MSc Project

MSc Computer Science
MSc Information Technology
MSc Advanced Computing Technologies

2018-2019
Overview

- Introduction and technical details
- Plagiarism
- General advice for writing your reports
- Choosing a subject and obtaining a supervisor
- Literature review
- Writing the report
- Working on your project
Important Contacts

- Project tutor: Oded Lachish
- Program administrator:
- Program directors (CS, IT, ACT)
- Project supervisor
- Intranet pages
  https://www.dcs.bbk.ac.uk/intranet/index.php/Student_Projects
  http://www.dcs.bbk.ac.uk/dcswiki/index.php/MSc_CS_project
  http://www.dcs.bbk.ac.uk/dcswiki/index.php/MSc_IT_project
  http://www.dcs.bbk.ac.uk/dcswiki/index.php/MSc_ACT_project
Project Tutor (PT)

- The project tutor is a lecturer in the department

The project tutors role:
- Project briefings
- Assigning second/third markers
- Dealing with issues arising in the rare case that there are problems with the supervision

Please understand:
- There are more than 200 student projects per year
- This is the reason that all your project specific decisions are done with your supervisor.
- It is not the role of the PT to approve any stage of your project.
- The PT does not set the criteria for marking, you program director does.
- There is a specific process for obtaining a supervisor. The PT’s role in the process is only to give final approval.
Important Contacts

▶ Project tutor: Oded Lachish
▶ Program administrator:
▶ Program directors (CS, IT, ACT)
▶ Project supervisor
▶ Intranet pages
  https://www.dcs.bbk.ac.uk/intranet/index.php/Student_Projects
  http://www.dcs.bbk.ac.uk/dcswiki/index.php/MSc_CS_project
  http://www.dcs.bbk.ac.uk/dcswiki/index.php/MSc_IT_project
  http://www.dcs.bbk.ac.uk/dcswiki/index.php/MSc_ACT_project
Project Aims

- Why do we make you do a project?
- Main aims of the MSc project: offer students the opportunity to
  - plan and execute a major piece of programming work
  - critically present existing approaches, place their own approach in the wider area, and evaluate their contribution
  - gain experience in communicating complex ideas/concepts and approaches/techniques to others by writing a comprehensive, self-contained report
  - develop their portfolio
What it is and what it is not

- The project needs to have a significant programming effort
  - We will discuss this a bit further on
  - Your supervisor is your go to address on the subject
- The project needs to be written well
  - We will discuss this more further on
  - Your supervisor is your go to address on the subject

Please Note
- You are judged on your programming effort and quality of your reports
- The topic is not judged
  - It does not need to be original
  - Even topics that may seem simple and mundane can lead to very interesting projects and reports (your supervisor can help with that)

Conclusion:
- The question "Is this a good topic?" is irrelevant
- The important part is whether you made significant programming effort and wrote good reports
Who owns the code you wrote?

According to the document: "Advice for Students and Researchers" "Who owns an Invention at Birkbeck", it seems that the code written by the student is their intellectual property.

The document is available here:
https://www.dcs.bbk.ac.uk/intranet/index.php/Student_Projects
Organizational Matters

- Two important documents have to be submitted:
  - Project proposal
    - About 2000-3000 words in length, providing essential background research, problem presentation and development plan for carrying out project
  - Project report
    - About 10,000 words in length, explaining what you did in the project: design, implementation, testing and evaluation
Deadlines

- Project Proposal Submission: ??th of April 2019 (cut-off date ??th of April 2019)
- Project Report Submission: 16th of September 2019 (cut-off date 30th of October 2018)
- Remark: if you submit late but before the cut-off date, then your grade will be capped at 50.
- If you have an SSP that enables you to get extra time, you must update the administrator in advance
- More details about these documents in just a moment...
Submission of Proposal

- The proposal must be submitted electronically (for plagiarism detection) via Moodle
- The file name should begin with PROP_ followed by your surname and an initial and the programme (e.g., PROP_SmithJ_CS.doc)
- Plain text, Word, postscript, PDF, HTML, or RTF formats are accepted
- A filled-in project proposal form (can be obtained from the intranet page of the module) must be also submitted
- If you need the installation of particular software on Department machines, please note so on the form
  - If you’re not sure about your software requirements, please contact someone from the systems group
Submission of Report

