

Preface

This handbook contains information regarding all aspects of the Undergraduate Computer Science Programs offered by the Department of Computer Science, Memorial University of Newfoundland - from requirements for admission to our programs to degree requirements for graduation. It should prove to be a useful source of information for potential computer science majors as well as for those already in our program.

This handbook is to be considered a guide and not a legal document. Students are advised to consult the 2011-2012 university *Calendar* regarding specific degree regulations and university policies. For further information regarding the Department (including an on-line copy of this handbook), please consult our web-site at www.mun.ca/computerscience or contact our general office located in the S.J. Carew Building, room EN-2021, or write us at the following address:

Head

Department of Computer Science Memorial University of Newfoundland

St. John's, NL

Canada, A1B 3X5

E-mail: cs-compsci@mun.ca

In order to consult with our Undergraduate Advisor regarding our programs and courses, please e-mail cs-ugradadv@mun.ca

Contents

| | | |
|----------|--|----|
| 1 | General Information | 4 |
| 1.1 | The Computer Science Student | 4 |
| 1.2 | Improving Program Accessibility | 4 |
| 1.3 | Computer Science Program Offerings | 4 |
| 1.4 | Selecting Your Courses | 6 |
| 1.4.1 | Courses For Non-Majors | 6 |
| 1.4.2 | Minor in Computer Science | 6 |
| 1.4.3 | Computer Science Majors | 7 |
| 1.4.4 | First Year Course Selection (B.Sc.) | 8 |
| 1.4.5 | First Year Course Selection (BA) | 9 |
| 1.5 | Ethics of Computer Science Students | 11 |
| 1.6 | Information Required in Certificates from Health Professionals | 11 |
| 1.7 | Availability of Assistantships to Computer Science Students | 12 |
| 1.8 | Computer Science Student Help Centre | 12 |
| 1.9 | Fees | 12 |
| 1.10 | Job Opportunities and Continued Study | 13 |
| 1.11 | Faculty Listing | 13 |
| 1.12 | Staff Listing | 15 |
| 1.13 | Computing Resources | 15 |
| | | |
| 2 | Basic Structural Information | 20 |
| 2.1 | Prerequisite Structure of Core Courses | 20 |
| 2.2 | Prerequisite Structure of Required Mathematics Courses | 21 |
| 2.3 | Course Offerings | 21 |
| 2.4 | Course Numbering Scheme | 23 |
| 2.5 | Undergraduate Programs involving Computer Science | 23 |
| 2.5.1 | Major in Computer Science (BA or B.Sc.) | 23 |
| 2.5.2 | Honours in Computer Science (BA or B.Sc.) | 24 |
| 2.5.3 | Honours in Computer Science (Software Engineering) (B.Sc.Only) | 25 |
| 2.5.4 | Computer Industry Internship Option (CIIO) | 25 |
| 2.5.5 | Minor in Computer Science (BA or B.Sc.) | 28 |
| 2.5.6 | Computer Science and Statistics Joint Major (B.Sc.Only) | 28 |
| 2.5.7 | Computer Science and Statistics Joint Honours (B.Sc.Only) | 28 |
| 2.5.8 | Computer Science and Pure Mathematics Joint Major (B.Sc.Only) | 29 |
| 2.5.9 | Computer Science and Pure Mathematics Joint Honours (B.Sc.Only) | 29 |
| 2.5.10 | Computer Science and Applied Mathematics Joint Major (B.Sc.Only) | 30 |
| 2.5.11 | Computer Science and Geography Joint Major (B.Sc.Only) | 30 |
| 2.5.12 | Computer Science and Geography Joint Honours (B.Sc.Only) | 30 |
| 2.5.13 | Computer Science and Physics Joint Major (B.Sc.) | 31 |
| 2.5.14 | Computer Science and Physics Joint Honours (B.Sc.) | 31 |
| 2.5.15 | Computer Science and Economics Joint Major (B.Sc.Only) | 32 |
| 2.5.16 | Bachelor of Commerce (Co-operative) Concentrations | 33 |

| | | |
|----------|--|-----------|
| 2.5.17 | Joint Degrees of Bachelor of Arts and Bachelor of Commerce (Co-operative) | 33 |
| 2.6 | General Degree Regulations | 33 |
| 2.7 | Second Degree in Computer Science | 34 |
| 2.8 | Deferred Exams | 35 |
| 2.9 | Supplementary Exams | 35 |
| 3 | Description of Undergraduate Courses | 37 |
| 3.1 | 1000-Level Courses | 37 |
| 3.2 | 2000-Level Courses | 49 |
| 3.3 | 3000-Level Courses | 67 |
| 3.4 | 4000-Level Courses | 91 |
| 4 | Appendix | |
| 4.1 | Prerequisite Structure of Core Courses | 129 |

1 General Information

This section is an introduction to activities and resources available in the department. It is intended to help students pursuing or wishing to pursue undergraduate studies in the Department of Computer Science.

1.1 The Computer Science Student

We strive to turn our students into good problem solvers - whose problem solving tools are the technology and ideas of computation. Whether your interest is games programming, media design, mathematics, science, humanities, or computer systems design, every discipline - practical, theoretical or creative - is experiencing an ever-increasing dependence on computer technology.

