Your Project Title

FORMAN CHRISTIAN COLLEGE
(A Chartered University)
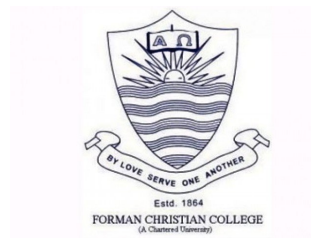
Jane Smith   John Doe   John Snow

12-34567   34-56789   56-78901

Submitted in partial satisfaction of the requirements for the degree of

BSc (Hons) Computer Science

Department of Computer Science

Forman Christian College (A Chartered University)

Primary Advisor: Prof. ABCD

2021

A blank page

This is an optional page environment you could use for things like:

- Your own custom preamble chapters (use `\chapterTitle` for titles!)

- *"This work is dedicated to..."*

- A copyright notice

- Additional notes

- Quotes

- List of publications

- An actual blank page

- Nomenclature / glossaries, etc.

It is not recommended to use this in your undergraduate thesis for submission. It has been left in here to make you aware of its existence. This is largely because it does not follow the guidelines set out for undergraduate theses, but you could include this environment for your own personal printed copy. Consult with your supervisor to see if you can use it.

In addition, you can pass an optional parameter to this environment with the value `c` to centre this text vertically on the page (use the `center` environment to align horizontally).

Acknowledgements

Firstly, I want to thank somebody, and somebody else.[1] Here is another note.

Abstract

This is where your abstract will go. Usually this is written last, after writing the entirety of your thesis.

<Despite the fact that an abstract is quite brief, it must do almost as much work as the multi-page paper that follows it. This means that it should in most cases include the following sections. Each section is typically a single sentence, although there is room for creativity. In particular, the parts may be merged or spread among a set of sentences. Use the following as a checklist for your abstract but do not write any headings in this abstract:

Motivation: Why do we care about the problem and the results? If the problem isn't obviously "interesting" it might be better to put motivation first; but if your work is incremental progress on a problem that is widely recognized as important, then it is probably better to put the problem statement first to indicate which piece of the larger problem you are breaking off to work on. This section should include the importance of your work, the difficulty of the area, and the impact it might have if successful. Problem statement: What problem are you trying to solve? What is the scope of your work (a generalized approach, or for a specific situation)? Be careful not to use too much jargon. In some cases it is appropriate to put the problem statement before the motivation, but usually this only works if most readers already understand why the problem is important. Approach: How did you go about solving or making progress on the problem? Did you use simulation, analytic models, prototype construction, or analysis of field data for an actual product? What was the extent of your work (did you look at one application program or a hundred programs in twenty different programming languages?) What important variables

did you control, ignore, or measure? Results: What's the answer? Specifically, most good computer papers conclude that something is so many percent faster, cheaper, smaller, or otherwise better than something else. Put the result there, in numbers. Avoid vague, hand-waving results such as "very", "small", or "significant." If you must be vague, you are only given license to do so when you can talk about orders-of-magnitude improvement. There is a tension here in that you should not provide numbers that can be easily misinterpreted, but on the other hand you don't have room for all the caveats. Conclusions: What are the implications of your answer? Is it going to change the world (unlikely), be a significant "win", be a nice hack, or simply serve as a road sign indicating that this path is a waste of time (all of the previous results are useful). Are your results general, potentially generalizable or specific to a particular case? >

Table of Contents

List of Figures

List of Tables

Chapter 1

How to use this template

Thank you for using the Official FCCU Computer Science Undergraduate Thesis template! This template was originally authored by Benjamin Williams of University of Lincoln. If you have any troubles, requests, or issues please get in touch with `mumtazsheikh@fccollege.edu.pk`.

1.1 Cautionary Note

It is worth noting that:

1. This template *should* provide *all* of the required formatting for your under-graduate thesis, as required by the module;

2. Provides auxillary environments for typesetting ease – such as ludography and in-built features of LaTeX, and;

3. Formats references automatically in accordance with Harvard referencing, a requirement for the FCCU's DCS undergraduate dissertations.

