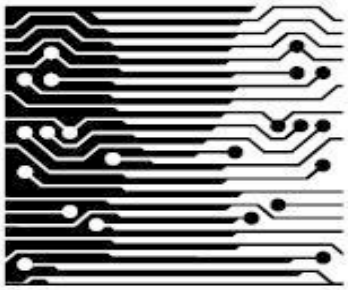

Department of Computer Science

Computer Science Building
18 University Avenue
Upper Campus
Rondebosch
Cape Town

Telephone: +27(0)21 650 2663
Fax: +27 (0)21 650 3551
Email: dept@cs.uct.ac.za

University of Cape Town
Private Bag X3
Rondebosch
7701

South Africa

Honours Handbook

2015

Revision January 2015

Compiled by Michelle Kuttel

Our Mission

As the Department of Computer Science, our mission is to develop and impart knowledge and skills in the field of Computer Science.

Our Vision

The Department of Computer Science strives to be of the first rank, maintaining excellence in both research and teaching, producing high-quality graduates skilled in problem solving, and thus play an influential role in the development of Information Technology, both within Africa and internationally.

The University of Cape Town is committed to policies of equal opportunity and affirmative action which are essential to its mission of promoting critical inquiry and scholarship.

1. Welcome to Computer Science Honours at UCT

Congratulations on your acceptance into the Honours Programme of the Department of Computer Science at the University of Cape Town! The Honours course completes your qualification as a computer professional: it is an opportunity to undertake advanced courses given by enthusiastic lecturers in their areas of specialization, as well as to complete a major research and development project. This will be an enriching and challenging year, which will require your firm commitment to full-time, hard work.

We hope that you enjoy the course and wish you every success for the year ahead.

Note Well: As discussed below (Section 2.2) you must be at the University *two weeks before the normal start of the (undergraduate) academic year* in order to complete certain compulsory modules of the Honours course.

1.1. Professional Qualification

In Computer Science, four years of study are required for a professional qualification: it is only after four years that you will have covered the ACM/IEEE Computer Science Curriculum[1] in sufficient breadth to practice with confidence in the field of Computer Science. It is also only after four years that you will have fulfilled the Requirements of the British Computer Society (BCS) for a professional Computer Science qualification (this only applies to students have also completed their undergraduate studies in our department). Our Honours course is recognized as providing full Chartered IT Professional (CITP) accreditation[2], as well as partially meeting the education requirement for Chartered Scientist (CSci) registration.

Your Honours degree will certainly improve your career prospects locally and is an essential qualification for the global marketplace. Our Honours students have found employment in: large South African corporations, such as Investec, Old Mutual and Standard Bank; local branches of international companies, such as Amazon.com Development Centre in Cape Town; local games and media companies, such as Black Ginger and Wisdom Games; and innovative businesses, such as Wizzit Bank in Sandton. Many of our alumni also work internationally, for software giants such as nVidia, Facebook, Microsoft and Amazon.

1.2. Graduate Studies

There are significant differences between an undergraduate and a graduate degree. Graduate students are expected to be interested in deepening their knowledge and experience, particularly in Computer Science, but also in related fields.

Many of you will appreciate the additional freedom you have in the Honours course to choose coursework modules in specific fields that interest you. However, the additional freedom comes with additional expectations. As a graduate student, we expect you to:

- Work hard and go beyond the basic requirements. This is your year to stretch yourself!

[1] http://www.acm.org/education/curricula-recommendations

[2] http://www.bcs.org/server.php?show=nav.7065

- Behave as a professional; arriving punctually for all classes, meetings and seminars and submitting all assignments by the posted deadline (assignment deadline extensions are not allowed in Honours).
- Take responsibility for your own study programme.
- Work largely unsupervised and independently.
- Approach problem solving independently and creatively and show an appreciation for concepts and principles.
- Evaluate, criticise and justify your own work - and others' work.
- Respond to criticism by improving your work.
- Communicate ideas clearly and succinctly.
-

Our Honour's programme produces skilled graduates, graduates who understand the process of research and development, who are critical thinkers and who can work productively, both independently and as part of a team. During the course, you will learn to read relevant literature, to formulate research and and write development proposals. You will also gain practical experience in teamwork, through initiating and managing a major project. You will learn to communicate your ideas and results clearly in your final thesis.

1.3. Role of the Department

We have a team of qualified, established researchers in the Computer Science department, comprising some of the best Computer Scientists in the country. In general, our role is to:

- Produce skilled, high quality graduates who are familiar with the principles, theory and practice of Computer Science.
- Carry out innovative research.
- Provide services to Industry, through technology transfer and applied research.
- Take an active part in the academic and governance affairs of the University.
- Provide opportunities and support for students from disadvantaged backgrounds.
- Promote, support and advise schools in the teaching of Computer Science Technology.

In our graduate programme, our role is more specifically to produce individuals who are educated, articulate, and able to exercise critical judgement in the field of Information Technology. Our core function, therefore, is not that of providing vocational training, but to impart the fundamental skills that are needed for decision making or creative thinking. We do not aim to train people how to use computers and become programmers to meet the immediate demands of the marketplace (although you will learn those things as a matter of course) - we want our students to remain useful scientists a decade from now.

1.4. Further Study

Students who successfully complete Honours in Computer Science at UCT are eligible to proceed to an M.Sc. in Computer Science by dissertation, provided that they find an approved supervisor for their intended course of study and research.

1.5. Programme Coordinator

The Honours programme coordinator is ***Assoc. Prof Michelle Kuttel***, Room 304.02, Computer Science Building (email: mkuttel@cs.uct.ac.za). Class representatives will be elected to whom issues may be addressed.

1.6. Financial Assistance

Financial assistance is available for prospective Honours students. Please look at the University's Postgraduate Degree Funding web page[3] or email pgfunding@uct.ac.za. The South African National Research Foundation (NRF) provides a *limited* number of bursaries to South African citizens. The closing date for NRF bursaries is usually around the 30 September; the notification date is February of the next year. The University offers *UCT Council Honours Merit Scholarships* for students who achieve at least 65% in their undergraduate majors. A number of other bursaries are also available. You are advised to apply for all bursaries as early as possible.

Bursaries for Computer Science Honours at UCT are also generally available from the department's Advisory Board members and other companies. Please apply to the head of department for these bursaries.

In addition to the above, some members of the department have funds for research purposes, which may be available to students involved in specific projects. Students with legitimate financial difficulties may also receive departmental support. Any queries in this regard should be addressed to the programme coordinator.

1.7. Facilities

Honours students are accommodated in a dedicated laboratory with 24-hour access, a small kitchen and coffee area, lockers and workstations. Our stated machine policy is to have at least two machines for every three students: in practice every student has in recent years had access to their own machine.

Computing resources include servers and workstations running BSD, Ubuntu Linux and other Linux distributions and Microsoft Windows.

Software resources for teaching and development are drawn from Public Domain offerings, third party releases (e.g. IBM Rational Suite) and the Microsoft Developer Network Academic Alliance software program.

Computers are connected to 100Mb switched network, whilst servers are connected via 1Gb links. A wireless infrastructure provides connectivity wireless peripherals and notebook computers.

[3] http://www.uct.ac.za/apply/funding/postgraduate/applications/

2. Honours in Computer Science Programme

The Honours programme in Computer Science - CSC4000W or CSC4003W or CSC4016W - is designed to provide students with the professional basis for a career path in the computer industry, and/or to enable them to embark upon a research programme at Masters level.

CSC4000W is taken by students with a BSc with a major in Computer Science from UCT. CSC4003W is taken by students in their final year of the Bachelor of Business Science (Computer Science). All other students take CSC4016W.

2.1. Admission Requirements

The number of places in the Honours programme is limited and students are selected on merit from the list of applicants each year. Criteria for selection are your Computer Science mark achieved in each of your three years of undergraduate studies and, to a lesser degree, the marks achieved in mathematics. Students who have not achieved at least a 65% average in their final year of Computer Science will only be admitted in exceptional circumstances.

2.2. Structure of The Honours Year

The Honours year commences two weeks before the undergraduate courses. Since the courses given in the initial weeks are compulsory, it is **not possible to excuse any student from attendance during this period**. The Professional Communication Unit (PCU) segment of the Research Methods module is completed during the first two weeks of Honours. The New Venture Planning (NVP) module also runs in this period.

The remaining optional modules are given in three blocks: the first two blocks correspond roughly to the first two quarters of the first semester, while the shorter third block is at the start of the second semester. After the third block, students devote themselves exclusively to their full-time projects. The only exceptions to this structure are in the case of external modules taken by students.