Algorithmic problem solving - creating solutions (algorithms) that be executed on a machine - is fundamental to Computer Science. With a grounding in the fundamentals, a student can continue on to explore computer systems, software architecture, web applications, graphics, databases, computer networks, scientific applications, computer graphics, games systems, numerical applications, and any other subject for which computers have an application. Our job in the Computer Science Department is to provide opportunities for students to explore their computing interests, and one way we pursue this objective is with a selection of required and elective courses covering topics of interest and importance. It is our hope that this handbook will reflect this objective, and that you will find at least some of this material useful in matching your own interests in computing. Of course, feel free to us what we are missing!

The Department of Computer Science also provides a number of venue for activities, including seminars by prominent computer scientists, programming competitions and a student-run Computer Science Club.

1.2. Improving Program Accessibility

We have introduced a number of changes in the last two years to make our program more accessible: There are new courses available for non-majors in multimedia programming (Computer Science 1550) and in computation as a means of understanding the world around us (Computer Science 2000), and a distance offering for the information technology course (Computer Science 1600). We are working on a project to offer our courses for majors in a self-study mode for those who want accelerated access to advanced (and possibly more interesting) courses. In the fall of 2011, we are exploring the possibility of eliminating of mathematics prerequisites for our introductory courses.

If you have an interest or suggestions for developing our program, or you want to know if we have something to fit your interests, please come talk to us.

1.3 Computer Science Program Offerings

The Department of Computer Science has a wide range of offerings, incorporating hardware, software, theory and practice. Students with a strong interest in another major subject can consider

a minor in Computer Science to supplement their skills in their major discipline, or one of the joint degrees available to specific disciplines. If a minor will not fit your program of studies, a few or even a single course in computing may be worthwhile addition(s) to your education.

Students may declare their intent to enter one of our computer science programs at any point in their course of studies, but they are encouraged to do so as soon as possible. Our main program offerings are a B.Sc. degree (Faculty of Science) and a BA degree (Faculty of Arts). The main difference between these programs is the student's choice of electives to fulfill Science or Arts degree requirements. General and honours degrees are offered in both faculties. Students who continue from the general to the honours program have the opportunity for additional advanced course work and research activity under the supervision of a faculty member.

A separate application is required for entrance into an honours program, normally at the beginning of the third year of studies. Acceptance of students into an honours program is based on their academic performance in the Computer Science courses taken up to the time of application. Students intending to pursue graduate work in computer science are strongly encouraged to undertake an honours degree.

The department offers a number of joint interdisciplinary programs, the majority of which are available for both majors and honours. Interdisciplinary programs allow for a concentration of courses in computer science as well as in another discipline of the student's choice. Interdisciplinary programs are available with the Departments of Mathematics and Statistics, Geography, Physics and Physical Oceanography, and Economics. Specialized programs include a Software Engineering Option and an Industry Internship Option.

The undergraduate programs offered by the Department of Computer Science include:

Minor in Computer Science

Major and Honours in Computer Science (B.Sc. or BA)

Honours in Computer Science (Software Engineering, B.Sc. only)

Computer Industry Internship Option (CIIO) (B.Sc. or B.Sc. Honours)

Minor in Computer Science (B.Sc. or BA)

Computer Science Joint Major (B.Sc. only) with either

Applied Mathematics

Pure Mathematics

Statistics

Physics

Economics, or

Geography

Computer Science Joint Honours (B.Sc. only) with either

Pure Mathematics

Statistics

Physics, or

Geography

Detailed information on each of these programs is available later in this handbook and in the university *Calendar*.

Declaration of Academic Program forms are available from either the Office of the Registrar or the computer science general office. Applications for Honours degree are available from the Office of the Registrar.

Our Undergraduate Advisor can provide information, guidance and counselling. The incoming student enrolling as a computer science major is strongly advised to immediately contact the Undergraduate Advisor or the General Office, Department of Computer Science. **Students should consult the Undergraduate Advisor when preparing their course schedules each semester, in dropping or adding courses during a semester, or whenever a need warrants.** This can be done through email correspondence at cs-ugradadv@mun.ca. Note, however, that the final decision and the responsibility stay with the student.

1.4 Selecting Your Courses

1.4.1 Courses For Non-Majors

The Computer Science department offers a number of individual courses designed to appeal to non-majors who want to explore and develop their computing skills and knowledge:

Computer Science 1600 is an introduction to information systems technology, and develops database, spreadsheet and Internet knowledge and skills using a business application context. This course is recommended for business-oriented students but is open to everyone. It can be done concurrently with a first-year mathematics course.

Computer Science 2000 explores the world from a computational perspective. Robots, the Internet, the human mind, biological growth, language and logic are investigated in a series of classroom and lab experiences. This course is recommended for Arts students but is open to everyone. It has no prerequisites.

Computer Science 1700 is a general introduction to computer science and programming. In this course you can get some basic programming skills and find out what computer science is about. This course can be done concurrently with a first-year mathematics course.

Computer Science 1510 is about numerical problem solving and programming skills, directed at students majoring in the mathematical sciences, including mathematics, physics, and chemistry. It requires Mathematics 1000.