- You submit *two* copies of the printed report to the programme administrator. When the time becomes relevant there will be a clearly marked pigeon hole near the DCSIS administration office for submission out of office hours.
- The submission date will be recorded, late submissions will result in capping.
- One copy of your project report stays with the Department.
- There is no provision for formal extensions, however,
  - you can submit mitigating circumstances, see: www.bbk.ac.uk/registry/policies/documents/MitCircs.pdf
  - providing sufficient written evidence, you can defer your project to the following year — this has to be done before the submission deadline.
You also submit an electronic copy of the report via Moodle
Plain text, Word, postscript, PDF, HTML, or RTF formats are accepted
The file name should begin with REP_ followed by your surname and an initial and the name of the programme (e.g., REP_SmithJ_CS.doc)
This document will be run past the JISC plagiarism detection service
A running version of the software is demonstrated to the supervisor (and an executable/source code on CD/DVD/USB is turned in with both paper copies of the report, these are the only forms allowed)
Submission in general

▶ Please do not wait until the last moment for submission!
  ▶ If you do so and your connection does not work or your computer crashed it is your responsibility
▶ If you work on your own computer, then it is your responsibility to back up the project (suggestion use cloud based Version Control like Bitbucket)
Assessment

- The proposal and the report are assessed by your supervisor and a second (occasionally third) marker.
- The overall mark for the project is made up like this:
  - 20% for the proposal
  - 80% for the report
The proposal is judged according to the following criteria:

- Background research
- Presentation of the problem
- Plan for developing the solution
- Presentation of the proposal
- Any other aspect (optional)

Some MSc programmes may have additional criteria. Check the programme intranet pages.
The report is judged according to the following criteria:

- Specification and design
- Implementation
- Testing, results, analysis, and critical evaluation
- Presentation of report, documentation
- Any other aspect (optional)

Some MSc programmes may have additional criteria. Check the programme intranet pages.
Feedback

- After the supervisor and the second marker have marked the proposal, you’ll receive their comments on Moodle.
- Moodle will update you when the feedback is available.
- An “official” transcript of your results is sent out in January or February by the College.
- One copy of your report stays with the Department, you can collect the other copy.
Ethical issues
Ethical Issues

▶ According to the Birkbeck, University of London Responsibilities and Procedures for Ethical Review

▶ Ethical requirements arise from an evolving understanding of the rights and duties of human beings. Ethics are broader than law, though the law can both reflect and clarify ethical duties. Birkbeck staff and students are expected to exercise ethical principles of honesty, rigour, transparency, care and respect in relation to all their activities, including the planning and conduct of research.

▶ All activities carried out by Birkbeck staff and students that involve one or more of:
  ▶ intervention or interaction with human participants;
  ▶ the collection and / or study of data derived from human participants;
  ▶ a potential impact on animals or the environment;
  ▶ a potential risk of significant reputational damage to the College
  ▶ requiring an individual to step outside accepted regulatory or legal norms
Ethical Issues

- The bottom line
  - If you have any interaction with human participants you will need to:
    - Tick a dedicated box in your project form; and
    - Discuss it with your supervisor.
  - For extra information see https://www.dcs.bbk.ac.uk/intranet/index.php/Research
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 Policy on Assessment Offences lists examples (http://www.bbk.ac.uk/reg/regs/) and penalties.
Examples

- Plagiarism can take many different forms, the policy mentioned above gives examples:
  - copying a whole or substantial parts of a paper from a source text, without proper acknowledgement
  - paraphrasing of another’s piece of work closely, with minor changes but with the essential meaning maintained
  - piecing together sections of the work of others into a new whole
  - procuring a paper from a company or essay bank
  - submitting another student’s work, with or without that student’s knowledge
  - submitting a paper written by someone else, and passing it off as one’s own

- You should not quote large chunks of text from another source! doing so may raise a Copyright issue.
Trying to Get Away With It

- It is not that easy to pass something off as your own work
  - Your report will be run through a plagiarism detection software
  - Most supervisors have seen plagiarized work before and can identify typical give-aways
  - Markers will also be familiar with the main ideas and publications in certain areas
  - We get alerts if suspicious jobs are posted on the net
However, nobody expects you to reinvent the wheel

You are allowed to use existing work as a basis for your project, provided that

- you reference this work properly
- there is enough of your own work in the project

You should not quote large chunks of text from another source! doing so may raise a Copyright issue.

