your.name@cs.ox.ac.uk) alias is fine, since you can later add the other as a secondary email. By using an Oxford email, you will get free access to the premium features.

2. In your user settings, link your OverLeaf account to a GitHub account.

3. Go to `github.com/Ned-Stevenson/OxCSProject` and create a fork of OxCSProject. This will create a new repo that you own as a copy of OxCSProject, which you can then import into OverLeaf.

4. At the OverLeaf home page (`www.overleaf.com/project`), create a new project, and select "Import from GitHub". From the list, select your forked repo.

2.2 Offline

To install offline, you need only clone this repository into whichever folder you'd like to work in. I would recommend making a fork of the repo and cloning that so that you can version control your work, protecting your efforts in the case of loss, damage, or theft of your favourite LaTeX writing device. You will also need to install something to render your LaTeX into a PDF, such as TeX Live [3] and an IDE to work in.

1. (Optionally) Go to `github.com/Ned-Stevenson/OxCSProject` and create a fork of OxCSProject

2. Clone this repo or your fork onto your local device:

   `git␣clone␣https://github.com/Ned-Stevenson/OxCSProject.git`

3. Install a tool such as TeX Live [3], following any relevant instructions.

I can also recommend the LaTeX Workshop VS Code extension [4] as well as latexindent [5] to make it easy to both render your code as well as keep it tidy and readable. Note that latexindent comes pre-installed with TeX Live, and the configuration options that I used to format this project are available in the GitHub repository.

3 Background

3.1 Benefits of using LaTeX

LaTeX brings all of the benefits of software development, particularly the ability to reuse work (both your own and other people's) through the use of subroutines and packages. In LaTeX, packages are widely available for many different uses, some of which I try to demonstrate in Section 4. They are imported through the use of the `\usepackage` command, and allow you to subsequently use the commands that they define. You can define your own subroutines using the `\newcommand` command, helping you stay on top of keeping your documents concise and fast to work on.

Another benefit that LaTeX affords us is the separation of our work into multiple, more easily navigated files. I found it helpful to separate my project to include one file per section, to keep any one file where I was actually working and writing content to a manageable length. Of course you don't need to copy me if you prefer to write your documents in a single monolithic file, but at least you have the option.

3.2 Challenges of working in conventional word processors

One of the challenges of working in a word processor is the way that it can slow down as the size of the document you're navigating grows. This can get quite frustrating, particularly as you try to get the finishing touches on your formatting the night you're trying to submit your writeup.

Another challenge is working with automatically generated components of your document, such as bibliographies, tables of contents, and glossaries. In LaTeX, this type of thing can be neatly tracked for you by whichever package you're using. And often formatting can be as simple as a sequence of command arguments, rather than doing it manually after regenerating your tables.

And not least, displaying your code, something I hope you'll do a fair amount, feels far more challenging in a word processor than in LaTeX. Your only real option

is to copy from your IDE and keep formatting, which doesn't let you elegantly deal with long lines of code, change your formatting after the fact, or keep your code in its source files and import it into your document directly. All these things are handled much more elegantly in LaTeX, as will be demonstrated in Section A.

4 Usage

This section is intended to demonstrate some of the basic LaTeX commands and features you may find helpful to be familiar with. This is likely not a comprehensive list, so please do go looking for other features and packages that you think you may find helpful. As with everything else, Stack Overflow is the best teacher there is!

4.1 Using Images

Throughout your project writeup, you will need to include figures, such as Figure 1 below. This is generally done using the `\includegraphics{}` command. However this doesn't include `.svg` files, so the `\includesvg{}` command from the `svg` library should be used in that case instead. Wherever possible, you should aim to use vector graphic formats such as `svg` or `pdf` to ensure there's no chance of low resolution, hard to read images in your writeup.