All modules given in a block will be completed by the end of that block and ***no extensions will be granted to complete work after this period***. The projects will be allocated by the start of the second quarter and various project-related milestones have to be met from then on until the final report for the project is due (usually in late October or early November, depending on the year).

2.2.1. Work Load

Honours is an intensive, full-time course and may not be taken together with other courses or while you are employed. Permission to deviate from this will only be given in exceptional circumstances by the Programme Coordinator. Your weekly workload will be between 40 and 48 hours per week.

For each lecture hour you should allocate at least two hours of extra work to review material and for the associated tutorials and practicals[4].

[4] These two hours could be allocated as one hour of theory review and one hour of practical work for a standard module or some other appropriate combination for a more practical or theoretical module.

Approximately eight weeks have been reserved in the final term to allow students to focus entirely on their Honours project. All lectures and practicals will be concluded before this period commences. Before this final eight week block, you should allocate at least 5 hours per week to supervisor meetings, planning your project, reading background material etc., during the second and third quarters.

You should also allocate one hour per week to attend departmental colloquia. Computer Science colloquia are normally held during the lunch hour and normally on a Thursday. **Please note that attendance of at least 70% of the colloquia is mandatory.** Failure to do this will impact on the mark for Research Methods. A register of attendance will be kept.

2.3. Coursework

The Departmental coursework modules are listed in Section 4 (Compulsory Modules) and Section 5 (Elective Modules). We offer sufficient modules at Honours level for you to fulfil your coursework requirement. However, you may, subject to the restrictions mentioned below, take selected modules from other departments. Approved External courses are listed in Section 6. Students are also encouraged to take other external modules (given by other UCT departments) to broaden their education. Note, however, that any module you register for outside the department must be of an appropriate level, have relevance to Computer Science and **be approved by the programme coordinator**.

2.3.1. Lecture Periods

Lectures are scheduled in lecture periods 1–8. The timetable is drawn up in consultation with lecturers to best accommodate their lecture commitments and to even out the work load. We attempt to avoid lecture clashes, but these will occur in exceptional cases. If the lectures for two different modules coincide, you may only register for one of the modules.

2.3.2. Module Registration

A list of the modules available for the year will be handed out at the start of the course. You will be asked to indicate your choice of modules within the first 2 weeks of the course. **You may not register for more than 120 course credits all together.** Only the registered modules will be considered in calculating your final mark.

Additional modules for credit may be offered during the year, to take advantage of the expertise of visiting lecturers.

Apart from such additional modules, you may only drop or add a module with the approval of the lecturer concerned and the programme coordinator. Such approval must be requested and given *in writing* and will not be granted if more than 1/6th of the lectures have already been given in the module concerned.

2.3.3. Examinations

Modules are usually examined after the completion of the block in which the module was given. External courses are usually examined in the University examination periods (May/June and October/November). However, the Department is free to schedule examinations at any sensible time after the completion of the relevant coursework. Examinations written outside the department are scheduled by the department in question. There is typically one two-hour examination per 10 credit module in Computer Science Honours. Open book and take-home examinations are preferred by some lecturers.

The examination timetable is the responsibility of the teaching assistant and lecturers concerned and is drawn up shortly before the examination period.

After the mid-year examinations, students may be given an indication of how they performed. Note that only a provisional symbol is released after the mid-year examination, as it is University policy to release only a single mark for the whole Honours course and the exam papers will not have been seen by the external examiner at this stage.

2.3.4. MODULE CREDITS

An Honours coursework credit corresponds to 2.4[5] lectures worth of material. A 10 credit module would thus correspond to 24 lectures on the Honours timetable. The practical assignments associated with a module can either be included in the course or can make up an additional 5 credits. If the practical work is considered part of a module, then proportionately fewer lectures are given and this will be stated in the entry for the module.

A total of 160 credits must be obtained during the course of the academic year. Research Methods and New Venture Planning are compulsory modules that must be completed by every student. You may select any remaining modules as electives, with a **minimum of 80 credits**, up to a maximum of 100 credits.

In order to gain credit for a module, students have to obtain **at least 40% for the module**.

2.3.5. COURSEWORK MARK

To fulfil the Honours coursework requirement, the following rules apply:

- You must complete the compulsory **Research Methods** (10 credits) module successfully.
- You must complete the compulsory **New Venture Planning** (10 credits) module successfully.
- You must obtain credit for at least **80 credits of elective material** (you may take at most 100).
- The **best 100 course work credits** (including Research Methods and New Venture Planning) will count towards your final mark. Of these:
    - At most 40 credits from Mathematics of Computer Honours (MOCS) courses will be taken into account.
    - **At most 20 credits from outside the department** or **40 credits from Maths Of Computer Science (MOCS)** courses will be taken into consideration; this *does not apply for Business Science* students[6].
    - At most 10 credits may be practical credits.

2.4. THE MAJOR PROJECT

Students are required to complete a major project under the supervision of a member of staff, possibly in conjunction with an outside supervisor. The project offers you the challenge of completing a substantial research or software development task in a professional manner. Another objective of the project is to teach you how to plan and work as a team.

[5] The number was arrived at via a circuitous historical process.

[6] We regard the CSC4003W course for Business Science students as one with only 80 credits of coursework. A course in Business Strategy is taken and no courses outside of Computer Science or MOCS are counted. The average mark is therefore calculated as described on the best 80 credits.

Projects involving multiple students are required, but they are structured so that there are *readily identifiable components for each person to complete*. Your contribution to the overall project will be written up separately and so must constitute a piece of work that can be *independently assessed*.

2.4.1.1. Timing

The project topics are presented to the Honours class towards the end of the first quarter and allocated to the teams by the start of the second quarter.

You are expected to start work from the second quarter onwards and meet your project supervisor weekly. A final block of about eight weeks has been set aside in the last term, to allow you to work only on your project.

A great deal of importance is placed on making regular progress throughout the project period. A detailed list of milestones has been drawn up to help you plan your work. It contains deadlines and specifications of what has to be handed in or presented. The list is handed out when the projects are assigned.

2.4.2. Project Choice and Allocation

Normally academic staff will propose the projects, but you are welcome to submit your own project idea, provided that the project has significant Computer Science content, it can be run as a tema and that a staff member can be found to oversee the project. Contact the Honours programme coordinator at the start of the year for the full requirements — the department reserves the right to reject such proposals.

Once the list of projects has been released, it is your responsibility to discuss the projects you are interested in with the supervisors concerned. You need to form a team of students with the right expertise to complete the task. You will then make a prioritised list of project preferences. You have about two weeks in which to make your choice after receiving the list of projects. As soon as possible thereafter, the projects will be allocated. Every attempt will be made to accommodate your wishes, while equalizing the workloads of staff.

Each project group is required to produce a formal project proposal, which will be vetted by the staff at a formal project presentation. Guidelines for the proposals will be distributed once the projects have been approved.

2.4.3. Deliverables

The final project report must be handed in to the Honours Coordinator no later than the specified due date. A maximum of three days beyond the official hand in time is permitted, but you will incur a penalty of 10% of the allocated marks per day for such a delay. Extensions are only granted if the delays in completing the project are beyond the reasonable control of the student(s) concerned.

The project report should constitute a comprehensive description of your project. A document detailing what such a report should contain will be handed out when the projects are allocated. *No report may be submitted without the prior approval of the project supervisor*. The supervisor may require alterations and so the final draft must be available in good time for it to be read by your supervisor and for you to then revise it.

The deliverables of the project may change from year to year, but are likely to include:

- A formal project proposal and presentation
- A prototype deliverable
- A poster
- A project web page

- A self-reflection paper
- The (most important) project report

2.4.4. Award for Best Project

In 2008, the department together with Business Systems Group (BSG) instituted an award for the best project. This goes to the team who has achieved the best overall result in their project in a particular year. Winners to date are:

2008 *WiiRobot: Teleoperation of Rescue Robots in Urban Search and Rescue Tasks* by Jason Brownbridge and Graeme Smith.

2009 *Dynamic Content in Procedural Generation* by Richard Baxter and Zacharia Crumley.

2010 *Gesture-based Games with the iPad* by Pierre Benz, Nina Schiff and Daniel Wood.

2011 *A Sketch-based Interface for Modelling Trees and Plants* by Matthew Black, Mark Dahoner and Neil Goldberg.

