The intention of the following deadlines is to encourage you to work steadily throughout the year. This timetable does not absolve you from the proper planning of your project. In fact you should have many more deadlines for your group than just the few listed below. You should be able to cope with unexpected developments. Please note that only the dates in bold are departmental dates requiring mandatory hand-in or presentation. All other dates are for internal project use only and are intended as checkpoints for you and your supervisor.

The project evaluation will include:

- a poster (5%)
- a project web page (5%)
- the final presentation (10%)
- comprehensive project report (80%)

Please note that the MOCS minor project will be subject to the same final handin deadline as the major project, but will only count 1/6th of the final mark. The minor project will not have an associated poster or web page --- unless the students wish to make their work available on the web (for no credit) --- and will be evaluated using slightly different criteria. The dates listed below can be more flexibly interpreted by supervisors of MOCS projects since the project is significantly smaller.

*The project proposal document, prototype and poster must be prepared by the entire group. The project report and web page will be individual pieces of work. In addition to the web page for each student, there must be a group web page linking to each of these, which is maintained by the group. These web pages should be thematically consistent.*
1. **Milestones**

1. Provisional Project proposal document to supervisor. 30/4
2. Final proposal due. **4/5**
3. Project presentation. **10-11/5**
4. Report: Definition and Theory / Background Chapter (including literature survey). **25/5**
   - [Study Week 28/5-1/6]
   - [Exams 4-15/6]
   - [June Vac 18/6-27/7]
5. Report: Chapter on Design. 30/7
6. Prototype demonstrations to supervisors and second readers. 1-2/8
7. Weighting chosen for project assessment. **30/8**
   - [Spring Vacation 10-14/9]
8. Evaluation of implementation by the users, write-up of “experiment”. 24/9
9. Report: Chapters on Implementation and Testing. Final implementation (optimised, etc.) completed, testing completed. Outline of complete report: chapter and major section headings with 1-2 sentence descriptions for missing sections. 1/10
10. First draft of report. 15/10
11. Final report handed in. **19/10**
12. Due: poster **23/10**
13. Due: Web page **24 or 25/10 (Open Day)**
14. Project demonstrations to supervisors and second readers. **24 or 25/10 (Open Day)**
   - Study week 29/10-2/11
   - [Exams: 5/11-13/11]
15. Final project presentations. **14-15/11**
2. Research Method Assignments Related to Your Honours Project

Refer to Research Methods in Computer Science - A Summary and the Practical Work Assignments by Edwin Blake (RM Summary)

These assignments are to be done once you have chosen a project. Your final mark will be made up of these assignments. As you will see, you and your project coordinator will together decide on the balance between research and development in your project.

2.1 Preliminary Work

These parts are not for marks but are needed for the remaining assignments. Some of these questions are a kind of checklist you should go through in preparing your project.

1. As a start to the assignments start collecting background information around the subject of the project. Do this in consultation with the project coordinator. As a first step identify the major points covered by textbooks in this area: try the library if you don’t own the books already.
2. Make a list of ten papers that seem relevant to the research project. Arrange the ten papers from that with the most potential to that with the least (as you did in problem 6.1 RM Summary).
3. Find two papers that relate to your project. Work through the papers and compare the contribution they can make to your research project. Attempt to find positive and negative aspects that you find in these papers.

2.2 Checklist

1. Which of the methods from Section 1.1, RM Summary can form the major thrust for your research project? Which other methods will be required to support this major thrust? Describe and justify the structure of your project.
2. Will a model be of use to address some aspect of your project? If so:
   a. Sketch the outlines of a possible model. Do this in two pages or less, diagrams included.
   b. Compare the model you proposed with other models proposed in the literature to address a similar problem.
3. Will you be constructing a prototype as part of your project? If not, why not? If you are:
   a. Why are you? What purpose will it serve? What will you learn from it?
   b. How would you go about constructing it?
   c. Which aspects of a full system may be omitted in the prototype?
   d. How would you describe the prototype in your report to highlight only the relevant aspects?
4. Is there any part of your project where either a new algorithm will be required (or be beneficial) or where inclusion of pieces of code in your report will improve its clarity? Explain. How would you report such algorithm(s) or fragments of algorithms? Will it be necessary to prove any aspects of your algorithm(s)? If so, how would you approach it?
5. Should any part of your project be expressed in mathematical notation? Describe the notation and why it would be useful. Would it be helpful to prove any results in your project formally?
6. Would an experiment help to address some aspect of your project? If an experiment would not be useful for your project, give reasons for this. If it would be, sketch the outlines of a possible experiment. Do this in two pages or less. State whether a single object to study would be sufficient. Would you need a control group? How would you select these groups? What do you hope to prove with the experiment? How would you measure the results?