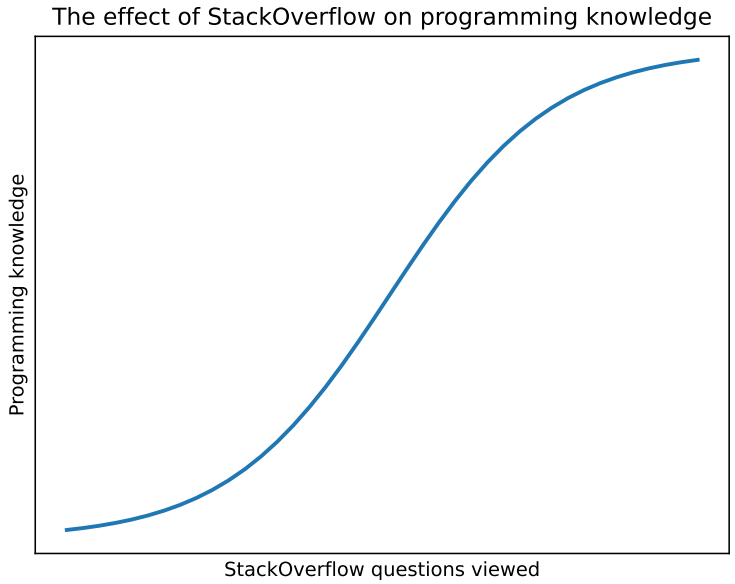

Figure 1: A plot of Computer Science knowledge against Stack Overflow use

4.2 Using Tables

You can see below a short list of some of the more helpful LaTeX commands to know, as well as a demonstration of how to use tables. It's okay if not all of these are helpful to you, but it may be a good starting point for things to be aware of.

`\cite`	Insert a citation using the bibliography key in your `references.bib` file
`\label`	Add a point that can be referenced elsewhere in the document, such as on a section heading, figure, or table
`\ref`	Add a reference to a label elsewhere in the document. You should precede this with `~`, which is a whitespace that won't be split across new lines.
`\SI`	Add a number formatted with SI units e.g., `\SI{5}{\kilo\byte}` = 5 kB
`\verb`	An inline way to render text exactly as written in a monospace font, like the commands in this table.

4.3 Other features

Throughout this template, I've also used a few other features that you ought to make use of, but which weren't big enough to deserve a whole subsection. You're encouraged to go out looking for them throughout this template to get a handle on how to use them. Some are also mentioned in the table above with explanations of how they can be used.

- Citations
- Acronyms
- References to sections and figures
- Lists and enumerations

5 Other tools

This section describes the range of tools aside from LaTeX that you may want to be familiar with while you're working on your project. This is not intended to be a comprehensive list and is only the tools that I found useful in my project. Alternatives of course exist to everything I talk about here, and you should investigate what works for you before simply copying my method.

5.1 Matplotlib

Matplotlib [6] is a Python library that provides easy to use graph plotting functionality, along side other visualisation features that you may find helpful. By producing your graph plots in Python, you can integrate it easily into the data processing stage of your project, if one exists.

This lets you be very versatile with how you display your data, without needing to work within the more constrained options of something like Microsoft Excel. I also found it more intuitive to understand the relationships I'm trying to demonstrate when working in Python compared to a spreadsheet.

See Figure 1 for an example of a plot made using Matplotlib, and see appendix A.2 for source code of the plot.

5.2 Zotero

Zotero [7] is a reference manager that allows you to easily collect and keep track of all the references that you'll need to include in your project writeup. By using the Zotero browser extension [8] and the Better BibTeX add-on [9], you can make it easy to add to your bibliography with the browser extension. The Better BibTeX add-on will then make sure that the citation keys that are exported from Zotero are consistent and do not clash, making it easy to see what you're referencing when writing your LaTeX.

If you use this method, make sure you set the format to Better BibLaTeX when exporting to your `references.bib` file (or whatever name you end up using for your bibliography file).

5.3 Draw.io

Draw.io [10] is a tool used to draw whatever diagrams you need to explain the core concepts of your project. Maybe it's important to explain an inheritance diagram, relational database, or just a simple flowchart for your code.