2012 *Smart Security Systems in an Internet of Things Environment* by Alexander Comer-Crook, Simon Groll, Shaun Michaels

2013 *StockOut Web Services* by James Lewis, Sven Siedentopf

2014 *Detection and Visualization of Radio Frequency Interference* by Philippa Hillebrand, Gerard Nothnagel

2.5. Credit Requirements

2.5.1. Duly Performed Certificate

Students will only be allowed to proceed with the second semester if, by the end of the first semester, they have an overall **average of 50% in their course work,** having gained credit for at least:

1. 100 credits of course work (this includes the compulsory modules) in CSC4000W and CSC4016W
2. 80 credits of course work in CSC4003W (business science course).

Students who do not meet these requirements will be listed as having been *Refused a Duly Performed Certificate* and their class record will show DPR. Such students will be entitled to a refund of 50% of their course fees and may apply to repeat the course as outlined in Section 2.6.

2.5.2. Passing Honours

In order to obtain the Honours in Computer Science degree, students must fulfill ALL these requirements:

- at least **50% overall average**.
- The **Major Project** (60 credits):
    - in order to pass CSC4000W & CSC4003W, students have to obtain at least 50% for their project mark. CSC4016W students must obtain at least 40% for the project and an overall average of 50% for the course as a whole, in order to pass.
- At least 40% must be achieved in the Research Methods (RM) module, including full participation in the professional communications (PCU) module and attendance at least 70% of the departmental colloquia and other research meetings for Honours students.
- At least 40% must be achieved in the New Venture Planning (NVP) module.

- An average mark of at least 50% must be attained in the modules making up the best 100 course credits.
- No module will be considered for course credits unless a student has obtained at least 40% in that module.
- The final mark, which will be calculated as explained in Section 2.5.3, must not be less than 50%.

A student who achieves each of the above subminima will pass the course.

2.5.3. Computing the Final Mark

The final course mark will be computed as follows:

- The project mark counts 3/8 of the total (60 credits out of 160)
- The remaining 5/8 of the mark (100 credits) is calculated from the best courses taken. The modules which come into consideration for this mark are specified in Course Work. The courses counted must include Research Methods and New Venture Planning.

2.6. Computer Science Department – Policy on Repeating Honours

Students have no automatic right to repeat Honours if they fail to meet the requirements for awarding the degree. If a student wishes to repeat Honours, a new internal Honours application and a letter of motivation have to be addressed to the Honours Course Coordinator. Such applicants will be considered after all new students for the course who meet the criteria for admission have been accommodated. All applicants wishing to repeat the year, as well as students who do not meet normal admission criteria, will be considered together. All special applications for admission to Honours have to be made by the end of last week of December.

3. Honours in Mathematics of Computer Science

The departments of Mathematics and Computer Science offer a joint Honours degree in the Mathematics of Computer Science in the School of Mathematics: course MAM4007W.

The Programme convener for this course is ***Dr. Jonathan Shock***, of the Mathematics Department.

This specialized programme provides the background for a further research career in theoretical Computer Science. A wide range of advanced modules are offered and, by the end of the course, you will be able to read the relevant literature and formulate further research proposals. Adequate experience and appreciation of the practice of Computer Science is provided by the project, which all students have to complete. Emphasis is again placed on the effective communication of ideas and results.

The entrance requirements and available modules are listed in the Mathematics Department Prospectus.

3.1. Further Study

The Honours degree could lead to a Masters degree either in the Department of Computer Science or the Department of Mathematics.

3.2. Structure

The course structure is similar to the Computer Science Honours Degree. It is being revised and may differ from this description. A 160 course credits were required: made up of course work from the two Departments (at least 60 credits from the Mathematics Department and at least 40 credits from the Computer Science Department) and a minor project from the Computer Science Department counting 30 credits.

Students have to complete the Research Methods/PCU module course from Computer Science for 10 credits.

Students do not have to do the New Venture Planning course.

For details see the Mathematics Department Prospectus.

3.3. Minor Project- Computer Science

The requirements for the minor project are essentially the same as that for the major project except that a reduced workload is envisaged.

Students are encouraged to team up with Computer Science Honours students to take a smaller role in a project team. However, in exceptional circumstances, if no appropriate team can be found permission may be given to do a single person project.

As with the major project, it is a condition that the minor project has a significant Computer Science content and that a staff member from the Computer Science Department supervises the project, possibly in conjunction with an outside supervisor.

Unlike the major project, at least 40% must be achieved in the minor project (subject to an overall average of 50% for the year).

4. Compulsory Honours Modules

The modules listed below are all compulsory for all Computer Science Honours students.

4.1. Research Methods (RM)

Prerequisites: None

Course Objectives: The research methods module forms part of your practical training as a researcher and computer professional. The course emphasizes communication skills and introduces basic research methodology.

Credits: 10 credits (16 lectures)

Lecturer or Convener: ***Assoc. Prof. Michelle Kuttel*** and ***Dr Melissa Densmore***

Course Content:

- Communication and presentation skills (PCU – see next section)
- Types of research, how to find papers and how to read papers (MK).
- Research Ethics (MD)
- Scientific and technical writing (MK)
- Assessment - Reviews and rebuttals (MD)
- Research methods: qualitative and quantitative and both (MD)
- Literature reviews (MK)
- Research proposals and teamwork (MK)

Assessment: The practical aspects of the work will be evaluated through the Professional Communications course (25%), a submitted literature review for your project (35%) and the project proposal (40%).

Prescribed Book: Notes will be distributed.

All Computer Science Honours students have to obtain at least 40% for this module in order to continue with Honours.

4.2. Professional Communication Unit (PCU) [*PCU is a component of RM*]

The general aim of the course is to equip you with essential theory and practice in spoken, written and visual communication so that you can communicate more effectively at university and also in your chosen professional career. Your PCU course starts in February and runs over a 2-week period. It provides a foundation for other courses (e.g. New Venture Planning course, research module, etc.) as well as other oral and written tasks you will be set later in the year.

Course Objectives: You will be expected to:

a) Select a topic (based on theme provided) and plan and present a business presentation. This presentation is expected to be a professional standard in terms of:
   - format and organisation
   - appropriateness to audience
   - tone, language and style
   - visual material

b) Function effectively in small group activities. Team work is essential in university as well as professional life and you will be introduced to the following concepts:
   - interpersonal, non-verbal and intercultural communication
   - impromptu presentations
   - small group communication

- problem-solving and decision-making

c) Gather information, plan and write selected documents according to acceptable standards. These documents are expected to:
   - confirm to acceptable research methods, formats and referencing requirements
   - be suitable for target readers in content, style, tone and vocabulary
   - be well-planned, logically set out and argued
   - include appropriate, professionally executed and well integrated graphic materials.

Credits: the course contributes 25% to the RM course.

Lecturer or Convener: ***Assoc. Prof. Terri Grant***, e-mail: Terri.Grant@uct.ac.za.

Course Content:

Written Communication Content:

- selecting a document
- gathering information
- planning a document
- selecting appropriate style, tone and vocabulary
- writing logically and persuasively
- evaluating formats (conventional and electronic) and layout (principles of readability)
- integrating verbal and visual communication modes and practices
- documenting/citing accurately
- applying professional standards of presentation and delivery

Oral and Group Communication Content:

- communicating effectively - a cyclical and negotiated process
- listening actively
- considering differences in perception, values and beliefs
- assessing verbal and visual cues
- planning an oral presentation
- integrating visual modes
- checking timing and balance
- handling questions
- handling stress/anxiety
- introducing group dynamics
- problem-solving and decision-making

Practical Assignments:

As the PCU course is a practical, highly intensive and a compulsory DP course, you will be expected to attend all workshops. Some of the sessions will be divided into separate time slots for (e.g. video).

You will receive various short assignments for both courses. Only the final written assignment (feasibility study/proposal/business plan) is a shared assignment. These will be marked separately using different course criteria. Your marks may therefore differ.

Your PCU and NVP facilitators will mark your final team presentations. This mark, collaboratively negotiated, will be added to your PCU and NVP marks respectively to provide an overall mark for each course.

Prescribed/Recommended Books: *Communicating @ Work* is the prescribed text. This, together with an exercise and assignment booklet, will be handed out in the first session

(and is provided as part of your Honours module). You must provide your own writing and visual/graphic materials for exercises/assignments/orals.