Computer Science 1550 is an introductory course in multimedia programming. You can build interactive animations for the web while you learn to program. This course is pending approval for fall or winter semester; check back with us as this new course offering develops.

1.4.2 Minor in Computer Science

Beyond one or two courses, a Computer Science minor supplements any degree with career enhancing skills. Our minor is very flexible, with only three required courses, allowing students to tailor their courses to a particular interest. For example, it is possible to accommodate a specific interest with courses related to computer graphics, software development, scientific computing, computer networking, artificial intelligence, robotics, information systems, mobile computing or games development.

For a Minor in Computer Science, a student must complete at least 24 credit hours in Computer Science courses, including:

1. Computer Science 1710, 2710, 2711.
2. At least 6 credit hours selected from Computer Science 3715, 3716, 3719, 3724, 3725, and 3754.
3. Three additional credit hours at the 3000 level or above.

For example, a student interested in software development might take a minor consisting of 1710, 2710, 2711 (the required programming and algorithms courses), 3715 (Network Computing with WEB Applications), 3716 (Software Methodology), 3718 (Programming in the Small), 4768 (Software Development for Mobile Devices), and 2760. A student with an interest in numerical or scientific applications might select the courses 1710, 2710, 2711, 3715, 3716 and 1510 (An Introduction to Programming for Scientific Computing), 2500 (Data Analysis with Scripting Languages) and 3731 (Introduction to Scientific Computing). For a focus in theoretical Computer Science, the courses might be 1710, 2710, 2711, 3719 (Theory of Computation and Algorithms), 3724 (Computer Organization), 2742 (Logic for Computer Science), 4742 (Computational Complexity) and 4743 (Graph Algorithms). Someone interested in information systems as a speciality could take the courses 1710, 2710, 2711, 3715 (Network Computing with WEB Applications), 3754 (Introduction to Information and Intelligent Systems), 2742 (Logic for Computer Science), 4754 (Database Systems) and 1600 (Computing and Information Technology). For a games programming orientation, a student could select 1710, 2710, 2711, 3724, 3719, 2742, 4751 (Computer Graphics), and 1550 (Multimedia Application Development).

In creating your minor program, you have to be careful to schedule courses and their prerequisites into your major program. Many of the Computer Science courses have Mathematics or other prerequisites which have to be accommodated, so be prepared to seek some help (such as from the Computer Science Undergraduate advisor) in laying out your program.

1.4.3 Computer Science Majors

The general computer science degree program, like most general degree programs in the Faculty of Science or the Faculty of Arts, specifies approximately half of the courses required for your degree. The required computer science courses are those felt by the department to constitute the **minimum** which could be expected of **any** student in computer science. A great deal of your program, therefore, will be made up of **elective** courses, both in computer science and other disciplines. These electives can be chosen from many areas, subject to the general regulations for the Faculty of Science or the Faculty of Arts. Those general regulations can be found in the university *Calendar*. The quality and usefulness of your degree to both yourself and prospective employers depend, to some extent, on the elective courses you choose. A general recommendation is to try to have a concentration of courses in one or two other areas: for example, an area in which you have a professional interest (say, an area in which you wish to work in the future) or an area in which you have a personal interest. In order to accomplish this, we recommend that you choose some area as a minor, and work to fulfill the requirements for a minor in that area as specified in the university *Calendar*.

The computer science programs offered by the Department of Computer Science keep pace with advancement in the discipline, and provide a well-balanced foundation in the science of information processing. Our programs cover all aspects and levels of computers and computing. We have expertise in a variety of areas, including artificial intelligence, bio-inspired computing, complexity theory, computer architecture, computer graphics and image processing, database systems, design and analysis of algorithms, distributed computing, human-computer interaction, information systems, parallel processing, pattern recognition, programming languages, robotics, scientific computing, software engineering, VLSI design and web-based systems. Computer science is a rapidly evolving discipline; the program at Memorial emphasizes the fundamental ideas and methods underlying the discipline.

There are many current software and hardware products in common use that you will not use in your program. In order to be aware of such products and ideas, you should undertake to read some of the popular journals and magazines. Early in your career as a student, magazines like *LINUX Journal* and *PC Magazine* may be of interest. As you progress through your program, we recommend that you read some of the more advanced journals such as the *Communications of the ACM* and *IEEE Computer*. Many more journals on particular technical and scientific topics are published by professional organizations or commercial publishers and are available in the university library. Finally, a whole range of web-sites is available at your fingertips, providing information on the newest processor designs, software methods or scientific applications of computing.

Degree programs in computer science have considerable flexibility. It is **your** responsibility to ensure that you fulfill all the requirements of your chosen program. Students may request, from the Registrar's Office, a course audit to determine their status on their program.

1.4.4 First Year Course Selection (B.Sc.)

All candidates for the B.Sc. degree must include certain core requirements in their program. These core requirements are:

- (a) six credit hours in English
- (b) six credit hours in mathematics (Mathematics 1090 and 1000, or Mathematics 1000 and 1001)
- (c) Computer Science 1710 and/or Computer Science 1700
- (d) six credit hours in **each** of two sciences other than mathematics.