This is your introductory chapter. The template is pre-formatted in accordance to the Department of Computer Science's undergraduate thesis requirements. For example, this is double-spaced with the correct margin values and header formats.

1.2 Template Structure

Over on the left of this screen (assuming you're in Overleaf), you should see the file structure of this template. There are a few folders and files which are important. In this template, the `chapters` folder is where all the chapters throughout the document are located. This includes other sections, such as the abstract, acknowledgements and appendices.

By contrast, the `preamble` folder contains files which are used before the document is rendered. Think of this like the `head` tag in HTML: the files in this folder provide important meta-data (such as your name, student number etc) prior to the document being rendered. Your first task should be to modify this template, by opening up `preamble/details.tex` and inserting your own details. If you wish to use a package in your document, you can easily add it to the `preamble/packages.tex` file, and it will be imported. Furthermore, the `preamble/bib-setup.tex` file is used to import references and set-up BibLaTeX, but that is covered later in this document.

1.3 Table of Tables (ToT) and Table of Figures (ToF)

At the beginning of this document, you probably noticed these two sections which were included before this chapter, and after the abstract. These are the table of figures and the table of tables. Lets include an image, and notice how it appears in the table of figures above as Figure 1.1.

Figure 1.1: FCCU's logo, used in the title page of this thesis.

In addition, here is an example of a table. Whilst not strictly required, it is worth

noting that Table 1.1 (on the next page) is formatted in accordance to normal usage in scientific articles. If you wish to create tables, it's probably best to use a LaTeX tables generator, like `http://tablesgenerator.com/`.

Table 1.1: A list of house plants and their details.

Name	Type	Location	Size
Billy	Peace Lily	Living room, on bookshelf	Massive!
Jezza	Calathea	Living room, on fireplace	Fairly big
Juno	Laurel Bush	Kitchen Window	Fairly big
Jade	Jade Plant	Bathroom Window	Mediumish
Bernard II	Ivy	Kitchen Window	Big
Shelley	Geranium	Bedroom Window	Small
Benth	Kalanchoe	Desktop	Big

1.4   Typesetting Math

As computer scientists, we often need to describe things with mathematical symbols or operators. Luckily, mathematical typesetting is part and parcel of LaTeX, with many solutions to this problem. This template uses the `amsmath` package for mathematical typesetting, alongside the `amssymb` package. For example, you can use in-line math formatting (with single dollars) like this:

For each $\mathbf{p} \in P$, the weight $w = (\mathbf{p}_0 \cdot \mathbf{p}_2) - \mathbf{p}_1$ was calculated. This linear operation had a time complexity of $\mathcal{O}(n)$.

Double dollars will centre your math without an equation number, like so:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i x \xi} d\xi$$

And finally the `align` environment will allow to reference and align equations, like in Equation 1.1 and 1.2 below. Notice how both = symbols are aligned horizontally.

$$a = b + 1 \tag{1.1}$$

$$\frac{\hbar^2}{2m} \nabla^2 \Psi + V(\mathbf{r}) \Psi = -i\hbar \frac{\partial \Psi}{\partial t} \tag{1.2}$$

But what if you want to align two or more equations but have no equation numbers? Well, thats a job for the `align*` environment:

$$
\begin{aligned}
d(\mathbf{a}, \mathbf{b}) &= \sqrt{(a_0 - b_0)^2 + (a_1 - b_1)^2 + ... + (a_n - b_n)^2} \\
&= \sqrt{\sum_{i=0} n(a_i - b_i)^2} \\
&= ||\mathbf{a} - \mathbf{b}|| \\
&= \sqrt{(\mathbf{a} - \mathbf{b}) \cdot (\mathbf{a} - \mathbf{b})}
\end{aligned}
$$

1.5 Hmm, the margins seem big? Especially on the left of the page?

Yeah, that's okay. This is done by design as your thesis will be printed and bound on one side, so you need extra room on the left so the text doesn't fall into the bind on your page.