- in that case it’s better to summarize it in your own words and reference the source
Using Existing Material (2)

- When in doubt, ask your supervisor
- See
  
  http://www.bbk.ac.uk/mybirkbeck/services/facilities/support/plagiarism
  
  for College guides and policies on plagiarism
Supervision and Topic
Choosing a topic and then finding a supervisor

- It is the students responsibility to choose a topic for their project.
- Once the topic is chosen the student is responsible to set meetings with potential supervisors.
- The selected topic is the basis for the meeting.
- Next we give advice on how to do both of these things.
Choosing the topic of your project, Generic Advice

- The projects on the MSc programmes don’t have to be novel ideas.
- The work you do MUST be your original work.
- Copies of selected project reports are available here:
  https://www.dcs.bbk.ac.uk/intranet/index.php/Student_Projects
If you have a problem coming up with your own idea pick

▶ Remember:
  ▶ your idea does not need to be original
▶ You can choose to pick the idea to be
  ▶ any application you like or just encountered,
  ▶ a game you like to play,
  ▶ the idea of a different project
▶ Pick anything that will get you started and enable you to practice the knowledge you’ve gained.
▶ Once you start working you or your supervisor may suggest other options.
▶ It is your responsibility to ensure you have an idea.
When am I ready to speak with a potential supervisor

- Your are ready when
  - Your idea has to seem to you as if there is some reasonable effort in it. Once you meet a potential supervisor, they can be the judge
  - You can write about two non generic paragraphs about your idea

This means:
  - If you have more than one idea, then
Strategy for choosing idea

- What is your goal?

  1. To build your IT confidence.
  2. An impressive CV.
  3. Implement an idea you already have.
1. Building IT Confidence

▶ Suggested approach

1. List the technologies you have been or will be taught (programming languages, data bases etc.)

2. Choose your idea so that you can practice this knowledge.

▶ Remarks

▶ This approach minimizes risk.
▶ The challenge is to demonstrate strong capabilities in the taught material.
2. Impressive CV

▶ Suggested approach

1. List the technologies you want to add to your CV.

2. Choose your idea so that you can practice these technologies.

▶ Remarks

▶ This is a high risk approach that requires a strong background.
▶ When choosing specific technologies ensure their stability and the availability of sufficient support.
▶ Before finalizing your choice, check that you can actually use the chosen technologies.
▶ Take into account a steep learning curve.
3. Pre-existing Idea

- **Check**
  1. Can your idea be implemented?
  2. What are the technologies required for the implementation?
  3. Are these technologies available?
  4. Is it reasonable to assume that you will be able to use these technologies?

- **Remarks**
  - This is a medium risk approach.
  - Find a supervisor to help you answer the above questions.
3. Generic Advice

- Do not be a perfectionist, there isn’t any perfect idea
- If you have numerous ideas, pick one and pitch it, you may mention that you have others
  - pick the one you like best.
  - If you can’t, pick at random.
- What about the rest of the topics? when setting a meeting with a potential supervisor you can also say that you have other options if they think the current is not viable
  - Note that the ability to choose one topic is a good skill to have
- Do not over think, delving into the details is much easier when you have a supervisor to help
Finding a supervisor
Finding a supervisor

» Follow the slides on choosing an idea for your project.
» Pick a member of staff by:
   » having a look at the research interests of staff (published on their web pages);
   » choosing a lecturer whose module you have enjoyed; and
   » any other reason

» The choice does not need to be clear-cut
Setting a meeting and the result

- Contact the potential supervisor by a polite e-mail
  - Say you want to discuss the possibility of supervision
  - Describe the project in about 2-3 paragraphs
  - Mention when you are available for a meeting
- Please understand that some of the staff are ridiculously busy or have already reached their quota (if this happen try someone else)
- Have a meeting
- The result of the meeting should be
  - You have found a supervisor
  - You have a recommendation for a more suitable supervisor
Finding a supervisor

- The process described above may take two to three iterations.
- This may happen either because of the student or because of the staff member.
- Only if it takes more than three iterations contact the Project Tutor.
- **Contacting a staff member that has reached their quota does not count.**
- Please note that during these discussions you may ask any questions you have.
- **The goal of this process is to divide the load of hundreds of students between all staff.**
- In general, every staff member should be able to supervise almost all potential projects.
- If you can’t get the staff member that is an expert, please be aware that they usually will answer topic specific questions.
- You should find a supervisor as soon as possible!
Finding a supervisor