The following is a typical first year course selection:

| Semester I | Semester II |
|--|--|
| Computer Science 1700* or 1710** (or science elective) | Computer Science 1710** or 2710*** |
| Mathematics 1000 (1090)**** | Computer Science 2742*** or science elective |
| English 1080 | Mathematics 1001 (1000)**** |
| Science elective | English 1101, 1102, 1103 or 1110 |
| Elective | Elective |

*Computer Science 1700 is recommended for students wishing to major in computer science, giving a general introduction to computer science and programming.

**Computer Science 1710 has a prerequisite of Mathematics 1090 (or equivalent), or Mathematics 1000. Only Mathematics 1000 can be taken concurrently.

***Computer Science 1710 is a prerequisite for Computer Science 2710 and 2742.

****Students completing Mathematics 1090/1000 will be required to complete Mathematics 1001 as well.

1.4.5 First Year Course Selection (BA)

All candidates for the BA degree must include certain core requirements in their program. These core requirements are:

- (a) six credit hours in English at the 1000 level
- (b) six credit hours in a single second language other than English (satisfied by French, German, Greek, Hebrew, Innu-aimun, Inuktitut, Italian, Irish Gaelic, Japanese, Latin, Mandarin Chinese, Russian, Sanskrit, Spanish or other languages that may be offered from time to time). This requirement may also be met by demonstration of equivalent competency in a second language.
- (c) six credit hours in numeracy/science (satisfied by biochemistry, biology, chemistry, computer science, earth sciences, Economics 2010, Economics 2020, engineering, environmental science, geography laboratory courses, mathematics and statistics, physics, psychology, Science 1000, 1150, 1151, 3000, 3001)

- (d) twelve credit hours in at least two humanities subjects (satisfied by classics, communications studies, English, French, German, history, Italian, Linguistics 2025, 2026, 2030, 2031, 2700, 2701, 2702, 2703, 2704, 2705, 2706, medieval studies, philosophy, religious studies, Russian, Spanish, or women’s studies)
- (e) twelve credit hours in at least two social science subjects (satisfied by anthropology, archaeology, economics, Environmental Studies 1000, Environmental Studies 2000, folklore, geography (non-laboratory courses only), history, law and society, linguistics, police studies, political science or sociology)

Some of the courses in (d) and (e) may be used to satisfy major and minor program requirements.

- (f) six credit hours in research/writing. These may be counted towards the major and minor programs, and may also satisfy six credit hours from (d) and (e). It is recommended that these courses be completed within the first 60 credit hours in your program. Prior to registration, a list of courses which may be used to fulfill this requirement will be posted on the website of the Faculty of Arts at www.mun.ca/arts.

Priority registration for research/writing courses will be given to students who have indicated Arts as their intended faculty on the form used for application to Memorial University.

Note: To satisfy the minimum core requirements, students shall take no more than nine credit hours in courses from any one discipline.

The following is a typical first year course organization:

| Semester I | Semester II |
|---|--|
| Course in major subject or core requirement | Course in major subject or core requirement |
| Numeracy/science course | Numeracy/science course |
| Second language | Second language |
| Research/writing course | Research/writing course |
| English 1080 (or 1020) | English 1101, 1102, 1103 or 1110 (or 1080, 1021) |

1.5 Ethics of Computer Science Students

All computer science majors are issued a password for usage on the computers maintained by the Department. Once computer passwords have been issued, students are requested to check their system accounts for electronic mail and messages frequently. Throughout the semester, announcements concerning social functions, part-time jobs, scholarships and general items of interest to majors will be posted on the network. As well, if for any reason the department needs to contact a particular student, we endeavour to do so via the electronic mail system. Inappropriate use of the electronic mail could result in a student losing computer privileges.

The department uses homework assignments both as a teaching device and as a major component of its assessment of each student. **It therefore requires that all programs, assignments, etc. submitted bearing an individual student's name be the work of that student alone.** Under normal circumstances, students may discuss assignments but may not jointly write solutions. In particular, direct copying of another student's assignment is regarded as cheating by all parties knowingly involved, and will be dealt with as such.

Also, devices, electronic or otherwise, are not permitted in the room during examinations unless specifically allowed by your instructor. Please leave your phone at home or somewhere safe. Under no circumstances will students be allowed to access phones or devices during examination periods.

Computer science students are expected to maintain a high degree of honesty, dignity and respect for their fellow students, faculty and associates, particularly while using the computing facilities. Students are advised to refer to the General Regulations of the university *Calendar* for further information regarding academic misconduct.

1.6 Information Required in Certificates from Health Professionals

Students who request permission to drop courses, to withdraw from university studies, to have examinations deferred or to obtain other waivers of university, departmental or course regulations based on health issues are required by the University to produce, in support of the request, a certificate from a health professional in the form of a note or letter. Such notes must be sufficiently specific to allow a proper consideration of students' cases. The University requires that all such notes must be on **letterhead**, must be **signed by the health professional**, must **confirm the specific dates on which the student visited the health professional and should include details on the following**:

- the degree to which the illness (or treatment, in the case of medication, for example) is likely to have affected the student's ability to study, attend classes, or sit examinations.
- the length of time over which the student's abilities were likely hampered by the condition (e.g., recurring and severe back pain over a two-month period would likely have a more adverse effect on studies than a single episode of back pain requiring bed rest for a week).
- the fitness of the student to resume studies (it is in the student's best interest not to return to studies prematurely).