The margins used in this template are the standards for postgraduate research theses, which are also applicable for undergraduate theses too. The reason for the large margins and big spacing is so those marking your work can write in notes and red-pen your thesis easily. So yeah, don't worry about it!

Chapter 2

Introduction

2.1 Background

<You are required to write a brief introduction of your project and provide background. In this section you should provide the context and initial knowledge of the domain. You should also highlight the significance of problem and provide motivation behind the work being done.

This complete section should be an improved version of the introduction provided in previous phases of the project. It is expected that there will be complete details available at this stage to provide in this section>

2.2 Objectives

<Objectives are the final results that were to be achieved after the completion of your project. List all of them here.>

2.3 Problem Statement

<Provide a short description of the software being specified. State the problem solved. Convert your language from future to past/present since the project is now complete. Also state whether your system is a program that applies certain algorithms to some application, a tool that is end product of a research, a dataset,

a simulator developed as a result of research, an animated movie developed using graphics techniques, a system that achieves better performance than its competitors, or a software package that is useful in certain application etc.>

2.4   Scope

< Describe the scope of the product that is covered.>

Chapter 3

Requirements Analysis

Another cool thing about LaTeX is its referencing system. This template is set up to use harvard-style referencing. You can do this by using `\citep{citekey}`. It will print out something like this: (Aad et al., 2012). Or alternatively, you can use `\cite{citekey}` to cite things like this: Chatrchyan et al., 2012. This template uses BibLaTeX for referencing, with a Biber backend. This is primarily due to the extensive features BibLaTeX provides, along with the option of glossaries. If you want to customise the referencing style, you can either modify the template slightly to use different options, or use `\usepackage` again to reimport it. There's probably some commands to change its options after its been imported too.

3.1   Ludography

This thesis template also contains an optional ludography. This is primarily for Games Development students, who wish to cite games in their thesis. To use this, just put references into your bib file as usual with the game's details. Then, make sure `keywords` is set to `{game}`. This is what is used to determine which references are games, and which are actual papers. For a more elaborate example, see `bib/ludography.bib`.

Also, make sure that the `title` key is actually the author of the game, and the `author` is the title of the game. The reason this is swapped around is because BibLaTeX likes to print references out with the author first. Then, just add `\printLudography`

with an optional title argument to print out all citations like `\printLudography` or `\printLudography[Games]`.

You can also use the `ludography` environment if you wish to print out some text before the list of games is printed. An example of this can be seen in `main.tex`. To cite games, you can `\cite` it like any other reference. However, if you want it to display the title instead of the standard referencing style, you can use `\citeGame` instead.

Here is an example of a cited game with a normal reference style: Space Invaders, 1978. Ugh, pretty ugly. Instead, here the two are cited in the next sentence as games with `\citeGame`. Both *Space Invaders* and *Breakout* were games made by Atari. Much better!

3.2   Literature Review

<Provide an overview to the projects background knowledge without too much in detail (stick to the scope of the project). The background can refer to previous work referenced from journals, articles, newspapers, or any academic literature providing evidence that the proposed problem is significant and real problem worth solving. If available, provide closely related work done within the project scope and the challenges or defects identified which can be considered as part of the new solution. Describe why you worked on this project in light of the literature review?>

3.3   User Classes and Characteristics

<Describe the various user classes that you have identified. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy.>

3.4   Design and Implementation Constraints

<Describe any items or issues that limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

3.5   Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that affect the requirements stated in the document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3.6   Functional Requirements

<All functional requirements are expressed as use-cases. Fill out the following template for each use-case. Don't really say "Use-Case 1." State the use-case name in just a few words e.g. "Withdraw Cash from ATM". A use-case may have multiple alternate courses of action.>

Table 3.1: UC-1

Identifier	UC-1	
Purpose	...	
Priority	<Choose one from {High, Medium, Low}>	
Pre-conditions	...	
Post-conditions	...	
Typical Course of Action		
S#	Actor Action	System Response
1		
2		
3		
...		
Alternate Course of Action		
1		
2		
3		
...		