4.3. New Venture Planning (NVP)

Entrepreneurship is a critical element of economic development, for both developed and developing nations. An understanding of the process of entrepreneurial development and the elements of the entrepreneurial process is a valuable part of any learning programme. The New Venture Planning course will introduce students to the ideas, theories and concepts associated with entrepreneurial ventures, with a focus on the elements needed to develop a viable business plan. The discipline of thinking through all the various aspects of starting and operating a business or innovation.

Course Objectives:

Forecasting the future is a hazardous occupation, and no business, no matter how carefully planned, ever ended up exactly as it was intended to. However, the very process of producing a business plan provides benefits that are of enormous value to the business, even if in the final reckoning things turn out quite differently to how they were planned.

The learning experience will enable you to develop:

- The discipline of thinking through all the various aspects of starting and operating a business or new innovation;
- The skill to identify areas in which a business or innovation has particular advantages or weaknesses;
- The skill to determine with a reasonable degree of certainty whether or not a business or innovation is viable - before investing (and potentially losing) money;
- The ability to know why and when you need to write a business plan and be able to produce a document which demonstrates to business associates - bankers, investors, and others - that you have carefully considered the options and the practicalities of starting or expanding a business.
- An understanding of the lean start-up methods and approach.
- A greater understanding of the importance of the process of team formation and management in a multi-disciplinary project team.

The most effective way to understand the entrepreneurial process is to take a hands-on approach. To get financing for your idea you need to formulate your business idea and form your management team with the investor's perspective in mind. This means showing clearly and concisely what the customer benefits are and how it will generate revenue. You will focus on primary sources of information. You will need to talk to customers, users, distributors, industry organisations, competitors, designers, engineers, vendors, consultants, investors etc.

Credits: 10 credits

Lecturer or Convener: ***Sarah-Anne Arnold***

Practical Assignments:

Idea Pitch

Teams will have three minutes to pitch their business ideas. No PowerPoint may be used for the presentation, and not all team members need present. The elevator presentation is a fast, slick pitch of the idea that the team has developed.

Business Model Canvas

The business model canvas is a one-page framework with the nine basic building blocks that show the logic of how a company intends to make money. These nine blocks cover the

four main areas of a business: customers, offer, infrastructure and financial viability. Teams will blueprint their idea via this framework and present this to the class for discussion and input. Grades are awarded for depicting the business idea in a manner that is simple, relevant and intuitively understandable yet still demonstrates the complexities of how an enterprise functions.

Final Presentation

Each team will have 20 minutes to make a formal presentation of their business idea. This presentation will include all the relevant high-level information that is contained in a business plan. The intended audience are funders (banks, venture capitalists, etc). The presentation must be done using presentation slides and all team members are required to present.

Final Written Business Plan

The final written business plan should not exceed 30 pages including preliminaries, appendices and references and will be assessed based on its completeness in covering all aspects of the business venture. A financial viability model will need to cover income, expenses and cash flow projections for the business as well as capital expenditure and calculations of return on equity/capital.

Class sessions: Class sessions will focus on both content and general discussion and class activities; to ensure your active participation please read suggested pre-readings before class and come prepared to share and discuss your own ideas and experiences.

Each class session comprises three lectures; two lectures will be used for class activity and theory discussion and one lecture will be used for supervised group work. Attendance at all lectures and group work sessions is compulsory. Group work sessions will culminate in a number of 'one page' deliverables, the content of which will be discussed at each group session.

Assessment: The course assessment is made up as follows:

Task	Weighting
Idea Pitch (team)	10%
Business Model Canvas (team)	15%
Final presentation (team)	35%
Final written business plan (team)	40%

All Computer Science Honours students have to obtain at least 40% for this module in order to continue with Honours. This course is not compulsory for Mathematics of Computer Science Honours.

5. Elective Honours Modules

Coursework modules offered at the Honours level vary from year to year, depending on the interests of the current Computer Science staff. The following list of modules is provisional for this year and subject to change. Modules are listed in alphabetical order. You may take any module, as long as you satisfy the individual prerequisites listed.

5.1. Advanced Topics in Information Security (ATIS)

Prerequisites: None

Course Objectives: This module focuses on selected topics in information security from a theoretical perspective. *(Please note well: this is not a computer hacking course!)*

Credits: 10 credits (12 lectures, and 6 tutorials).

Lecturer or Convener: ***Dr Anne Kayem***

Course Content:

- Authentication and Authorization
- Access Control
- Software Security
- Privacy
- Topics on Unconventional Security Issues

There will be six in-class tutorials, comprising either pencil-and-paper exercises or programming exercises

Practical Assignments: None

Assessment: Test (In-Class) - 20%, Examination – 80% (Eligibility for sitting the exam is conditional on attendance of 75% of lectures and tutorials in addition to achieving a 50% average on the test held in the middle of the course).

Prescribed/Recommended Books:

- Computers and Security: Principles and Practice by Stallings and Brown (3rd ed).
- Introduction to Automata Theory, Languages, and Computation by Hopcroft, Motwani, and Ullman
- Distributed Algorithms by Lynch

5.2. *[CANCELLED]* Community-Based Co-Design (CBCD)

Prerequisites: Students are required to attend the lectures given in the ICT4D module, even if they do not formally complete the rest of the ICT4D module. Intake is limited to 16-20 students because of the field work component. Students must have a professional and respectful attitude towards clients and co-designers. It is an absolute requirement that students actually complete all the deliverables agreed to: results, not excuses, are required.

Course Objectives: Information and Communications Technology for Development (ICT4D) projects, that is, projects with rural and disadvantaged communities, tend to be failure prone. We have developed a method of Community-Based Co-Design (CBCD) [1, 2,3] in response to this situation. This course uses field work to convey CBCD in a practical

and effective fashion. “Community-based” means that design is user centred, and driven by a community. “Co-design” means both the computer experts and the community members are designers on an equal footing and work cooperatively. In a cyclical fashion the designers develop according to their skills and learning and according to the users’ expressed requirements and their learning. Conscious reflection is conducted after every cycle so that you, the designer, can learn.

Credits: 10 credits

Lecturer or Convener: ***Prof. Edwin Blake***, with outside lecturers.

Course Content: This module starts with a few lectures (7 or so), but most of the course is based on experiential learning doing field work outside the lab. This will involve whole mornings or afternoons on days to be decided.

The lecture topics are drawn from the following:

- Introduction to ICT4D.
- Design approach fusing action research, industrial design approaches, software engineering and participatory design.
- User requirements elicitation. Identification of stakeholders.
- Review of techniques for conducting focus groups, creating paper prototypes and technology probes.
- Field work will cover some of the following:
- focus groups
- generative sessions
- cultural probes
- context mapping
- technology probes
- interviews

The outcome of the course will be a practical deliverable that can be used by the community.

Practical Assignments: Field work will be required: details to be decided collaboratively by students, staff and community.

Assessment: Based on field work reports: both group-based report and a final personal report. Weighting will be decided but will be almost equal for the two. The final handin will be a given as a 24-hour exam.

Prescribed/Recommended Book: here is no prescribed book. To get an idea, have a look at:

- [1] E. Blake, W. Tucker, M. Glaser and A. Freudenthal, “Case study 11.1: Deaf telephony: Community-based co-design,” in Interaction Design: Beyond Human-Computer Interaction, 3rd ed., Wiley, 2011, pp. 412-413. www.id-book.com/casestudy_11-1.php
- [2] H. Winschiers-Theophilus, N. Bidwell and E. Blake, “Community Consensus: Design Beyond Participation,” Design Issues, vol. 28, no. 3, pp. 89-100, 2012.
- [3] Edwin Blake, William Tucker, and Meryl Glaser. “Towards communication and information access for Deaf people”. South African Computer Journal, 54:10-19, 2014.

5.3. Computational Geometry for 3D Printing (CGP)

Prerequisites: Computer Graphics from CSC3020H or Applied Mathematics 2. C++ from CSC3022H will be very useful but not essential.

Course Objectives: To master surface and volumetric modelling concepts applicable to 3D printing.

Credits: 10 credits (10 lectures and three practicals), 15 credit option possible, with additional practical work.

Lecturer or Convener: ***Assoc. Prof James Gain***

Course Content: The use of 3D printers for rapid prototyping is becoming increasingly prevalent. However, the process used by most current 3D printers of depositing thin layers of semi-molten material, which is known as Fused Deposition Modelling (FDM), is not without limitations. Factors such as material thickness and support structures need to be considered. This course will cover the theoretical concepts required for creating geometric models suitable for 3D printing. From a practical perspective, students will code modelling software, then design and ultimately print a 3D model.