You can either use the online editor or download the draw.io app to use locally. You can then export the diagrams you create to a bitmap image format, such as `.png`, or to a vector format, such as `.svg`. I recommend keeping your draw.io save file in the folder with your figures whether or not you're using an OverLeaf project.

6 Future Work

6.1 Things to add

At time of writing, it doesn't feel like there are many particularly pressing missing features, save one. I found at the end of my 3rd year project that getting a word count was much more painful than it needed to be, since I discovered that figure captions are not included in the built-in OverLeaf wordcount. I ended up taking the PDF produced by my LaTeX and copying any text to be included into a Word document. I then took the count that I got from that document over to my LaTeX to be re-rendered. I looked briefly for a word count library, but could find surprisingly few packages that seemed to offer a word count, and none that I noticed were suitably configurable to make the word count usable. If there is one you're familiar with, please read on!

For a current list of missing features and required bug fixes, please see the GitHub issues page.

6.2 How to contribute

If you feel that there are any features that need be added or changes to be made, please go to the GitHub repository where you got this template and add an issue for it. And if you feel like trying out working on open source projects, consider picking an issue and implementing it as I describe below.

If you do want to do the work that resolving an issue would require, then make a fork of OxCSProject in the same way as described in Section 2. From there, you can make whichever changes and write in whichever features you'd like. To make them available for all to enjoy, you can then create a pull request to merge those improvements back into the template for the next generation of Oxford Computer Scientists. This is the same process as contributing to pretty much any open source project, and something worth trying at least once.

Appendices

A Code listing

This allows us to demonstrate the code that we've written for our Project

A.1 Demo code

```
class Project:
    def __init__(self, author: str, title: str, part: str, words:
   int, candidate_number: int):
        part = part.upper()

        if part not in ["B", "C"]:
            raise ValueError

        if (part == "B" and words > 5_000) or words > 10_000:
            raise ValueError

        self.__part = part.upper()
        self.__author = author
        self.__title = title
        self.__word_count = words
        self.__candidate_number = candidate_number
```

A.2 Matplotlib example

```
from matplotlib import pyplot as plt
from math import tanh

x_points = list(range(-20, 20))
y_points = [tanh(x/10)+1.1 for x in x_points]

plt.xlabel("StackOverflow questions viewed")
plt.ylabel("Programming knowledge")

plt.title("The effect of StackOverflow on programming knowledge")
```

```
11
12 plt.plot(x_points, y_points, linewidth=2)
13
14 plt.xticks([])
15 plt.yticks([])
16
17 plt.show()
```

References

[1] N. Stevenson, *Ned-Stevenson/OxCSProject*, Jun. 11, 2024. [Online]. Available: https://github.com/Ned-Stevenson/OxCSProject (visited on 07/04/2024).

[2] "Overleaf, Online LaTeX Editor." (), [Online]. Available: https://www.overleaf.com (visited on 07/04/2024).

[3] "TeX Live - TeX Users Group." (), [Online]. Available: https://tug.org/texlive/ (visited on 07/04/2024).

[4] "LaTeX Workshop - Visual Studio Marketplace." (), [Online]. Available: https://marketplace.visualstudio.com/items?itemName=James-Yu.latex-workshop (visited on 09/13/2024).

[5] C. Hughes, *Cmhughes/latexindent.pl*, Sep. 13, 2024. [Online]. Available: https://github.com/cmhughes/latexindent.pl (visited on 09/13/2024).

[6] "Matplotlib — Visualization with Python." (), [Online]. Available: https://matplotlib.org/ (visited on 09/21/2024).

[7] "Zotero | Your personal research assistant." (), [Online]. Available: https://www.zotero.org/ (visited on 09/21/2024).

[8] "Zotero | Connectors." (), [Online]. Available: https://www.zotero.org/download/connectors (visited on 09/21/2024).

[9] "Better BibTeX for Zotero," Better BibTeX for Zotero. (), [Online]. Available: https://retorque.re/zotero-better-bibtex/index.html (visited on 09/21/2024).

[10] "Draw.io." (), [Online]. Available: https://www.drawio.com/ (visited on 09/21/2024).