- Once you found a supervisor send an e-mail to the project tutor (with cc to supervisor and postgraduate administrator) to the project tutor. The e-mail should contain the name of the supervisor.
  - Your name
  - The Course you are on (MSc CS, MSc IT, etc)
  - A few words about the topic of your project (you may change it and update later on)
- Wait for approval
- The allocation of supervisors is here:
  http://www.dcs.bbk.ac.uk/r/doc/studentprojects.php
- **Important:** if you haven’t received an explicit approval by the Project Tutor (me), you are not supervised!
- This holds even if the staff member agreed or promised otherwise!
Given permission from the relevant program director, it may be possible to have an external supervisor. In addition to the external supervisor a student **must** have a supervisor that is a staff member. This requires an explicit agreement of the staff member and the project tutor. Please remember that the internal supervisor has the final word.
Please do not

- Send a request for supervision without anything else
- Send a request for supervision with a generic area (I would like to do a project in vision)
Reporting
Writing reports like the project proposal and project report

- This is not an easy task even for experienced writers. So:
  - Start the process of writing as soon as possible
  - The purpose of the early start is to give you time to learn how to prepare and write such reports

- Writing the report three weeks before the deadline (even full time) is not a good idea, because
  - learning to write these documents requires the benefit of retrospect. You won’t have time to write and look at the results a week later
  - Your supervisor will not be able to give you significant feedback
    - reading a whole report takes a while
    - your supervisor does not have to give you feedback on weekends
    - your supervisor does not have to give you feedback more than once a week

- We later on discuss what you should do instead
Aims of Reports

▶ You should demonstrate that

▶ you can explain your project and its background clearly and concisely to third parties who may not have expertise in the specific area of the project
  ▶ you may assume that a reader has a Computer Science background

▶ you can relate your project to the wider context of IT
▶ You must demonstrate that you understand what you are writing about
At this stage we assume you are not an expert in the area, so
- you cannot use terms and descriptions that are obvious only to the experts
  - explain any acronyms you use before you use them
  - explain any standard techniques you use
  - minimise the number of citations you use
- Not following this advice may result in getting a grade that is lower than you expect
Formulae and algorithms

- There are two central ways to describe a formula or an algorithm
  - A high level description of what are the inputs and what are the resulting output, plus general considerations such as running time
  - A formal description, actual formula or algorithm. In this case, everything in the formula/algorithm needs to be defined formally
    - Every variable
    - Input/outputs
    - and any other relevant detail

  You have to explain the formula/algorithm using the definition

- It is fine to use the first option, if you are using the algorithm/formula in a black box manner

- Be careful, if you don’t really understand the formula/algorithm, yet decide to present it anyway, you may omit or over abstract important details.
  - This does the opposite of demonstrating that you understand what you are writing about.
Code in reports

- When should you add code to the body of the report (not appendix)
  - you want to demonstrate an interesting way use used a programming language
  - you want to demonstrate that you know how to use unit testing
- How to present the code
  - The code needs to be clearly explained and put in context
  - You should also explain why you chose to present the code.
    Why is this important to the presentation of the project?
- Common mistakes
  - The code contains just a few standard function calls. This adds nothing to the understanding
  - The code is not explained in context
  - The code is from an external sourced and this is not clearly indicated
Project Proposal
Writing Your Project Proposal

- The purpose of the project proposal is to demonstrate that you have put some thought into choosing your topic and you know what you are talking about.
- It is not a full-fledged project report, but should cover the areas shown on the next slide.
Writing Your Project Proposal(2)

Your project proposal should consist of the following parts:

1. A brief description of the topic and where it fits into the field
2. Literature Review
   - Two or three softwares that are closely related (relevant features, one or two screen shots, technology used, pros cons) 
   - Relevant algorithms
   - Related research
3. A high level description of the proposed project.
   - Main requirements/features and the process for obtaining them
   - May include
     - Sketch of a GUI
     - Algorithms you plan to use
     - The practical problem you would like to solve/ the motivation for your project
4. High level system/software architecture
5. Tools and programming languages
6. Methodology and work plan
Writing Your Project Proposal remarks

- You do not have to have all the answers at this stage
- This is more about how you plan to find/develop the solutions
Writing Style

- Your proposal should be well-structured and written in an understandable way.
- As many of the aspects of good writing are also relevant for the final project report, they are covered further on.
In order to hand in a proper project proposal, you need to do some background research.

Usually this involves having a look at existing systems, ideas, algorithms, and approaches.

The hardest part is getting started.

Once you have identified a couple of sources, they’ll point you to other sources.