7. Would a survey be useful to address some aspects of your project? If so, describe its purpose, the population, your hypotheses, selecting the sample (if required), use of questionnaires and/or schedules and your envisaged data collection process.

8. Consider the experiment you designed in question 6, RM Summary and/or the survey you described in question 7, RM Summary. Formulate your hypotheses given there in statistical terms. Also describe how you would test your hypotheses once your data has been collected.

9. Consider your project again. Will a case study be of any use to address part of your problem? Discuss.

10. Outline a possible line of argument for your project. Assume you have observed the phenomena concerned as expected or constructed the required model or algorithm — as applicable.
   a. Briefly outline the main argument that you would use when you report on the results of your project. Do it in the same way as summarized under arguments in Section 1.1, RM Summary, but you may use more steps if you want to give a more detailed account of your reasoning.

11. Consider your project again. Describe how you would structure the main section of the report (see Section 0 below for some formatting and length requirements). List all the chapters and the major section in each chapter. Indicate the length of the chapters bearing in mind the length requirements given.

2.3 Assignments for Evaluation

[1] Proposal [Due 4 May - Marks to be decided still]
This proposal is a team effort and the whole project team will hand in a single proposal. Thus all members must contribute equally to the proposal.

Purpose: To enable the Department to judge the feasibility and acceptable standard of the project as a whole, as well as each student component.

It is essential that each student have a clearly identifiable individual piece of work that constitutes a fair subdivision of the load. The work done by each individual student must be able to stand alone as a Computer Science project for examination purposes.

Clearly formulate your research problem according to the guidelines presented in the lectures. Select the most appropriate research method and the supporting research methods. Write a research proposal for this project.

Decide on the stages of the project and the dependencies between them. Draw up a list of deliverables and milestones. Compile a project plan and draw the corresponding Gantt chart. Use specific dates so that you finish on time for the final hand-in and presentation.

Length: 2000-4000 words (the upper limit must be strictly observed).
**Required Components**

The document must contain the following information:

1. **Project Description**: high-level detail
2. **Related Work**: Preliminary outline of the key works identified (need not be comprehensive, that comes later in Assignment 0). All work must be properly cited.
3. **Outcomes**: Major results, including:
   a. System (software, key features, major design challenges)
   b. Questions tackled (What is being investigated? Why is it important? What methods are you going to use?)
   c. Expected impact of your project (What results do you expect? What difference will they make?)
   d. Key success factors – how will you judge whether the project has succeeded or not
4. **Work Detail**
   a. Risks (e.g., delays in obtaining key resources) and Risk Management Strategies.
   b. Timeline, including Gantt chart.
   c. Resources required (equipment, people, special software etc)
   d. Deliverables
   e. Milestones (which should refer to the Timeline)
   f. Work Allocation to team members

**[2] Background [Due: Due 31 July - Marks to be decided still]**

You are required to write a literature synthesis covering the specific topic of your Honours project. It is intended that this will later serve as the basis for a chapter in your final Honours dissertation. In the unlikely event that a literature synthesis is not appropriate to your particular topic then you may choose another topic but this must first be approved by the research methods lecturer.

In writing your literature synthesis you should pay careful attention to the following issues:

a. **Structure**: A mere paper-by-paper summary of individual research papers will not be positively received. This is after all a “synthesis”, which implies that you have not only read the relevant background literature but arrived at some means of logically organizing it. You should derive an organizing structure or categorisation for the research and fit individual research papers within appropriate categories.

b. **Summary**: Your synthesis should conclude with a comparison of the competing methods. This can also be usefully summarized in a tabular format.

c. **References**: You must use a consistent and correct referencing style. Your use of web-based (on-line) references should be sparing. References to journal articles and conference papers are much preferred.

d. **Technical Writing and Presentation**: You will be marked on both the quality of writing and your document presentation, so be sure to adhere to acceptable technical style and professional formatting.

Required length: 2000-3000 words (including figure captions, table contents and references)

*Each member of the project team has to hand-in their own Literature Synthesis.*
3. Suggested Format for Honours Project Reports
This section only discusses the appearance of the report. For further information refer to your lectures in Research Methods and consult your supervisor.