Confidentiality: The University respects the confidentiality of all material contained in such notes. Students should request that their health professional retain a copy of such a note in case the note needs to be verified or reissued at a later date.

1.7 Availability of Assistantships to Computer Science Students

In any given semester, the Department of Computer Science employs a limited number of students to work as markers to correct student homework assignments, to work as laboratory assistants in computer labs or to work in the Computer Science Help Centre, advising and assisting students enrolled primarily in 1000- and 2000-level computer science courses. These positions are normally filled during the first week of the semester. Application forms are available in the general office, EN-2021. These positions are competitive and students are hired and assigned primarily on the basis of their academic record in computer science courses and previous work experience.

In addition to the previously described student assistant positions, senior undergraduate majors sometimes have occasion to be hired by various faculty members or local research institutions. These positions are variable; some can be specifically for a small programming task for a short duration, while others can be major programming projects extending over one or more semesters, with summer work on a full-time basis also being possible. Information on these positions is placed on the electronic mail system as positions become available.

If any student is seeking employment, it is advisable to keep a frequent check on the computer listings as the positions are usually taken quickly.

1.8 Computer Science Student Help Centre

The Department of Computer Science Student Help Centre is located in room EN-2031C of the S. J. Carew Building. A timetable of the hours of operation can be accessed on the department web page at www.mun.ca/computerscience and is posted throughout the department area. During academic terms, our instructional assistants hold scheduled hours to give help with 1000- and 2000-level laboratory courses and student assistants (senior computer science majors) are available who can help with non-laboratory courses at the 2000, 3000 and 4000 level. All staff will review general concepts and address difficulties associated with computer science course work. We strongly encourage students to take advantage of the expertise of the individuals at the Centre.

1.9 Fees

Tuition Fees:

See 2011/2012 university *Calendar*. <http://www.mun.ca/regoff/calendar/>

Books and Supplies:

Students should be prepared to meet an expenditure to cover the cost of textbooks and supplies each semester. Students are cautioned to check edition number and printing date of textbooks and manuals prior to purchasing second-hand books; these books may have become obsolete. Within the Department of Computer Science, there is a charge for use of the laser printers

and the student card must be used for payment. Further information on this service will be given to new students during the first week of classes in each semester.

Damages or Destruction to University Property:

Unwarranted damages or destruction to university property will be charged to the responsible party or parties, and are payable when assessed.

1.10 Job Opportunities and Continued Study

Students graduating from Memorial with either a B.Sc. or a BA general degree, majoring in computer science, have many job opportunities available to them in Newfoundland and Labrador as well as throughout Canada and internationally. There is an ever-increasing demand for programmers and analysts in all aspects of data processing as well as in scientific applications. There are numerous career opportunities in hardware and software design, software and technical support, sales, management and various levels of education.

Students who choose to do an honours degree in computer science have the same job opportunities available to them as do those with a general degree, typically at a higher starting salary, and, in addition, have better opportunities to pursue further study in specialized areas of computer science at the M.Sc. or PhD levels. Many Canadian universities, including Memorial, recommend an honours degree or equivalent in computer science or a closely related discipline for entrance to the M.Sc. program.

1.11 Faculty Listing

Following is a list of faculty members and their areas of interest/research. If you want to get a glimpse of what research in computer science is about, take some time to visit the respective faculty member's web-site or www.mun.ca/computerscience/research. Find out more about the fascination and challenges computer science offers to you.

W. Banzhaf -- artificial life, bio-inspired computing, computational intelligence, genetic programming, self-organization

M. Bartha -- programming language semantics, algebra and category theory in computer science, automata theory, graph theory

D. Batten -- programming languages, computer-aided instruction

E. Brown -- human-computer interaction, hypertext, scientific visualization

S. Bungay -- genetic algorithms, mathematical modelling, numerical methods, optimization techniques, physiological systems, computational chemistry, dynamical systems

R. Byrne -- software architecture for embedded systems

P. Chen -- computer networking, distributed computing, combinatorial optimization, approximation algorithms and heuristics, graph theory

A. Deb -- parallel processing, functional languages, logic programming, dataflow architectures, graph reduction machines, vector machines, parallel garbage collection

A. Fiech -- programming languages, ML, lambda calculus, polymorphism, domain theory

P. Gillard -- computer architecture, VLSI design, computer graphics

M. Gong -- computer graphics, computer vision, image processing

R. Gupta -- e-learning, microcomputer based information systems, database systems, computer science education

O. Hoerber -- information visualization, web search, human-computer interaction, information retrieval, web intelligence

A. Kolokolova -- theoretical computer science, complexity theory, mathematical logic

S.W. Lu -- computer vision, artificial intelligence, pattern recognition, image processing, neural networks

M. Mata-Montero -- theoretical computer science, serial and parallel computational complexity

G. Miminis -- scientific computing, numerical methods in control engineering, numerical methods for vector and parallel architectures

L. Peña-Castillo -- data analysis of large scale “omic” data, bioinformatics, machine learning

J. Shieh -- artificial intelligence, computer vision, knowledge-based systems, pattern recognition, robotics

J. Tang -- database systems, distributed computing, fault-tolerant computing, design and analysis of algorithms, data mining