Here we’ll discuss how and where to start your search.
Search Engines

- Web search engines like Google are a good way to find lots of online resources.
- However, not all of these resources are reliable, so this can only be a starting point.
- You might also have to try out different combinations of search terms before finding the right terminology.
- There is a special service called Google Scholar for academic texts: http://scholar.google.co.uk
Libraries

- You can also have a look at what’s present in the College library
  - You could look for an (introductory) textbook about the area you are interested in
  - The library also provides online resources (http://www.bbk.ac.uk/lib/)
    - This includes their whole catalog
    - and access to electronic versions of journals
- The College also has subscriptions to libraries of professional bodies, e.g., the ACM Digital Library (http://www.acm.org/)
Other Online Resources

- There are other resources available on-line where professionals discuss certain topics
- Some examples are
  - mailing lists
  - newsgroups
  - discussion forums
Your Supervisor

- Your supervisor
  - might be able to point you to some sources
  - give you some suggestions on what to read first
- Obviously, your supervisor will not do the literature research for you, but can help you in getting started
  - So don’t expect complete reading lists
Some Further Suggestions

- You don’t have to read complete articles or books
  - Have a look at the abstract or summary first
  - Browse the article/book if it seems to be interesting

- Don’t overdo the literature research
  - There is a huge number of publications out there, nobody expects you to read them all (this could take years)
  - At some point you have to decide on what you want to do and write up your proposal
Project Report
Aims of Project Report

- Your project report is not simply a project management report or system documentation
- The main aims of your project report are:
  - To present your project in a meaningful way
  - To demonstrate that you can produce a document written in a well-structured and intelligent way
Why Is This Important?

- At least one of the people marking your project will not have followed your project closely (2nd/3rd marker)
- If you are on the boundary between two different marks, a well-written project report can make a difference
- The external examiners on the exam board only have your project report by which to judge your project
- To get good marks for your project, you need to do both:
  - Produce the software
  - Turn in a good report
- There are projects that have been graded below their potential due to an indifferent or poor write-up
Physical Appearance

- A tidy, well laid-out and consistently formatted document makes for easier reading
  - Use word-processing software
  - Leave margins to allow for binding
  - Use headings for chapters, sections, and subsections consistently

- Quantity does not automatically guarantee quality
- Project reports need to be **concise**, clear and readable
- The assessment is not about page count or word count
Mandatory: Title Page

- The title page should contain
  - (obviously) 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 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.
Table of Contents

- Gives the full headings of all chapters (and the sections within them) with the appropriate page numbers
- Page numbers should be right-margin aligned
Optional Sections

- Acknowledgements: if there are persons who you would like to thank for their support and help
- List of Figures/Tables: if you have used lots of figures and tables
- Abbreviation list: if you have used abbreviations (it’s also a good idea to spell out the meaning the first time you use an abbreviation in the text)
Report Structure

▶ A good document structure takes into account
  ▶ the purpose of the document (i.e., to report on your project)
  ▶ its target readership (i.e., the examiners)
▶ Many reports are too long, too unstructured, and lack purpose
▶ You should aim for
  ▶ continuity: order your material in such a way that a reader is able to follow your descriptions
  ▶ completeness: do not leave out significant parts
▶ Find a balance: cover the important things without overwhelming the reader with unnecessary details
Unlike an essay, a report contains headings and subheadings (to make its structure explicit)
Each subheading may be further divided into subsections or subdivisions
  - Usually it’s a good idea to number each section and subsection
To develop and improve the continuity, it helps to pay special attention to the report’s structure
Starting to work on the report

- Before starting to write, think about the structure of your report (in outline or even just as subheadings)

- Suggested outline:
  - Abstract
  - Introduction
  - Background (may be a subsection in the introduction)
  - Overall results description and "Project Trailer"
  - Main feature/requirements and process for obtaining them
  - Software architecture
  - Testing
  - Description of the development process
  - Tools and technologies
  - Summary and Conclusions
  - References
  - User Manual
  - Appendix: Code
  - Elaborate requirements list

- Remark:
  - The final structure must be adjusted to the project.
Abstract

- The abstract is a brief synopsis of your work, a bit like an executive summary
- It should be no longer than about 250 words
- It’s usually a good idea to write this at the end (when everything else is known)
- Beneath the abstract, put the name of your supervisor
Introduction

▶ Contains a brief outline of the topic as a whole
▶ Then state the aim and objectives of the project
  ▶ What was the purpose of the project and what did it set out to investigate?
▶ At the end of the introduction, provide a road map for the remainder of the report
This chapter should focus on the context that you are operating in, e.g., by describing:

- typical applications
- alternative tools and development approaches and how they have been used in practice
- alternative systems and what they achieve and do not achieve

This should be a synopsis of the relevant part of your project proposal (do not just copy your proposal)

Restrict yourself to what’s relevant to the specific context of your project (the proposal can have a more general look at the state-of-the-art)
Overall results description and "Project Trailer"

- An exhibition of selected features of the software you developed:
  - It is recommended to add screen-shots of interesting stuff.
    - the login page and long menus are usually not interesting.
    - Don’t just add screen shots, annotate them and refer to them
Main feature/requirements and process for obtaining them

- An overall specification of your project
  - Mostly the main features/requirement
  - This is not the place for a long detailed list, you can leave that to the user manual
- A description of the process for obtaining the requirements
Software Architecture

- High level description of the system, accompanied with relevant figures
- n-tier architecture or any other architectural pattern used
- Key parts of each layer
  - how you applied design patterns (how they fit together)
  - It is recommended not to have over-crowded or under-crowded UML diagrams.
- In general, should be written in a top down hierarchical manner
  - System description. Partitioning it into relevant parts 3-4
  - Software architecture. Partitioning it into relevant parts 3-4
  - Description of each of the parts from the previous point
Testing

- Describe the verification process you used in your project
  - Unit Testing, Static Analysis etc. - *give an actual code example* with a short explanation and a link to where tests are stored. It is not sufficient to say you did this
  - System testing, which can include manual testing, Selenium, Cucumber etc. Describe your testing plan and the consequences. It is not sufficient just to tell a story of how your testing was done
Description of the development process

- Software development process (beware of buzzwords like SCRUM, you are either admitting to group work or to a serious overkill)
- Timeline of the project development
Tools and Technologies

- Describe the technologies used in the project, why they were chosen and what were the other options:
  - Tools and programming languages.
  - Data-base, servers etc.
  - In most cases, the description should start from the choice, proceed to the motivation and end with a discussion of other possible choices that could have been made.
Summary and Conclusion

- The final section
  - summarizes the project as a whole
  - A critical evaluation by the student, emphasizing
    - strong points and weak points
    - lessons learnt
    - design decisions which, looking back, would be made differently
    - ways in which the project could be improved or extended
    - etc.
  - recommendations for the project
- You can also describe possible future work in the area of your project
References

- You have to provide a complete list of all the works mentioned in the text.
- For a book, this normally includes the name(s) of the author(s), the title, the publisher and date of publication.
- For an article, it would include the name(s) of the author(s), the title of the article, the name of the journal, the volume/issue number and date and page numbers.
- Examples:
Appendices

- Additional relevant material which did not make it into the main sections should appear in an appendix.
- It can also include lengthy items such as:
  - program code
  - raw data
  - detailed statistical analysis
- If you have very lengthy items, you can include a CD or DVD that contains these items:
  - Usually, there’s no need to print out the complete source code.
Adjusting the Structure of the Project

▶ You should avoid situations where the reader 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.
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.
1. A brief description of the topic and where it fits into the field
   This has to come from two angles; the computing framework angle and the automated framework angle
   For brevity I will start with small introduction on computational finance
   I will write about automated trading within the financial industry
   I will talk about the continuing growth of the automated trading industry
   I will give an introduction about retail trading and the wider public access to the markets. This could then be referenced as a "target audience for the framework"
   I will talk about both the fundamental and technical analysis.
   I will then talk about this in the context of foreign exchange and equities.
   Noting that automated trading is applied in a huge spectrum of markets but focusing on these two.
   I will follow this with an explanation on how floating exchange rates were introduced and give a short background on the London stock exchange (not the history of stock markets)

2. An account of the current work/applied technology in this area
   The three topics I will focus on are: High frequency trading (HFT), MAN Groups AHL fund and direct market access (DMA) I will talk about high frequency trading and the regulation surrounding it...
1. A brief description of the topic and where it fits into the field - in this section I will:
   - Introduce the two topics of computing framework and automated framework
   - Start with small introduction on computational finance
   - Write about automated trading and its role in the financial industry
     - Including retail trading
   - Talk about both the fundamental and technical analysis.
   - Introduce the foreign exchange and equities markets. Noting that automated trading is applied in a huge spectrum of markets but focusing on these two.
   - I will follow this with an explanation on how floating exchange rates were introduced and give a short background on the London stock exchange (not the history of stock markets)