3.6.1 Name of Use-Case 1

3.6.2 Name of Use-Case 2

3.7 Use Case Diagram

<Provide the use case diagram>

3.8 Nonfunctional Requirements

3.8.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

3.8.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

3.8.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

3.8.4 Additional Software Quality Attributes

<Specify any additional quality characteristics for the product that are important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

3.9 Other Requirements

<Define any other requirements not covered elsewhere in the document. These might include database requirements, external (hardware, software, or communication) interface requirements, internationalization requirements, legal requirements, and reuse objectives for the project.>

Chapter 4

System Design

4.1 Application and Data Architecture

< Complete logical or physical model is expected. Diagrams/Tools that may be provided in this section include Component Diagram, ER Diagram, Class Diagram (with complete inheritance, composition, and association details), Activity Diagram, Decision Table etc. All these diagrams should have more details than the details provided in Phase 1 of the SDP. Research based projects may provide complete design of the proposed system. Describe each diagram briefly.>

4.2 Component Interactions and Collaborations

<Provide interactions and collaborations between your system components/processing units. Diagrams/Tools that may be provided in this section include Design Level Sequence Diagram, Collaboration Diagram, Event Traces, Detailed DFD, Activity Diagram etc. All these diagrams should have more details than the details provided in Phase 1 of the SDP. Describe each diagram briefly.>

4.3 System Architecture

<Provide the technical architecture of your system. Include a high-level architecture diagram that highlights major subsystems and components.>

4.4   Architecture Evaluation

<Describe the reason(s) behind the decision related to selection of infrastructure/technology used. Discuss pros and cons of the selected technology/infrastructure and the alternative technology/infrastructure. For example why have your group selected a particular API from a list of APIs? Why have your group preferred a particular framework over an alternative framework.>

4.5   Component-External Entities Interface

< Express the communication between components of your system and components of other systems such as NADRA system, bank, credit card verification system, third party multi game server etc. Use appropriate diagram to show the interaction in a better manner.>

4.6   Screenshots/Prototype

4.6.1   Workflow

<Describe complete workflow of your system. Swim-lane diagram may be used. This section should be an improved version of the section presented in previous phases>

4.6.2   Screens

< Include all screenshots of your complete software application's graphical user interface.>

4.7   Other Design Details

<Describe all design details not covered in previous sections. Add subsections as required. There can be details regarding Research Oriented, Game Oriented, or Hardware based projects that have not been covered in this document before, those

details can be provided in this section. For example research based projects may use this section to present their results and analysis; hardware based projects may use this section to describe interface dependencies and issues etc.>

Chapter 5

Test Specification and Results

5.1 Test Case Specification

< Fill out the following template for each test case, also add any additional test cases that were not part of Phase 3 or 4 document. Provide separate tables for input data with each test case if applicable. Research based projects may need to replace this test specification with their own test mechanism.>

Table 5.1: TC-1

Identifier	TC-1
Related requirement(s)	<Include use-case identifier(s) for functional requirement(s) and document section/sub-section number(s) for other requirement(s).>
Short description	...
Pre-condition(s)	...
Input data	...
Detailed steps	...
Expected result(s)	...
Post-condition(s)	...
Actual result(s)	
Test Case Result	

5.2 Summary of Test Results

<Provide in tabular form the defects found in each of your software modules. For example see Table 6.2 below.>