A. Vardy -- robot navigation, bio-inspired computing

K. Vidyasankar -- distributed computing, database systems, graph theory

C. A. Wang -- design and analysis of algorithms, computational geometry, image processing, robotics, computational biology

T. Wareham -- computational biology, algorithmic design and analysis

T. Yu -- intelligent computing, interactive computing, reservoir modelling and simulation, genetic programming

W. Zuberek -- modelling and evaluation of systems, distributed computing, discrete-event simulation, programming languages, their specifications and processors

1.12 Staff Listing

Instructional Support Staff:

Steven Johnstone, EN-1063 (Laboratory Instructor)

Stephen Anthony, EN-1062

Cindy Milley, EN-1064

Ingrid Verbree-Barnes, EN-1065

Computer Support Staff:

Michael Rayment, EN-1060 (Systems Manager)

Paul Price, EN-1051A

Nolan White, EN-1057

Marian Wissink, EN-1059

Aaron Casey, EN-1051A

Tom Boland, HH-3059B, Mathematics and Statistics

Kerri Green, HH-3059A, Mathematics and Statistics

Internship Contact:

Inquire at Division of Co-operative Education, EN-2021 or EN-2025.

Our General Office is located in the S.J. Carew Building, room EN-2021 and is staffed by:

Regina Edwards

Darlene Oliver

Sharon Deir

1.13 Computing Resources

As students progress through their degree program, they are exposed to a wide variety of computing resources and environments which allow them to gain experience that will be a valuable asset to whatever career path they choose.

Each first-year student who takes an introductory course in computer science will receive a LabNet account that will provide access to any of the more than 300 PCs in the following labs:

- CS-1019 (teaching/general access) Computing Services Building
- C-2003 (teaching) Chemistry-Physics Building
- CS-1009 (teaching) Computing Services Building
- Commons (general access) MUN Library

This LabNet account will remain with the students during their academic careers, and as the students advance through the program, the accounts will be extended to allow access to further, more specialized, computing resources. These accounts provide general access to Microsoft Office, and Visual Basic, as well as web browsers and electronic mail. Students can also access their accounts from home using ssh (secure shell). Additionally, students who have their own personal computer will be able to avail of LabNet printing resources and file sharing resources through the university's wireless network using "Nomad". Students can author their personal web pages that are accessible via the URL:

<http://www.pcg labs.mun.ca/~<username>>

Student files are served by an **IBM** server connected to a **SAN** with a capacity of 4400 gigabytes for student files. LabNet serves a user community of about 15,000, providing each user with a 500 megabyte personal disk quota automatically backed up on a daily basis. All labs are equipped with up-to-date desktop computers with at least one gigabyte of memory and a 100 megabit ethernet LAN connection. The computers automatically boot diskless Linux but by choosing Windows from the session manager, the computers will reboot into a Windows XP environment after authentication. This gives students access to both Windows XP or our Gentoo Linux distribution.

As students enter their third year, programming projects and assignments are carried out primarily in EN-2036. These systems boot disklessly from redundant application servers. A computer will be supplied for those students progressing into the graduate degree program.

The department has a 9 node computing cluster for distributed computing applications. Each node has 2 Quad Core Xeon processors, 24 gigabytes of memory, and a Nvidia Tesla 1060 GPU computing card with 240 stream processors and 4 gigabytes of memory.

The Linux Operating System is a free operating system developed and maintained by computer professionals throughout the world via the Internet. Our Linux PCs support the very popular Kde, Gnome or Xfce desktop manager, as well as a complete software development environment, including such languages as C, C++, Lisp, Fortran, Java and Objective C. Many of these languages can be debugged under the general purpose interactive debugger gdb with its many GUI frontends such as kgdb. The Linux system is ideal for carrying out research in computer science because the vast majority of software that comes with the system includes source code so that students can compile their own version of the operating system or extend the functionality of the various software packages.

Since the software is free, students are encouraged to load Linux onto a partition of their personal PCs so that they can work in the comfort of their home. As an added bonus, students with

their own Linux box learn system administration skills which will be invaluable in the competitive job market.

Over and above the application and home directory servers, the department has a number of specialized servers. **mirror** is a server dedicated to mirroring Linux distributions and providing anonymous ftp for the department so students can avail of our network bandwidth while downloading Linux software. **stretch** is our departmental web server and **terra** is our SVN source code repository.

Senior students with an interest in computer hardware can avail of our Micro Electronics Lab, a joint project with the Department of Computer Science, the Faculty of Engineering and Applied Science and the Canadian Microelectronics Corporation (CMC). Students get to design hardware circuitry with the state of the art system on chip design environments listed below:

- Xilinx FPGA system level prototyping for multi media development
- Altera system level prototyping for digital signal processing (DSP)
- Altera system level prototyping for Embedded systems, with NIOS processors
- ARM/Xilinx rapid prototyping platform

Students doing the microprocessor design course will be assigned space in our special projects lab to assemble and test microprocessor embedded control systems. This same lab is used by students enrolled in our robotics course where they will program robotic devices to perform various autonomous activities. Another special projects lab outfitted with Macs is used for teaching students how to program applications on Apples Itouch platform.