2. An account of the current work/applied technology in this area
   - Focus on three topics: High frequency trading (HFT), MAN Groups AHL fund and direct market access (DMA)
   - High frequency trading and the regulation surrounding it...
1. A brief description of the topic and where it fits into the field
   1.1 Automated Trading
      1.1.1 Its role in the finance industry
      1.1.2 Include the retail industry
      1.1.3 High frequency trading
      1.1.4 Direct market access
      1.1.5 A framework to accommodate the ability to access the markets
   1.2 Fundamental and technical analysis.
      1.2.1 What they are
      1.2.2 How they are used to trade the markets
   1.3 how it has changed with computational finance
   1.4 Foreign exchange and equities markets.
      1.4.1 Foreign exchange moving to a floating exchange rates
      1.4.2 Lightweight introduction on the London stock exchange
      1.4.3 Briefly mention the others markets

2. An account of the current work/applied technology in this area
   2.1 MAN Groups AHL fund
      2.1.1 Its success - starting in 1987
      2.1.2 Recent poor performance
      2.1.3 Research into systematic trading
   2.2 Co-location ...
Using enumerated lists

- The lists help to communicate. It is easy for the supervisor to refer to specific points and easy for you to find them.
- It is really easy to reorder things since the points aren’t glued to together by text (paragraphs are).
- You can proceed this way until you know exactly what text you want to write (about a line per paragraph).
- Once you start writing the actual text, remember to make sure that you added the gluing words between paragraphs, since otherwise it reads badly.
- Unless it really makes sense, don’t make an enumerated lists your submission text!
1. Start with the top level, for example:
   ▶ Dedicated social network project: Start an overview of social networks; Do not start with a long description of the internet.
   ▶ time management application: Start with an overview of time management applications; Do not start with a long description of applications in general
   ▶ Lowest level that is sufficient for explaining the context of the project.

2. Explain at a high level what you plan to implement.

3. Unless really essential, the high level should not include technical details and exact specifications.
Advice for related work

1. Select about three similar application.
2. Describe each application shortly.
3. Elaborate on the application closest to your project. You may want to compare it to what you plan/achieved.

Remarks

▶ If your project is unique then explain this instead of doing the above.
▶ You should not have a long list of similar applications, each with its own description. If you feel you need to have one, then minimize the extra details, ideally have a list with names and references.
Avoiding Common Pitfalls – Style

Be to the point! Be explicit! Be concise!

▶ Starting sentences with ‘In my opinion’, ‘I think’, or ‘When I did’
  ▶ An examiner is not interested in your opinion, but in how you support your argument

▶ Using the phrase ‘It is obvious’
  ▶ What is obvious to you might not be obvious to someone else
  ▶ If it is really obvious, you can explain it in a few words

▶ Broad generalizations
  ▶ ‘All generalizations are untrue’
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

- 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
Common Pitfalls(1)

- Pushing in the wrong direction
  - To avoid this problem, arrange meetings with your supervisor and show up on time.

- Getting stuck
  - If you are stuck for any reason (and have no meeting scheduled), let your supervisor know immediately.

- Your supervisor can do nothing for you if they are unaware of your situation.
Common Pitfalls(2)

- 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
Common Pitfalls(3)

- Over-/under-ambition
  - Try to be realistic what you can achieve, a good project requires a lot of effort
  - However, it is better to do a smaller job well than it is to fail to do a big job at all
  - Ask your supervisor for guidance on the scoping of the project
Common Pitfalls (4)

- Failing to plan a fallback position
  - Have a plan B if you are not able to complete the planned work in time
  - Try to plan your project in stages, so you have a complete stage to fall back on
Inadequate literature review

- References should cover the relevant theory and/or technology that you use
- The literature review should demonstrate that you have an understanding of the current state-of-the-art and show how your project fits into it
- Cite your sources properly (more on this in the section on plagiarism)
Common Pitfalls(6)

- Deliverables of unknown quality
  - The work appears sound but there is no evidence of its validity
  - Include summaries of test results
  - Demonstrate an executable to your supervisor
Common Pitfalls(7)

- Deliverables of unknown origin
  - Sometimes a project seems to be of good quality but too extensive to have been done in a few months
  - If work existed before the start of the project, define what existed and who produced it
  - Nobody expects you to reinvent the wheel, but you have to document where pre-existing work comes from (more details in the section on plagiarism)
Common Pitfalls (8)