Topics covered include:

- Geometry and Topology for Computer Graphics
- 3D Printing Concepts: Printing Hardware, Overhang Support, Applications
- Volumetric Concepts: Voxels, Computational Solid Geometry, Isosurface Extraction
- Surface Concepts: Parametric Surfaces, Mesh Smoothing, Free-Form Deformation

Practical Assignments: Three practicals (4-5 hours each). These build successively towards a final self-contained modelling package. A substational practical extension (5 credits) is optional.

Assessment: Exam: open book, 2 hours, 40%; Practical assessments, 15% each; Final printed show piece, 15%.

Prescribed/Recommended Book: No prescribed text but research papers and readings will be provided.

5.4. Database Systems (DBS)

Prerequisites: CSC2002S/CSC3002F database module; or any other introductory module on relational database design and use.

Course Objectives: The aims of the course are to introduce new developments in database systems, to study how to use such technology effectively and to provide an understanding of established techniques as well as research issues in these areas.

Credits: 10 credits (16 lectures)

Lecturer or Convener: ***Assoc. Prof. Sonia Berman***

Course Content: Object-relational systems; NoSQL; Distributed databases; Data warehousing, OLAP and data mining; Spatial data management; Big Data; Research topics in database systems.

Practical Assignments: 1 group assignment, and 1 individual assignment.

Assessment: Examination 60%; Assignments 40%.

Prescribed/Recommended Book: There is no prescribed book.

5.5. Digital Forensics (DiFor) *Intensive course*

Prerequisites: No specific pre-requisites, but student should have an understanding of computer systems fundamentals, networking and preferably programming. Students should also be familiar with research methods and research techniques – either as a pre-requisite or as a co-requisite.

Course Objectives: The objective of this course is to develop students' knowledge and understanding of digital and computer forensics principles and applications. In the context of digital and computer forensics, the aim of the course is to provide students with the knowledge to professionally, systematically and impartially approach the identification, preservation, recovery and analysis of all relevant digital evidence from computers, computer systems and computer networks (including the Internet) using appropriate tools and techniques. The principles of maintaining the integrity of digital evidence in the securing, recovering and analysing of that evidence will be explored in depth from a range of different sources of potential digital evidence. As well as developing the theoretical and technical skills the course will critically examine the more contentious and ethically sensitive areas associated with digital and computer forensics.

Techniques such as the use of inferential statistics and quality assurance principles associated with digital and computer forensics will also be addressed. The particular ethical issues facing the computer forensics practitioner and the professional requirements of the computer forensics practitioner will be discussed in context throughout the syllabus of this course.

Upon successful completion of this course, students will have demonstrated the ability to :

- Critically evaluate the professional and ethical issues involved in computer forensics and critically analyse and evaluate the professional requirements of a computer forensics practitioner.
- Identify, analyse and evaluate the current technical developments, advances and trends in computer forensics and ethical hacking and appreciate the impact of these issues on the disciplines
- Provide judicious consideration of sensitive and emotive subjects in digital and computer forensics and how to maintain evidential integrity in these subject areas.
- Demonstrate the skills required to be able to create, maintain and preserve an audit trail or other record, to professional standards, of all processes applied to computer based electronic evidence, so that a 3rd party will be able to examine those processes and achieve the same results – from a variety of evidence sources including hard disks, peripheral devices, computer networks and mobile phones
- Demonstrate relevant skills in identifying issues, undertaking research and performing critical analyses in the domain of digital forensics

Credits: 10 credits (24 lectures and no practicals).

Lecturer or Convener: Guest lecturer ***Prof. Alastair Irons***

Course Content: The course contents are provided in a series of sections – the weighting given to each section is indicated in percentage terms.

Digital and Computer Forensics (30%)

The topics in this section will focus on the critical evaluation of the principles of evidential continuity and evidential integrity as well as the ACPO principles (and South African equivalent where appropriate) associated with gathering digital evidence, and apply these in order to introduce students to the fundamental and defining principles of digital forensics. The discussions on digital forensics will include an examination of digital evidence and how the use of digital evidence can be applied to resolution of computer crime. Students will be introduced to the range of techniques for identifying, gathering, recovering, preserving, analysing and presenting digital evidence. Students will be introduced to the evolving terminology of computer crime and computer crime investigation through a critical examination of the evolution of investigative tools, the language of computer crime investigation and the role of computers in computer crime, including hacking, cracking, phishing, virus generation and computer misuse. Students will be encouraged to develop skills in the use of computer forensics software (in both stand alone and network environments) identifying the strengths and weaknesses associated with particular software and develop an understanding of which software to use in particular situations.

Digital Forensics Practice (20%)

An introduction to the forensic techniques in the examination of common operating systems, such as Windows, Unix and Macintosh, covering boot disks, file systems, overview of digital evidence processing tools, data recovery, file traces, system registry, log files, Internet traces, web browsing, e-mail, network traces and network storage. The forensic examination and the gathering of digital evidence will include the development of techniques and the use of tools for a series of computer forensics tasks such as the exploration of “hidden locations” such as caches, spool files, swap files, unallocated disk space, backup media, memory and Internet addresses. Techniques and the use of computer forensics tools will also be developed to help in the analysis of file access, modification, and creation time. Students will have the opportunity to make use of hashing algorithms to find and confirm the identity of files and disks. Students will be introduced to the functionality of a range of computer forensics tools for resolving computer forensics problems in a variety of environments and problem situations. Students will be encouraged to develop their application skills in computer forensics tools, for example EnCase, FTK, Kazalyser and NetAnalysis (and making use of freeware tools), in order to analyse case studies and solve computer forensics problems.

Current issues in Digital and Computer Forensics (10%)

A critical examination of current issues developed through research and critical analysis of those issues. Students will develop skills in identifying current issues, analysing and evaluating those issues. Students will develop their research and critical analysis skills, knowledge and understanding. Current academic research will be incorporated into the module through “research informed teaching”.

Legal, Professional and Ethical Considerations (20%)

A critical examination of the legal aspects and legislation that have direct relevance to computer forensics and ethical hacking will be covered. The material covered in this section will develop a framework for professional responsibility within computer forensics and ethical hacking, drawing on research undertaken by the course lecturer. The section will examine the principles of professional responsibility and liability and set these in the context of ethical behaviour through the application and discussion of ethical theory. Professional and ethical issues will also be explored in relation to high profile case studies.

Students will be expected to appreciate the various legislative statutes pertinent to the discipline and understand how to comply with investigative principles pertaining to these statutes – particularly in the South African context.

Consideration of Sensitive Issues (20%)

Techniques and methods for enabling computer forensic practitioners to deal with such sensitive and emotive subjects will be considered, including the professional requirements in counselling and dealing with post-traumatic stress disorder.

This section of the syllabus will focus on sensitive subjects in computer forensics such as incidents on the high end of the COPINE scale, the tracking of sex offenders (predominantly on the Internet), tracking paedophiles and dealing with child pornography. The particularly sensitive subject area of child pornography will be covered in order to prepare students for handling such cases and to raise awareness of the types of perpetrator they will be dealing with in professional cases. Consideration will be given to the techniques associated with identify, tracking and analysing child pornography cases whilst protecting the professional integrity of the computer forensics practitioner.

This section will also include a critical discussion of topics such as Internet crime, surveillance, dataveillance, cyberstalking, the use of spyware and computer intrusions. The discussions will provide a computer forensics perspective on technical consideration of how viruses, Trojans, worms and other malware operate and what can be done detect such intrusions, such as; activity monitoring, integrity checking, string searching, real-time scanning and heuristic scanning.

Practical Assignments: None – although practical work undertaken on the course can be used as examples in the report assessment and the examination.

Assessment: Report – technical report on a digital forensics topic reflecting research undertaken by the student (50%)

Examination (50%)

Prescribed/Recommended Book:

- Casey, E., (2010) *Digital Evidence and Computer Crime*, 3rd edition Elsevier Academic Press

Supporting Texts

- Bryant, R., (ed) (2008) *Investigating Digital Crime*, Wiley
- Ferraro, M.M. and Casey, E., (2005) *Investigating Child Exploitation and Pornography; The Internet the Law and Forensics Science*, Elsevier Academic Press
- Bainbridge, D., (2007) *Introduction to Computer Law*, 6th edition, Longman
- . Lucy, D., (2005), *Introduction to Forensic Statistics*, Wiley

5.6. Evolutionary Computation (EC)

Prerequisites: Programming skills in Java (including data structures and algorithms) are required. A basic understanding of genetics and evolution is useful, but not required.

