MSc Final Year Project
Generic Briefing
Project Report

Overview

This briefing is regarding **generic skills** and information that apply to the programs mentioned below.

- MSc Computer Science
- MSc Information Technology
- MSc Advanced Computing Technologies
- MSc Data Science
- MSc Information Systems and Management
- MSc Computing for the Financial Services

A separate part of the briefing will be dedicated to program specific information

Overview

This presentation covers

- Plagiarism – quick reminder
- Project report advice
- Project report Common Pitfalls

Questions and Answers During this Briefing

- You can ask questions during and after the briefing.

During the briefing:

- Please refrain from asking questions that are not directly related to the topic discussed.
  - If you have such questions please remember to ask them once the presentation part of the briefing is over.
  - I apologise in advance that I will not answer out of context questions during the presentation part of the briefing.

Plagiarism

Plagiarism

- Unfortunately, there have been a few cases of plagiarism recently here at the Department
- This is an important topic, as the penalty for plagiarizing work can be very severe
- Plagiarism is using words and ideas from another text without proper acknowledgement
- The College's **PLAGIARISM GUIDELINES**

(http://www.bbk.ac.uk/student-services/exams/plagiarism-guidelines/) and penalties

Copying and collusion

- Copying the whole, or substantial parts of information (e.g. text, images) from a source (e.g. journal article, book, website, or any other source, both hard copy and online material) without proper acknowledgement. Proper acknowledgement includes the use of quotation marks around copied text, an in-text citation and a reference.
- This also includes copying from your peers (even with their permission), and submitting it as your own work. Unless otherwise advised, all submitted assessments should be your work alone.

Purchasing ready-made assessments

- Paying someone else to do your assessments is also considered plagiarism, and is a very serious assessment offence.

Self-plagiarism

- A special case of plagiarism is submitting work you previously submitted for another assignment. This is also considered an assessment offence, as you have already received feedback on that work (unlike your peers).
- This includes assessments you have submitted in previous years, on different modules, or in previous universities. Birkbeck uses Turnitin (used by the majority of UK universities) to check, which usually detects these types of plagiarism.
- If you want to rework a paper for an assignment, ask your lecturer whether this is acceptable, and if yes, whether and how you should acknowledge your reworking.

Inadequate paraphrasing

- This occurs when only small changes relating to word or phrase order and word replacement are made, but the overall structure of the text is maintained. These types of superficial changes are not enough to avoid plagiarising.
- To avoid plagiarism, you must fully rewrite the source text in your own words.

Avoid plagiarism

When in doubt, ask your supervisor

See

http://www.bbk.ac.uk/student-services/exams/plagiarism-guidelines

for College guides and policies on plagiarism.

From experience, students do not have initial intentions to plagiarise.

- This usually happens, when corners are cut because of time management issues.
- If you are stuck, please contact the your supervisor and do not wait too long.
- The supervisor is there to help you, and being stuck is definitely a good reason to ask for help.

If you ran out of time it is **better to defer than to be caught plagiarising!**

(it is amazing what Turnitin catches, and PDF images with loads of text, raise suspicions).

Avoid false positive

The project and the report are your proprietary. You can do what you want with them.

However, we recommend that

You do not do one of the following, since otherwise you may find yourself explaining why you did not plagiarise.

- **Put any of your work in the public domain**
- **Ask someone else to check your work on Turnitin**

The

Project Proposal remains

Marking and Feedback

- The criteria can be found on the project form, that should be available on your project Moodle Module.
- You proposal and project are marked by your supervisor, a second marker and possibly an external marker.
- **The identity of the second marker is not provided** (***please refrain from asking for it***)
- The marks and feedback take a while since they need to go through the exam board. This should not stop you from starting to work on your project. Your supervisor may provide you at least with part of the feedback.
- The administrators know best when the feedback will be available

- Not receiving the feedback is not a reason not to start working on your project.