- No acknowledgment of sources
  - Similar measures are applied to the report: you have to cite your sources
  - For every part of your report it should be clear
    - if this was the result of your own work
    - or if it has an external source
Common Pitfalls(9)

- Perfectionism
  - Don’t 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
Choiceism - Focusing on what to do instead of doing, for example:
  - Dedicating a very long time to choosing a database, programming language, algorithm etc.
  - If you are not sure, then pick the safest choice.
Common Pitfalls (11)

- 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’ve 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. You may even insist on adding this to your reports despite your supervisor’s advice and the implications to the final grade

- So if you see you are spending too much time on something to make it perfect check with your supervisor if it makes sense.
Common Pitfalls

▶ Time management
  ▶ Assume that you will lose up to 2 month for various reasons
  ▶ Last three weeks marathon, rarely works
    ▶ Your supervisor won’t be able to give you much feedback
    ▶ You won’t have time to properly go over what you write
    ▶ When reading a report it is rather obvious that it was written in this manner.

▶ Last moment bursts:
  ▶ Many times last minute bursts leave the illusion of doing significant work
  ▶ Usually the result is a number of long unreadable, inconsistent, incoherent paragraphs
  ▶ Unless you are very experienced, it may take longer to fix than to write properly
  ▶ Solution: make a plan and in advance (we already saw an example)
Generic
Assessment of Proposal/Report

▶ Let’s have a closer look at what the examiners will be looking for
▶ For each criterion, we specify
  ▶ what is needed for a **pass** mark
  ▶ what is needed for a **distinction** mark
  ▶ anything going well beyond a pass, but not quite reaching a distinction will be deemed a **merit**
To obtain a pass mark:

- The proposal specifies a suitable problem and discusses its requirements
- It also reviews potential approaches and evaluates them

To obtain a distinction mark:

- A challenging problem is specified and clearly outlined, this includes its context and the technical/user requirements
- The student shows a clear understanding of the researched material
- Potential approaches are reviewed and critically evaluated, highlighting strengths and weaknesses of each
Plan for Developing the Solution

➢ To obtain a pass mark:
  ▸ A suitable development/research method is chosen
  ▸ The project is broken down into manageable chunks

➢ To obtain a distinction mark:
  ▸ An appropriate development/research method is chosen and its suitability is well-justified
  ▸ The project is broken down into subtasks that are logically coherent
  ▸ In the case of unknowns (e.g. open research questions) “fallback” plans are laid out
Presentation (Proposal and Report)

- To obtain a pass mark:
  - The proposal/report are coherent in style and structure
  - They clearly communicated the student’s contribution
To obtain a distinction mark:

- Complex issues are explained clearly and concisely
- The content is well-organized and structured in a way that demonstrates the links between the concepts
- The proposal/report shows that the student clearly understands the researched material
- The solution and any other claims made by the students are well-justified
- The author uses various resources and cites relevant resources using an appropriate consistent referencing style
- The proposal/report is of professional quality and contains very few, ideally no, typographic errors.
Specification and Design

- To obtain a pass mark:
  - Before starting the implementation, a specification and design of the system/software is laid out

- To obtain a distinction mark: The specification and design of the system/software
  - shows a clear understanding of what needs to be done to meet the requirements
  - is well-rounded, i.e., the components fit together in a coherent way
Implementation

- To obtain a pass mark:
  - The key stages of the implementation are explained
  - The implementation is sound.

- To obtain a distinction mark:
  - The key stages of the implementation are clearly explained
  - The implementation is done to a high standard
To obtain a pass mark:

- The report attempts to provide a clear and justified reflection upon the contributions and its limitations
- It discusses how the software meets the specified requirements
- A running version of the software is demonstrated to the supervisor (and an executable/source code on CD/DVD is turned in with the report, USB stick of Optic disk are the only forms allowed for submission of code)
To obtain a distinction mark:

- The solution demonstrates real insight into the problem/research question
- There is a clear and justified reflection upon the contributions and its limitations
- The key results are accurately analyzed and their relevance is explained
- It discusses how the software meets the specified requirements and is shown to be reliable
- A running version of the software is demonstrated (as above)
Overall Assessment

► Work that meets some, but not all, of the criteria for distinction may be considered for a merit:
  ► This may be a respectable, if only partially successful attempt at a challenging project
  ► Or a less ambitious project carried out, and written up, to a high standard
► The examiners grade the project independently and then meet to arrive at an agreed grade
► Students may be called upon to make a presentation of their project to a subcommittee of the exam board to demonstrate their grasp of the material