Course Objectives: Evolutionary computation entails the use of simulated biological evolution to solve problems that are difficult to solve using traditional computer science and engineering methods. This course examines different evolutionary algorithms (EAs) and the types of problems EAs are best suited to solve. Course objectives include: gaining an understanding of various evolutionary computation techniques, identifying EAs suitable for solving different types of problems, and how to apply EAs to optimisation, machine learning, or design tasks.

Credits: 10 Credits (12 lectures)

Lecturer or Convener: ***Dr Geoff. S. Nitschke***

Course Content:

- Introduction to Evolutionary Computation.
- What is an Evolutionary Algorithm?
- Genetic Algorithms.
- Evolution Strategies.
- Evolutionary Programming.
- Genetic Programming.
- Niching.
- Multi-Objective Optimisation.
- Co-evolution.
- Working with EAs.

Practical Assignments:

Implement an evolutionary algorithm to solve a given optimisation problem, and give a 15 minute presentation on the solution implemented. The assignment is to be done in pairs.

Assessment: Assignment (40%), Exam (60%) .

Prescribed/Recommended Book:

- Eiben, A. E., Smith, J. E. - Introduction to Evolutionary Computing (Natural Computing Series) 1st ed. 2003. Corr. 2nd printing, 2007, ISBN: 978-3-540-40184-1

5.7. High-Performance Computing (HPC)

Prerequisites: Proficiency with C/C++ programming.

Course Objectives: In this era of heterogeneous computing, parallel programming is done using a range of technologies on a variety of multicore and accelerator architectures. This course will give both an overview of and practical experience with parallel programming for a range of architectures.

Credits: 15 Credits (20 lectures + large assignment)

Lecturer or Convener*: Assoc. Prof Michelle Kuttel*, guest lecturers ***John Stone*** (Associate Director, CUDA Center of Excellence at University of Illinois at Urbana-Champaign, USA and NVIDIA CUDA Fellow) and ***Assoc. Prof. Manuel Ujaldón*** (University of Malaga, Spain, NVIDIA CUDA Fellow)

Course Content:

- Overview of parallel computing [1 lecture].
- Parallel Computing with OpenMP for multicore [2].
- Parallel Computing with MPI for clusters [2].
- Parallel Computing with CUDA for Nvidia GPUs [10 lectures + tutorials].

Practical Assignments:

There will be a single assignment comparing parallel programming technologies. The assignment may be done in pairs.

Assessment: This course is evaluated on the basis of a single large assignment, with a report component. All students are expected to do the full 15 credits.

Prescribed/Recommended Book: None.

5.8. Human Computer Interaction (HCI)

Prerequisites: None.

Course Objectives: This course covers how to design and evaluate interactive systems for real users both in the developed and developing worlds. We will look at both theory and practice of designing digital systems.

Credits: 10 credits

Lecturer or Convener: ***Dr Melissa Densmore***

Course Content:

- The practice of Interaction Design
  - Methodologies
  - Evaluation
- Insight into Design
  - Putting design principles into practice
  - Case Studies
- Human-computer interaction for development
  - Constraints and context (literacy, environment, networking, etc)
  - Methods and Case studies

Lectures will consist of both teaching and practical work – students will be expected to practice the methods taught for an overall goal of creating a usable and practical application, and to submit reports on their insights and results on a weekly basis. On days prior to theory and case study lectures, students will be required to submit brief structured summaries of the assigned readings, to help facilitate discussion in class.

Practical Assignments:

- Reading summaries
- One group project on design
  - Weekly reports/reflections on in-class practical work
  - Poster and demo of final application
  - Final report on rationale for final design and future work

Assessment:

- Reading summaries (10%)
- Group Project
  - Weekly reports (30%)
  - Poster and Demo (10%)
  - Final Report (20%)
- Exam (30%)

Prescribed/Recommended Book:

- Mobile Interaction Design, by Gary Marsden and Matt Jones (optional)

5.9. Information Retrieval (IR)

Prerequisites: Basic understanding of XML data is required. Some background on statistics and linear algebra will be useful.

Course Objectives: Understand how search engines work at an algorithmic level. Learn how to build and incorporate basic and specialized search engines into your own projects.

Credits: 10 credits (16 lectures and some practical work).

Lecturer or Convener: ***Assoc. Prof. Hussein Suleman***

Course Content:

- Introduction to Information Retrieval (IR)
- Models of Basic IR (Boolean, Vector, Probabilistic)
- IR evaluation and testbeds
- Stemming, Stopping, Relevance Feedback
- Models of Web and linked-data retrieval (Pagerank, HITS)
- Latent Semantic Analysis and Clustering
- Multimedia IR
- Cross-lingual and multilingual IR
- IR in Practice (CMSes, digital libraries, Web, social media, etc.)
- Selected topics from:
    - Distributed and Federated IR
    - Recommender Systems
    - Natural Language Processing for IR
    - Sentiment Analysis
    - Opinion Retrieval
    - Text Summarization

Practical Assignments: 1-2 programming assignments: to use and/or extend existing IR tools or build a new tool from scratch.

Assessment: Exam (take-home): 40%; Assignments: 40%; Class participation: 20%

Prescribed/Recommended Book: There is no prescribed book, but after the course you will know how to find all the information you need online!

5.10. Introduction to Image Processing and Computer Vision (ICV)

Prerequisites: Basic Linear Algebra (matrices, vectors etc); familiarity with Fourier Analysis or functional analysis would be useful. For the prac, familiarity with a GUI toolkit would be very useful.

Course Objectives: To introduce students to basic concepts in computer vision and image processing, oriented towards solving real world, practical image analysis problems. The student will be introduced to basic concepts from digital signal processing, and a foundation built that will allow understanding of how more sophisticated schemes such as image analysis/segmentation which can be used to describe image and volumetric data at a higher, more useful, levels of abstraction. case studies and papers will be examined which relate this to real-world problems.

Credits: 10 credits (10 lectures + 3 paper sessions + practical).

Lecturer or Convener: ***Assoc. Prof. Patrick Marais***

Course Content: A number of lectures will be presented by the course convener, interspersed with paper/review sessions in which topical papers are discussed and followed up by review questions.

- Basic Signal processing 1
- Image Transforms 2
- Feature Detection 1-2
- Object Descriptions 1
- Segmentation 1-2
- Registration 1

- Genetic Algorithms in Computer Vision 1
- Case Study 1
- Paper Reviews 6

Practical Assignments:

- Self-assessment exercises available (not for credit)
- 3 Paper Sessions (assessed by Review Questions)
- 2 week programming project.

Assessment:

- **Exam:** Open Book; 2 hours
- **DP Requirement:** 50% in class record (composed of prac and review questions)
- **Class Record**: Practical 50%, Review Questions 50%
- **Final Mark**: Exam 40%, Class Record 60%

Prescribed/Recommended Book: There is no prescribed book: notes will be handed out.

- **Recommended Book**: *Algorithms for Image Processing and Computer Vision*, J. R. Parker, Wiley Computer Publishing, 1997.
- **Useful Reference:** *Image processing, Analysis and Machine Vision (4th Ed.)*, Milan Sonka et al, Wadsworth Publishing, 2014.

5.11. Introduction to ICT for Development (ICT4D)

Prerequisites: None

Course Objectives: Understand basic ideas underlying ICT4D and how it is effected in practice. Learn about and critically evaluate ICT4D projects. Learn how to design and evaluate development-oriented computing projects.

Credits: 10 credits (24 lectures).

Lecturer or Convener: ***Assoc. Prof. Hussein Suleman*** (convener) with ***Prof Edwin Blake***, ***Dr Melissa Densmore***, ***Assoc. Prof. Ulrike Rivett, Dr Brian DeRenzi, Dr David Johnson*** and others…

Course Content:

- [1] Key concepts in ICT4D, e.g., digital divide, leap-frog
- [1] Development and ICT4D
- [1] Qualitative research methods
  - epistemology, action research, triangulation, pilot studies
- [1] Requirements elicitation
  - setting data goals, identifying participants & intermediaries, relationships with participants
- [5] Specialist area talks:
  - Edwin: Design and Prototyping
  - Hussein: Open Movements
  - Ulrike: TBD
  - Melissa: TBD
  - David: TBD
  - Brian: TBD
- [5] Invited talks:
  - On various technologies and techniques from external practitioners and researchers
- [1] Summary / workshop
- [8] Student talks:

- Critical evaluation of different ICT4D-aligned projects

Practical Assignments: None.