The total length of the main report should be between 35–50 pages and must not exceed 55 A4 pages (or about 30 000 words). The text of the main report should be single spaced, with a font size of at least 11 pts. Use at least a 2.5 cm margin top and bottom and 3cm on the sides of the pages.

Appendices and other manuals can be 10pt font size. Please use “styles” for formatting if you are using a word processing package (such as MS Word) to ensure consistency (or use LaTeX).

Appendices (such as test results) should be kept short and bound together with the main report. However, user manuals, programmer manuals and bulky data dictionaries should be bound as separate volumes. Please consult your project supervisor if you are unsure which materials you should include in the main report and appendices and the length of appendices.

The report should be clearly written, and should include only relevant material. Indeed including only enough detail shows you have learnt how to distinguish the important issues from the trivial ones.

All Honours project reports must be prepared in the following sequence:

i. Front cover (clear plastic sheet)
ii. Title page
iii. Abstract
iv. Acknowledgment page (if any)
v. Table of contents
vi. List of Figures (and perhaps list of tables)
vii. Main report, this should be numbered from page 1 to a maximum of 55.
viii. References: Please use the ACM style for references.
ix. Appendices
x. Back hard cover

Ring binding should be used for all reports and appendices. You will need to produce two copies to hand in and you will probably want to keep one copy for yourself. In addition you will deposit an electronic copy together with your other project materials on the Department’s publications server.

Title Page
At the top you should have “Honours Project Report”. One third of the way down the page put the title of the project in a larger bold font, your name below that, then “Supervised by” and your supervisor(s) name(s). Towards the bottom of the page put “Department of Computer Science”, “University of Cape Town” and the year. All the text should be centred.

Abstract Page
The next page is titled Abstract and has the abstract of the report of between 200 and 300 words. Keyword and Subject Descriptions should follow immediately after the abstract in the same page, each with not more than five careful selected items. The descriptors should be chosen from the latest version of “The Full Computing Review Classification Scheme” of the ACM Computer Review. Any suitable word that reflects the nature and content of the project may be chosen as a keyword.
Below the abstract please list the documents and other deliverables that comprise the full report (such as manuals and guides).


The following marking scheme will be used to assess all Computer Science Honours projects in 2007. Projects that are cross-disciplinary will also be subject to this scheme, and the various assessment categories have been chosen to allow as broad a range of projects as possible.

Your mark is initially determined by your supervisor and second reader. The staff then meet to discuss all the projects after your final project presentation. That determines the mark that will be given to the external examiner for consideration.

4.1 Instructions to Supervisors and Co-Supervisors

The final project mark is composed of a number of components, as described below. The project mark is out of 100. This mark represents 33% of the entire Honours course mark.

Component I allows many different types of projects to be accommodated. All projects MUST assign 30 points to the optional component. Some possible suggestions for point allocations are provided later in this document. Components II and III have fixed weights which are the same for all projects. Note that these component mark categories may not necessarily map directly onto chapters in the Project Report, but those reading the report be able to assess the student’s work/contribution in each of the listed categories. The categories (and how they are assessed) are described later in this document.

Component I – Optional Project Report Categories (30 points)

No individual category may be assigned more than 15 points.

1. Software Engineering/Analysis
2. Theoretical Analysis
3. Experimental Design
4. Implementation
5. Findings

Component II – Compulsory Project Report Categories (50 points)

Every project report must contain this information. As before, the chapter headings may not map directly to the categories provided below, but the information must be present in the project report.

6. Achievement 15 points
7. Project Assessment 15 points
8. Evaluation 20 points
Component III – Ancillary Project Documents and Deliverables (20 points)
Note: components 9, 10 and Error! Reference source not found. have their own marks schedules, which are not reproduced here.

9. Web pages/presence  5 points
10. Poster  5 points
11. Final presentation  10 points

Allocating points for Component I
The supervisor(s), in consultation with the student, should select any of the categories 1 to 5 inclusive, depending upon two main factors:

i. the general applicability of the category to the work done,
ii. the objectives of the project.

Note that ALL projects must assign **30 points** to these various categories.

a. Please use the statements in each category as a guide to assessing what a student has earned for each category. For example, under normal circumstances, to award the maximum mark for a category all of the statements below the category heading should be true.

b. During the course of the project, supervisors should not normally make promises to students with respect to the marks / classes that they can expect.

c. When the project is in full swing and unlikely to be changed in any major way, the student and supervisor meet to agree on the categories to be used in the assessment.