Project Report Advice

Title Page

Title page

Must Contain

- The title of your project
- Your name
- MSc **YOUR PROGRAMME** project report, Department of Computer Science and Information Systems, Birkbeck College, University of London, YEAR
- This report is substantially the result of my own work, expressed in my own words, except where explicitly indicated in the text. I have read and understood the sections on plagiarism in the Programme Handbook and the College web site. I give my permission for it to be submitted to the JISC Plagiarism Detection Service.
- The report may be freely copied and distributed provided the source is explicitly acknowledged.

Title page example

Shoot-em-up Video Game

Oded Lachish

MSc **CS** project report

Department of Computer Science and Information Systems,

Birkbeck College, University of London

2021

This report is substantially the result of my own work, expressed in my own words, except where explicitly indicated in the text. I have read and understood the sections on plagiarism in the Programme Handbook and the College web site. I give my permission for it to be submitted to the JISC Plagiarism Detection Service.

Abstract

The abstract should come after the title page and before anything else.

The abstract should be a should high level summary of your project (1-2 paragraphs).

- This is **not the place** for:
  - ~~Technical details~~
  - ~~Report structure~~
  - ~~Long explanation of motivation~~
  - ~~Discussions about tools programming languages~~

Where to Start

Where to start?

You need a good structure!

You need a good plan for what you are going to write!

Initial Structure Template

1. Abstract
2. Introduction
3. Background (may be a subsection in the introduction)
4. Overall results and "Project Trailer"
5. Main features/requirements and process for obtaining them (in case of system implementation project)
6. Software architecture/data analysis flow
7. Testing
8. Methodology and Schedule
9. Tools and technologies
10. Summary and Conclusions
11. References
12. Appendix: User Manual, Full listing of code, Elaborate requirements list (anything that should not be in the main body of the report)

Role of the Introduction

- The goal of the introduction is to put the reader into context.

- Do not use the introduction for describing your project, that is the role of the rest of the report!

Initial Structure Template

- You can be inspired by available reports.

If you do this, then do so with care:

- All the reports were submitted by students and as such have flaws.

(We are not permitted to provide you with any information regarding the grade and marking of individual report, neither directly of indirectly, please refrain from asking for such information).

- Some of the reports have shortcuts in terms of effort required, using such shortcuts is not advised. Once you have a structure for the project discuss it with your supervisor, they will know if you used such a shortcut.

Adjusting the Structure of the Project

- You should avoid situations where the reader of your report needs to jump forward in order to understand something.
- The length and detail of each section should be proportional to its importance.
- Try to arrange the sections from most interesting to least interesting.
  - This is one of the reasons you want the project trailer as early as possible.
  - Example: if for some reason your testing is very interesting, try to push it forward.

Titles

- Make sure that your titles fit their content.
- Make sure that you understand what your titles mean.

- It is **not a good idea** to have a generic title under which you put all the information that does not fit other title, for example,
  - "~~Implementation~~" a section where you discuss, coding,because that was part of the implementation, design patterns, because that is something you implemented, the tools you used for the implementation and the forms your friend filled to help you choose what to implement.

Presentation Techniques

- Top-down

  Usually the proper technique for project specific information, for example,

  - in the introduction: start from the context of project and work your way towards the details of the project

- Bottom-up
  - Usually the proper technique for technical information, for example,
    - when describing the programming language chosen: start with the choice, then explain why and finally mention other options
- Linear Story
  - Usually the proper technique for the "project trailer“ describe how the software you developed is used in a linear manner.

Presentation technique

When discussing your choice of programming language, try this structure

1. What you chose
2. Why you chose it
3. What were the other options

Important:

- part (3) should not be longer than part (2). Part (3) is the least relevant and should not be too long (this is almost always true for anything under the umbrella of “what were the other options”)
- Write in context, do not add the history of python or a long list of important features that you do not use. Focus on what was relevant to your project.

What, why, what were…

1. What you chose
2. Why you chose this
3. What were the other options

This structure works well in many places.

In general, you should not mix these three elements.

Context

Do not write out of context:

- A long stand alone description of a design pattern
- A long stand alone description of the Agile methodology etc.
- The history of the French language for an app for teaching French
- The life of Django

Make sure you are in context

- Try as much as you can to explain how the design pattern fits your project and why its features are important to your project

Redundant text

Avoid using text that adds no extra useful information.