Within the Department of Computer Science most computers are running Linux with kde as the default window manager software. A number of software applications are available including some of the following:

- A wide range of high-level languages (C, C++, Java, Mono(.net), Fortran, Lisp, Lucid, Prolog, Python, Miranda, Icon, Scheme, Perl, Tcl, Tk, etc.)
- PC lab software consisting of (available only in labs with appropriate licenses):
 - Visual Basic
 - Matlab
 - Mozilla Firefox, Internet Explorer
 - Microsoft Office including Word, Excel, Access, PowerPoint
- MYSQL Database System
- The SPICE circuit analysis program, Leap Frog vhdl simulator, and the Cadence chip design package
- The OCTAVE, R+ and MATLAB mathematical and statistical computational packages
- Raster image manipulation software consisting of:

- gimp, with access to departmental scanner
- PBMPLUS from MIT
- Khoros from the University of New Mexico
- Distributed parallel computing
 - PVM (Parallel Virtual Machine)
 - MPI (Message Passing Interface)
- Software Engineering
 - Eclipse
 - Umbrello
- Robotic prototyping and simulation software
 - Webots
- Microsoft developer network academic alliance (MSDNAA)
 - Students enrolled in a computer science course can download software from MSDNAA.
- Multimedia software including applications such as gimp (image manipulation program), k3b CD burner program, xine video DVD and other codex viewers, audacity audio editor.
- Open Office suite including presenter, database, spreadsheet, draw and text editing programs.
- Text typesetting facilities through TEX and associated utilities.
- Highspeed long haul network access to other Canadian research institutes through CA*NET.
- Internet software:
 - kmail, thunderbird, and pine mailers
 - ftp, web browsers such as Firefox, Konqueror

In addition to the facilities offered by the department, the Department of Computing and Communications has the following resources (in the Queen Elizabeth Library):

- Digital Media Center with the following facilities:

- access to Macs and PCs
- b/w and colour scanners
- HP colour Laserjet printer
- CD Rom Mastering System
- graphics software for preparing diagrams and slides
- OCR software
- video and audio editing equipment

Computer Access

Workstations for general use are located in several areas: the Chemistry-Physics Building and the Queen Elizabeth II Library. The hours of operation will be posted on the doors of the various buildings. Students are reminded that they should not be in these buildings outside normal hours unless they have received written permission from the head of the department with a copy of such to the security office.

Specialized terminal areas and laboratories for computer science students are located in the following rooms of the Engineering Building:

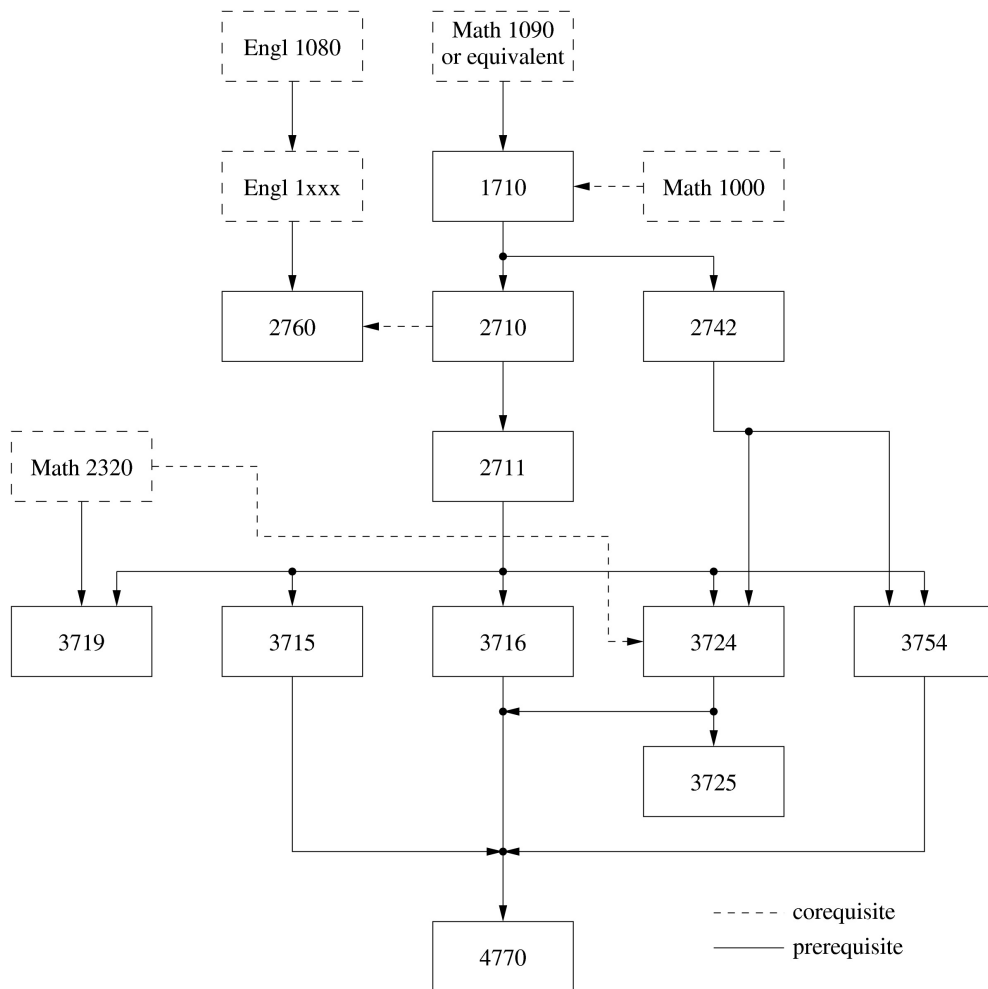
| | |
|---------|---------------------------------|
| EN-2036 | Student Lab |
| EN-2013 | VLSI Lab |
| EN-1049 | Microprocessor and Robotics Lab |