To accommodate the diverse range of honours projects offered in the department, the supervisor(s) can choose any reasonable set of weights/points in the optional component. The following table provides possible point allocations for 3 quite different kinds of projects.

<table>
<thead>
<tr>
<th></th>
<th>Commercial Project</th>
<th>Experimental Project</th>
<th>Theoretical Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SE/Analysis</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2. Theoretical Analysis</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>3. Experiment Design</td>
<td>0</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>4. Implementation</td>
<td>10</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>5. Findings</td>
<td>5</td>
<td>10</td>
<td>15</td>
</tr>
</tbody>
</table>

Assessing the Quality of Writing
To assess the quality of the writing in the project report, the 70 marks comprising component I and II are multiplied by a factor in the range 0.5-1.0. If the writing is particularly poor and the document incomprehensible, a weighting of 0.5 should be used. If the writing is excellent, then a weighting of 1.0 should be applied. Most well presented reports should score close to 1.0.

Among the elements that should be considered when arriving at a weighting include:

- the succinctness and clarity of the abstract;
- the clarity with which the student introduced their project and explained their objectives;
- the presentation of their arguments/rationale for pursuing their particular solution;
• the conclusion is well structured: it revisits the issues they addressed and recaps their major results/contribution
• the use of sufficient and correct references in an appropriate style (ACM for example).

4.2 Assessment Categories

Software Engineering/Problem Analysis
This category deals with the strategy adopted in analysing the problem at hand.

a. The experiment / system is well justified (i.e. the student shows a knowledge of how the final product is going to fulfil what he expects to achieve).
b. The data structures which have been selected are appropriate for the objects and stores which were identified during analysis.
c. The system is well structured i.e. the various processes in the analysis phase have been grouped in such a way that a well-structured, modular, robust system results.
d. The student used relevant design tools/techniques in an intelligent manner.
e. The student demonstrated originality and ingenuity in approach.
f. The project uses concepts and techniques appropriate to modern computer science. It is readily apparent that the project, particularly a programming project, was not written by a first year.

Theoretical Analysis
This category is intended to evaluate the theoretical analysis for those projects which require such e.g. Proofs, derivation of computational complexity.

a. How sound is the analysis? Is it correct, are the assumptions valid?
b. What is the relative complexity? Is it appropriate for a 4th year project?
c. How novel/useful is the work?
d. Is it well presented/accessible?

Experimental Design
This category is intended for those projects which are oriented towards experimental methods.

a. Is there a well formulated hypothesis?
b. Do the experiments confirm/refute the hypothesis (i.e. are they appropriate)?
c. Is the experimental methodology sound?
d. Can the experimental system be systematically modified to check the effects predicted by the hypothesis?
e. Is the work reported in such a way that the experimental results can be replicated?

Implementation
a. How the design was implemented. For example, was a good language chosen to do the project in? Are the algorithms chosen the best ones available?
b. The work is original and / or ingenious and shows imaginative invention.
c. The student is usually self-motivated and is usually willing and able to work and devise plans of action on his / her own initiative.
d. The programming techniques used are highly professional, particularly in terms of structure and testing (extent and thoroughness).
e. The student exhibits an excellent knowledge of the computer language concerned.

**Findings**
- a. The results / findings / conclusions are presented clearly.
- b. The results / findings are discussed in a mature manner, worthy of honours level study e.g. if the results are contrary to expectations, there is an explanation of why this is so.
- c. Insight in the interpretation of the results was clearly shown (i.e. valid conclusions were drawn from the results).
- d. The results were placed in context and related to other research.
- e. The candidate shows understanding of the research findings by suggesting valid extensions / new areas / new ideas.
- f. There is suitable reflection i.e. highlighting the problems/successes of the project.

**Achievement**
This section tries to answer the question: Is there a deliverable present? This can be concrete (like a piece of software) or a theoretical result (new proof, useful extension of theory etc).
- a. If it was necessary to learn a new system/concepts/language, the student has shown a good grasp of this new material/information.
- b. The product/solution works well: it demonstrates or achieves whatever it set out to demonstrate or achieve. (This achievement includes an investigation of a situation, rather than necessarily producing a "final" product).
- c. The product/solution demonstrates state of the art techniques or extends existing knowledge in a useful manner.