Assessment: Exam (take-home): 60%; Class participation: 40% (includes discussion and presentations)

Prescribed/Recommended Book: None.

5.12. Logics for Artificial Intelligence (LAI) *Intensive course*

Prerequisites: Familiarity with basic discrete mathematics is highly recommended.

Course Objectives: This course will introduce students to logics used in the area of Knowledge Representation - a subarea of Artificial Intelligence.

Credits: 10 credits (8 lectures and some practical work)

Lecturer or Convener: ***Prof. Tommie Meyer***

Course Content: Logic plays a central role in many areas of Artificial Intelligence. This course will introduce students to logics frequently used in the area of Knowledge Representation and Reasoning. There will be a strong focus on Description Logics, the class of logics that are frequently used to represent formal ontologies. Topics covered include the following:

- Introduction to Knowledge Representation
- Propositional Logic
- Description Logics
- Modal Logics
- Applications of logics in Artificial Intelligence

Practical Assignments: Students will be given a number of practical assignments. This includes an assignment involving the Protégé ontology development environment.

Assessment: Exam - 50%; Assignments - 50%.

Prescribed/Recommended Book: None. The course is based on lecture notes and papers.

5.13. Mobile Game Development (MGDEV)

Prerequisites: CS2 Games Course. This can be waived if a student can demonstrate sufficient understanding of and experience with games technology.

Course Objectives: To develop, as part of a team, a complete mobile game from the phases of conceptual design through to final playtesting on the target device (mobile phone or tablet) .

Credits: 15 credits (8 lectures and an extensive practical)

Lecturer or Convener: ***Assoc. Prof. James Gain***, ***Assoc. Prof. Patrick Marais***, ***Dr Hendranus Vermeulen***

Course Content: This course is very practically oriented, with the intention of developing student's skills in game design and development. The course is broken into the following components:

- Essentials of Mobile Game Development

- Conceptual Design
- Paper Prototyping and the Game Design Document
- Technical Design Document
- Playtesting

There will be meetings on a weekly basis in which development artefacts will be presented and discussed.

Practical Assignments: Each team will develop a complete Mobile Game.

Assessment: Intermmediate artefacts such as a Conceptual Design, Game Design Document, and Technical Design Document will be evaluated (counting 30%). The completed App will also be subject to assessment (counting 40%). There will be a final exam (counting 30%).

Prescribed/Recommended Book: There is no prescribed textbook. Sample design documents will be provided.

5.14. Network and Internetwork Security (NIS)

Prerequisites: Assumed knowledge of Networks and Operating Systems from CS3.

Course Objectives: The objective of this course is to introduce cryptographic techniques and protocols for secure exchange of information on networks and inter-networks, and to examine the deployment of these in emerging technologies.

Credits: 10 Credits (16 lectures + 1 practical)

Lecturer or Convener: ***Adjunct Prof. Andrew Hutchison***

Course Content: The course covers risk concepts; security services; conventional encryption (classical encryption techniques, DES/AES, key distribution, key generation); public-key encryption (RSA algorithm, key management, certification hierarchies); authentication and digital signatures; LDAP directory services for authentication & authorisation; security protocol analysis; authentication and key exchange (Kerberos, Diffie-Hellman); electronic mail security (S-MIME/PGP); world-wide web authentication / security (S-HTTP, SSL, capabilities); secure electronic commerce (SET); webservices security (WS-Security, SAML); cloud computing security (public vs private clouds)

Practical Assignments: A practical involving deployment of cryptographic algorithms will be conducted (in groups).

Assessment: Exam – 60%, Practical – 40%

Prescribed/Recommended Book: Papers and readings will be made available.

5.15. Networks for developing regions (NET4D)

Prerequisites: A basic understanding of computer networks is essential. The student should have taken the UCT CSC3002F Networks course or an equivalent course.

Course Objectives: This course will introduce you to the challenges of building computer networks in developing regions and will work through novel networking solutions used to overcome these challenges over the past decade.

Credits: 10 credits (16 lectures and two practicals)

Lecturer or Convener: ***Dr David Johnson***

Course Content: Building computer networks in developing regions brings a unique set of challenges and constraints that often require connectivity solutions that are either specially

adapted to these challenges or that take advantage of the green-field opportunities often present in these areas. This module covers advanced networking techniques and architectures to deal with these challenges. Both theory and practice in the field are presented. The topics are:

- Networking challenges, constraints and opportunities in developing regions
  - The current state of Internet in the world
  - Constraints in developing regions (Environmental, infrastructural constraints, regulatory, cultural, economic, political, language, technical)
- Connectivity options for developing regions - alternatives to commercial networks
  - Wireless Mesh Networks
  - Delay Tolerant Networks
  - TV White Spaces
- Network architecture options for developing regions
  - Customized Internet applications for unreliable networks
  - Smart caching strategies
  - Peer-to-peer sharing architectures
- Network performance, Traffic analysis and network maintenance in developing regions
  - Monitoring: performance analysis and fault detection techniques
  - Network traffic pattern analysis techniques
- Case studies of usage and performance of networks in developing regions
  - Case Linknet, Macha, Zambia
  - Case AirJaldi, Dharamsala

Practical Assignments: Two practicals – (1) constructing a mesh network or improving the hit rate of a squid proxy cache (2) Traffic pattern analysis of traffic from a rural network.

Assessment: Exam: 24 hour take-home exam, 60%; Practical assessments, 20% each.

Prescribed/Recommended Book: Material will be based on a reading list of papers.

5.16. Ontology Engineering (OE)

Prerequisites: None, but experience in modelling with ER or UML and some familiarity of First Order Logic will be helpful.

Course Objectives: The principal aim of this module is to provide the participant with an overview of ontology engineering—including language features, automated reasoning, and top-down and bottom-up ontology development—and a main application field being the Semantic Web.

Credits: 10 credits.

Lecturer or Convener: ***Dr C.M. Keet***

Course Content: Ontologies are used in a wide range of applications, such as data integration, recommender systems, e-learning, semantic scientific workflows, and natural language processing. While some of these applications pass the revue, the main focus of the course is on the ontologies. The topics covered include the following:

- Logic foundations for ontologies
  - Languages (Description Logics, OWL)
  - Automated reasoning (class and instance classification, satisfiability and ontology consistency checking)
- Ontology development

- Ontology engineering, top-down: foundational ontologies, ontology design patterns
- Ontology engineering, bottom-up: exploiting legacy material, such as relational databases, thesauri, text
- Methodologies for ontology development and maintenance, methods to enhance ontology quality and to automate some aspect of the methodology

Practical Assignments: There will be three assignments: developing a small ontology in an Ontology Development Environment (Protégé), a group project on a selected topic, and populating the OE Semantic Wiki.

Assessment: Exam - 50%, assignments - 50%

Prescribed/Recommended Book: There is no prescribed book, but extensive lecture notes and papers will be provided.

5.17. Software engineering (SE) *Intensive course*

Prerequisites: Basic understanding of Java programming.

Course Objectives: This course teaches the basics of software engineering. Using UML-Models to get from software requirements to a formal specification and java code. Also the use of Statecharts to model reactive systems is covered.

Credits: 10 credits (16 lectures).

Lecturer or Convener: Guest lecturers ***Prof. Dr Stefan Jähnichen, Alexandra Mehlhase***

Course Content:

The course comprises lectures on theory and tutorials for hands-on experience with the software development process applied to specific examples. It starts with the analysis of requirements and systematically derives architectures and formal specifications using UML diagrams.

- Requirements engineering
- Using UML models to analyse requirements and develop a software design
- Formal specification of methods and system operations
- transformation of the design to Java skeletons
- Modelling reactive systems with Statecharts

Practical Assignments: Practical exercises and assignments will be given for all the topics listed. The assignments have to be solved by student teams.

Assessment: Practical assignments 30%, examination 70%.

Prescribed/Recommended Book:

- Object-Oriented Software Engineering: Practical Software Development using UML and Java by Timothy Lethbridge, Robert Laganiere
- Software Engineering by Ian Sommerville
- Objektorientierte Softwareentwicklung by Bertrand Meyer

5.18. User Experience in Games and Virtual Environments (UXG)

Prerequisites: Third year Games course. Students who have a proven ability to design and implement an immersive game or virtual environment may apply to the lecturer for

exemption from this requirement. In any event, the course assumes you know design techniques and assumes you *already* have a modifiable game that you bring to the course.