Table 5.2: Summary of all test results

Module Name	Test cases run	No of defects found	No of defects corrected so far	No of defects still need to be corrected
Module 1 (e.g. Bill calculation module or speech processing unit)	TC1, TC2, ..			
Module 2	...			
	...			
Complete System	<Sum all of the above>	<Sum all of the above>	<Sum all of the above>	<Sum all of the above>

Chapter 6

Conclusion and Future Work

6.1   Project summary

<Include a brief summary of how the proposed solution is going to/has addressed the problem statement specified in the introduction section. Provide an overview of what kind of evaluations were undertaken in order to prove that the solution really solves the problem with evidence on results findings.>

6.2   Problems faced and lessons learned

<Provide the details of problems faced during one year of tenure to complete the project. Problems can be technical, financial and motivational. List down all the lesson learned. >

6.3   Future work

<Provide an overview of the recommendations and Include a future directions which is required as part of the future work.>

References

Aad, Georges et al. (2012). 'Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC'. In: *Physics Letters B* 716.1, pp. 1–29 (cit. on p. 7).

Chatrchyan, Serguei et al. (2012). 'Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC'. In: *Physics Letters B* 716.1, pp. 30–61 (cit. on p. 7).

Ludography

Breakout (1976). *Atari* (cit. on p. 8).

Space Invaders (1978). *Atari* (cit. on p. 8).

Appendix A
Some random python code

This template includes the `minted` package, which allows you to import code and syntax highlight it. For example, the text below is imported directly from the `code/test.py` file using the `\inputminted` command:

```python
str = "The times table";
print(str);
print(len(str) * "=");

for i in range(1, 13):
  for j in range(1, 13):
    print("%2d x %2d = %3d" % (i, j, i * j));
  print("-" * 13);
```

And here is a snippet of Python with the `minted` environment:

```python
# Why don't you try running this?
# See what it does? hm?

m = [ 2, 3, 0, 1, 4 ];
x = [ 'rmdq', 'd', 'n`slk', 'odftp`v)', 'hdk' ];

c = ''.join(list(map(lambda y: chr(ord(y) ^ 5).upper() + ' ' if y
↪   != ' ' else '  ', ' '.join([ x[m[i]] for i, v in enumerate(x)
↪   ]))));

print('%s\r\n%s\r\n%s\r\n' % ('=' * len(c), c, '=' * len(c)));
```

Minted supports many, many languages – so you're not just limited to Python. For example, here's some random C++ code.

```cpp
void CTimesTable::Print(const int number, const int upTo) const
{
    for(int i = 1; i <= upTo; i++)
        printf("%d x %d = %d\r\n", number, i, number * i);
}
```

Appendix B
Glossary

<Define all the terms necessary to properly interpret the document, including acronyms and abbreviations.>

Appendix C
Deployment/Installation Guide

<Provide a list of instructions such that users of your system can deploy and install
your system on their own>

Appendix D
User Manual

<Provide a manual such that users of your system can use your system after install-ation. In business software applications, where groups of users have access to only a sub-set of the application's full functionality, a user guide may be prepared for each group. There should be step by step instruction for each user class.>

Appendix E
Student Information Sheet

Roll No	Name	Email Address (FC College)	Frequently Checked Email	Personal Cell

Appendix F
Plagiarism Free Certificate

This is to certify that, I am *Your Name Here* S/D/o *Father's Name Here*, group leader of FYP under registration no *Your Registration No* at Computer Science Department, Forman Christian College (A Chartered University), Lahore. I declare that my Final year project report is checked by my supervisor and the similarity index is X% that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix F. To the best of my knowledge and belief, the report contains no material previously published or written by another person except where due reference is made in the report itself.

Date: *Date Here* Name of Group Leader: *Name Here*

Signature:

Name of Supervisor: *Supervisor Name Here*

Designation: *Designation here*

Signature:

Co-Supervisor (if any): *Co-supervisor name here*

Designation: *Designation here*

Signature:

Senior Project Management Committee Representative:

Signature:

Appendix G
Plagiarism Report