**Introduction and Background**
- a. The research question or system requirements were clearly formulated.
- b. The background research was thoroughly done, i.e. the student has a clear understanding of the required system. There are no major gaps in the background research for the system specification. That is, all relevant material has been covered.
- c. The research question is well motivated / justified (i.e. the student shows a knowledge of why he wishes to do the project, and what he expects to achieve). The student shows an appreciation for the importance of their project in the overall context of Computer science.
- d. The reader can gain sufficient understanding of what the student proposes to follow their arguments and understand their work.

**Project Assessment**
The project plan will have been assessed in the Project Proposal. Here we wish to establish whether the students have adhered to the plan and if not, what justification the have for deviating.
- a. Does the project adhere to the Project Proposal originally presented?
- b. If there are differences, are these suitably justified?
- c. Did the project follow the proposed project plan? If not, is this justified? Were there problems with equipment/resources?
- d. What lessons were learnt during the course of the project? Are these clearly stated and discussed?
Evaluation
Note that evaluation will mean different things for different kinds of projects. These points are therefore suitably non-specific. Rather than enumerate all permutations, the following points have enough generally applicability that they should apply to all projects.

a. Was an evaluation of the product/system attempted? For theoretical work, this equates to a satisfactory proof mechanism. Experimental systems will usually have statistical verification and analysis.
b. Was the evaluation comprehensive? Are their gaps, obvious oversights?
c. Was the evaluation technique used appropriate for the problem?
d. Did the work evaluate the correct questions? Were the conclusions drawn appropriate/reasonable?
e. Could the work be presented at a conference?
f. Could the work be easily extended to an MSc?

Final Presentation
a. There was good use of all the potential media available for bringing home the particular message of the project.
b. The issue (problem) was clearly and succinctly stated. There was no rambling / padding / waffling.
c. The approach adopted was justified / explained.
d. The findings were clearly presented and explained.
e. The transparencies reflected and supported the statements in (a) to (d) above.
f. Time was not wasted by being filled with insignificant detail (e.g. detail about trees and even branches when whole forests are left unexplained).

4.3 Project Failure
It is sometimes instructive to define success in terms of what should be avoided. Hence the following definition of failure:

A failure means that at least 3 of the following are true:

a. The student is incompetent / never really knows what comes next / is lost.
b. The work is not presentable anywhere (including in private!) and / or is an embarrassment.
c. The work achieves nothing at all.
d. There were significant mistakes in the final implementation of the work and the student could not / did not correct them or never realised they were there!
e. The student has very little knowledge of the literature in the area of the research.
f. The student cannot self-start and needs constant prodding.
g. The student is unwilling and unable to devise plans of action on his / her own initiative.
5. Miscellaneous Notes and Guidelines

5.1 Web Pages

1. For examples from previous years, see http://pubs.cs.uct.ac.za/honsproj/
2. Individual Web sites must be a statement of the work of the individual for the purposes of assigning a mark. Group Web sites must indicate the accomplishments of the project for the purposes of future work and dissemination in the research and development communities. The latter must be the main entry point and should contain relative links to each of the former sites.
3. All Web sites - individual and group - must be self-contained, with no external links (except well-established ones e.g., W3C) and only relative links used internally (with the exception listed below).
4. All Web sites must be packaged and submitted along with other documents for archival purposes. The final submission must be a single ZIP file containing all files for all members of the project, named by an underscore-separated list of the last names of the students involved (e.g., jones_smith_baker.zip)
5. Web sites must be completely static - this means no PHP, ASP, Server-side JS or any such scripts. If your project is a Web-accessible application, you must add an absolute link from the project documentation Web site to the Web application, labelling it as a non-permanent external link.
6. All supporting documentation must be part of each Web site. Document files must be in a portable format (e.g., PDF, HTML), with Microsoft Word and LaTeX files converted as needed.
7. Source code must be linked into Web pages as one or more compressed archives.
8. To submit your website, place the ZIP file in a Web-accessible location and send the URL of the ZIP file to hussein@cs.uct.ac.za, by the indicated deadline.

5.2 Intellectual Property

1. Within written documents, all referenced work must be properly cited.
2. The relevant licence agreements must be checked and adhered to for all third-party modules included with your source code and all multimedia objects (e.g., images, sounds) used in your Web pages.
3. All software and written output is subject to UCT’s existing Intellectual Property and Copyright rules.
4. If opportunities for commercialization arise, these must be dealt with through the proper channels, starting with the Supervisor and the Department.