This course requires a lot of independent work and is only for students who are able to take responsibility for their own learning.

Course Objectives: This course has two aims:

- To build on practical skills in creating games (or virtual environments — VEs) with the scientific skills to analyse users' experiences (UX) in them.
- To give you a deep understanding of quantitative scientific methods from experiment design, through experiment execution to analysis and writeup as a scientific paper.

Credits: 10 credits (8 lectures taught intensively at the start, thereafter practical work and write-up).

Lecturer or Convener: ***Prof. Edwin Blake***

Course Content: We'll first examine concepts of user experience in immersive games and VEs. The objective is learning what makes such games and environments effective and engaging. We will consider how you can measure user experience in a VE and game and consequently how to make them better.

There will be a very few lectures at the start to introduce important topics and then it will be up to you to devise the experiments in consultation with the lecturer. The format will be one of seminars and design discussions. The course is open-ended and you may suggest topics of interest. The topics are drawn from the following:

- Introduction
- What are Immersive Games & VEs?
- What is User Experience?
- Immersion: Flow and Presence
- Presence and Perception
- Immersion and Flow in Games
- Measurement of Presence and Flow in Games and VEs

Practical Assignments: This course will be very practically orientated.

You (and perhaps a partner, but you may work alone) will take one of your games or VEs (or, in exceptional cases, create a new one) and vary some aspects in a way that you hypothesize will have an effect on the users' experience (UX). You will then use your theoretical understanding of UX to design an experiment and analyse the outcomes and write up what was achieved.

Typically the work involves modifying a game to create a control and experimental condition and performing an experimental test on users of the effectiveness of the intervention. Other forms of quantitative experimentation on UX are possible.

Assessment: Students will be assessed on the practical work, including experiment design, (60%) and paper written as a 24-hour take home exam based on the outcomes of the work (40%).

Prescribed/Recommended Book: There is no prescribed book, selected readings and notes will be provided.

5.19. Visual Thinking and Visualization (VIS)

Prerequisites: There are no specific prerequisites for this module, other than a background in computing. However, as it is a design course, some interest in graphics/visual art/aethetics/design is required to appreciate the course content.

Course Objectives: It is increasingly important to pay careful attention to the design of data displays and software user interfaces. This fact is highlighted by the huge success of companies such as Apple and Google, who have prioritized intuitiveness and ease-of-use in their software interface designs and interactions. In this module, we cover the field of *visual thinking*, outlining current understanding of how humans think visually from a neurological perspective and demonstrating how we can use this knowledge to design for more effective visual images and interaction. This knowledge will be applied to the design of user interfaces and data graphics to facilitate user queries.

Credits: 10 credits (8 lectures and one large assignment, with two class presentations and critique).

Lecturer or Convener: ***Assoc. Prof. Michelle Kuttel***

Course Content: This module will cover the following topics:

- Visual queries and how the mind works to process visual information
- Structuring two dimensional space
- Colour
- Visual space and time: depth perception and motion
- Visual objects: how to design visual objects that are easy to identify
- Theory and best practice in the design of data graphics, interfaces and visualizations.

Practical Assignments: The single major practical will involve multi-stage design and testing of a graphical display. Topics will be listed in the first week of the course and design stages will be presented to class for discussion and critique. The practical is expected to involve about 16 hours of work.

Assessment: Exam - 50%, Practical - 50%.

Recommended Books:

- Visual Thinking for Design by Colin Ware and The Visual Display of Quantitative Information by Edward R. Tufte (second edition).

These recommended books are highly regarded internationally and will make wonderful additions to any Computer Scientist's library, but you don't *have* to buy them for this course.

6. External Modules

Students are encouraged to take external modules, subject to the subminima for external courses (Section 2.3). These external courses have to be approved by the Honours Programme Coordinator. They will be weighted according to their relevance to computing in general. A critical aspect of undertaking such a module is your getting the permission from both the lecturer and the head of the relevant department to undertake the course. They have to provide a contact person who will be responsible for providing your final mark on time. Such arrangements obviously have to be confirmed in writing. Popular modules in the past have included Robotics and Agents offered by Mechanical Engineering.

6.1. Mathematics of Computer Science Modules

Modules within the Department of Mathematics and offered as part of the Honours in Mathematics of Computer Science (Section 3) have been approved as external courses by the Department of Computer Science. Students doing CS Honours can do an extra MOCS course on top of the normal allowance of 20 credits for outside courses. This is a maximum of 40 credits worth of MOCS modules (that is 2 out of the 3 MOCS modules offered by Maths: Maths modules are 20 credits).

Additional information can be obtained from the Department of Mathematics. Only certain courses will be considered for credit, specifically those dealing with the foundational aspects of computing or other pertinent topics, such as quantum computing.

These courses are usually counted as 20 credits within the department (36 lectures + tutorials).

6.1.1. Cryptography (CRYPT, 36 lectures, first semester)

Lecturer: *Dr. Christine Swart*

Semester: First

Prerequisites: The course is geared towards Honours students in either Maths or Computer Science. Having done 2IA and 2LA is an advantage, but is not necessary. We assume some familiarity with matrices, but we will cover all the number theory and probability theory you need in the course.

Course Description:

Cryptography is the mathematics of information security, which means keeping digital information secret or ensuring that it cannot be changed without detection. In this course we first cover the two kinds of secret key cryptosystems (block ciphers and stream ciphers), along with cryptographic hash functions. We then learn some computational number theory, before studying the public key cryptosystems and signature schemes RSA and ElGamal, and methods for solving the factoring and discrete log problems. If time permits, we finish with elliptic curve cryptosystems. Emphasis throughout is on how all these systems can be attacked (using maths).

6.1.2. Complexity Theory (CT)

Semester: Second.

Prerequisites: None.

Course Objectives: This course provides an introduction to major topics in computational complexity theory, which is one of the core areas of theoretical computer science. In computational complexity, we investigate the power of efficient computation. That is, we try to distinguish between computational problems that can be solved efficiently in practice and those that, though theoretically solvable, are not solvable in practice because of prohibitively large time or space requirements. The central open problem is the P versus NP problem.

Credits: 20 (CS)/ 21 (MAM) credits. (36 lectures + 12 tutorials)

Lecturer or Convener: ***Dr Holger Spakowski***

Course Contents: Topics covered include: Nondeterministic and deterministic Turing machines; basic time and space complexity classes; space and time hierarchies; NP and NP-completeness; the polynomial hierarchy; oracle complexity classes; relativisation of the P versus NP problem; memory-bounded computation; probabilistic algorithms; circuits; counting problems; interactive proofs.

Practical Assignments: None.

Assessment: Exam - 60%, Class Record - 40%

Recommended Book:

- Sanjeev Arora, Boaz Barak. Computational Complexity: A Modern Approach. (first chapters) Available online: For instance http://www.cs.princeton.edu/theory/complexity/

6.1.3. Graph Theory (GT)

Lecturer: ***Dr David Erwin***

Semester: First

Prerequisites: An undergraduate degree in mathematics, including some group theory.

Course Description:

Graph Theory is an increasingly important area of modern mathematics. There are numerous applications of Graph Theory: Modelling the World Wide Web, the spread of disease, driving directions, and electrical networks, to name a few. This course, though, is delivered as a course of Pure Mathematics, i.e., it is a sequence of theorems and proofs.

Course Content:

- Introduction: Graphs and digraphs, degree, isomorphism, operations on graphs,
- distance, bipartite graphs, cut-vertices and bridges, trees.
- Connectivity, vertex and edge cuts, Menger's Theorem.
- Planar graphs, Kuratowski's Theorem, the Four and Five Colour Theorems.
- Vertex colouring, Brooks' Theorem.
- Eulerian and Hamiltonian graphs.
- Graphs and groups, permutation groups and the Cauchy-Frobenius-Burnside Theorem.

6.2. School of Management Studies: Strategic Thinking (BUS4050W)

This external course counts 20 credits and is typically taken by Business Science students. BUS4050W is the capstone course available only to final year Business Science students.

BUS4050W aims to give students an opportunity to improve their strategic thinking ability. The course focuses on both classic strategic management thinkers and includes guest lectures who share their real world experience of strategic thinking. Please consult the Faculty of Commerce handbook for more details.

6.3. Department of Computer Science: Computer Graphics

You may take the Computer Graphics module from the undergraduate Games Course for 10 credits. Please contact Games course convenor (listed in the Science Faculty handbook) for more information.