This can be generic well known information.

- “Writing code is not an easy task”
- “There are slides in this presentation”

This can be a sentence that changes context by changing a single word

- “The goal of this **briefing** is to support the students”
- “The goal of this **class** is to support the students”
- “The goal of this **software** is to support the students”

Word count

Forget about word count until you know exactly what needs to be in the report!

- Writing a lot of text just to feel you have done work may have side effects:
  - you may feel that you have done significant work when you did not (the work was low quality at best)
  - You may have problems removing this work if it is not relevant
  - You might keep it despite the fact that it is not good/relevant and requires significant maintenance
  - You might neglect other important parts of your document

A supervisor can almost always tell when you wrote text for word count, if you are lucky the will notify you.

Jargon

- At this stage you are a student.

This means that you cannot get away by just using unexplained professional jargon.

In general, you need to explain every technical issue, your goal is to demonstrate that you know what you are writing about.

Working on the Report and the rest

Working on the Project

- Try to update you supervisor on agreed times.
  - If possible, do not do a lot of work before updating your supervisor about this
- When contacting your supervisor be concise!
- It is alright to remind the supervisor that you sent an e-mail after **3** working days (staff members get a lot of e-mails and sometimes may miss an important one)

- Try to dedicate a few hours every week for the report
- Do not do any writing marathons unless you really have to

Working on your project: Project Report

- Start to work on the project report as soon as possible. If possible, even before you start coding.
- Use an horizontal approach
  - Start with the structure of the report.
  - Proceed with the structure of each chapter etc.
  - At the paragraph level, write what you plan to write in the paragraph.
  - Write the actual text when it is clear if it is required and where.
- Remember
  - Writing actual text may require significant effort. So, doing so prematurely may result in a reluctance to remove or move redundant or misplaced text.
  - Meta-text is easier to move around.
  - Parts of the introduction and background may require early writing, since in general writing them is far from easy.

Working on your project: Coding/data

- Start as soon as possible.
- It is suggest to use an incremental approach:
  - Start with the minimal implementation that uses all the required technologies. For example,
    - A GUI with one button and one text box. When the button is pressed something happens.
    - Proceed by involving the data base
  - Once you have all the technologies working together start adding features and refactoring.

Advice:

- If you started to work on the report you can choose which features to add according to what makes your report look better.
- It is better to implement some features two weeks before the deadline, than to have a million features and two weeks to write the project report.

Common Pitfalls

When you are stuck or took a wrong turn

- Not contacting your supervisor when you are stuck
  - Being stuck is probably the time you need your supervisor the most.
  - There is no need to come up with something just to set a meeting, “I am stuck” is a legitimate reason for a meeting.

- Doing a lot of work on your own without prior discussion with your supervisor.
  - If a lot of your work turns out to be irrelevant, then you are wasting your time and may insist on holding on to it despite the irrelevance.

Over preparation

- You have many workable options to choose from, but instead of choosing you spend all your time in search for the best option.
  - You are not actually working on the project.
  - The time you are loosing doing this will have bad implications on your final submission.

(choosing a database, library, algorithm etc.)

When you need to make a choice pick the one you know best or will be supported the best.

Impressionism

- Impressionism - focusing on impressing your supervisor and not on the project
  - You get graded according to the final results: code and reports
  - If you only contact your supervisor after you are sure that everything you have done is perfect, then you may encounter the following problems:
    - You wasted a lot of time on something the supervisor could have easily helped you with
    - You wasted a lot of efforts on something that should not be in the report.

External customer

Trying to satisfy an external customer at the expense of your grades

- This is especially true for work-related projects
- Do not let outside interests interfere with your project
- The guidance for your project should come from your supervisor

Perfectionism

- Do not be too hard on yourself: try to avoid perfecting each and every task
- A `good enough' project is better than the promise of unfinished `perfection'
- Sometimes you just have to get on with it

Slides and Information

You will find all the deadlines and their explanations here:

https://www.dcs.bbk.ac.uk/intranet/index.php/MSc_Student_Projects