Students living in residence also have access to their computer accounts from computers located in various computer rooms in the residences.

2 Basic Structural Information

This section describes the structural relationship of computer science courses and the requirements leading towards the fulfilment of an undergraduate computer science degree.

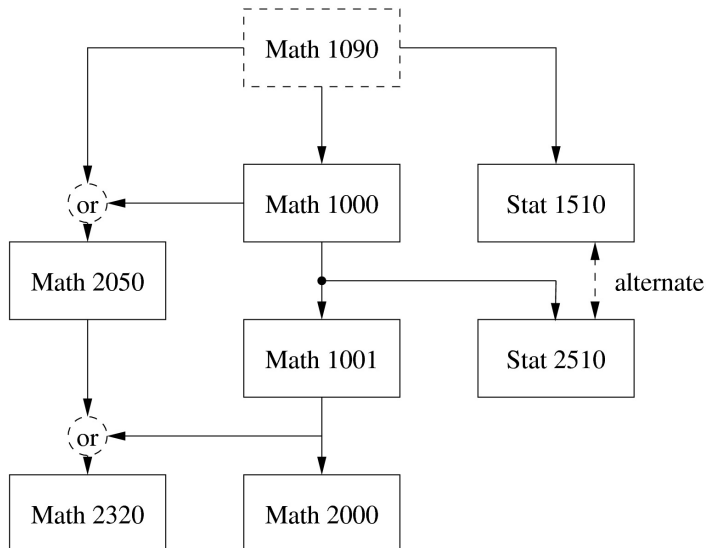
2.1 Prerequisite Structure of Core Courses



Note:

- Students are advised to check the specific prerequisite(s) of each elective course.

2.2 Prerequisite Structure of Required Mathematics Courses



- * Math 1001 **or** Math 2050 is a prerequisite for Math 2320.
- * Math 1000 and Math 2050 can be taken concurrently (with appropriate prerequisites).

2.3 Course Offerings

A selection of the following computer science courses is normally offered during the academic year - semester designations are tentative at the time of printing this handbook. Students should consult the departmental web-site, the postings on the bulletin board outside the general office or the general office staff, room EN-2021, to determine special topics courses to be offered in any given semester.

COMP 1510 - An Introduction to Programming for Scientific Computing

COMP 1550 - Introduction to Multimedia Application Development

COMP 1600 - Basic Computing and Information Technology

COMP 1700 - Introduction to Computer Science

COMP 1710 - Object-Oriented Programming I

COMP 2000 - Collaborative and Emergent Behaviour

COMP 2500 - Data Analysis with Scripting Languages

COMP 2602 - Computer Programming in Fortran

COMP 2650 - Problem Solving with Personal Computers

COMP 2710 - Object-Oriented Programming II

COMP 2711 - Introduction to Algorithms and Data Structures

COMP 2742 - Logic for Computer Science

COMP 2760 - Encountering the Computer: Society and the Individual

COMP 2801 - Introductory Computing for Business

COMP 3550 - Introduction to Bioinformatics
COMP 3700 - Industrial Experience
COMP 3710 - Vocational Languages
COMP 3714 - Programming Languages and their Processors
COMP 3715 - Network Computing with WEB Applications
COMP 3716 - Software Methodology
COMP 3718 - Programming in the Small
COMP 3719 - Theory of Computation and Algorithms
COMP 3724 - Computer Organization
COMP 3725 - Computer Architecture and Operating Systems
COMP 3731 - Introduction to Scientific Computing
COMP 3753 - Computational Aspects of Linear Programming
COMP 3754 - Introduction to Information and Intelligent Systems
COMP 4711 - Structure of Programming Languages
COMP 4712 - Compiler Construction
COMP 4715 - Special Topics in Programming Languages
COMP 4718 - Survey of Software Engineering
COMP 4721 - Operating Systems
COMP 4723 - Introduction to Microprocessors
COMP 4726 to 4729 - Special Topics in Computer Systems
COMP 4734 - Matrix Computations and Applications
COMP 4736 to 4739 - Special Topics in Numerical Computations
COMP 4740 - Design and Analysis of Algorithms
COMP 4741 - Formal Languages and Computability
COMP 4742 - Computational Complexity
COMP 4743 - Graph Algorithms and Combinatorial Optimization
COMP 4745 to 4749 (excluding 4748) - Special Topics in Theoretical Aspects
COMP 4751 - Computer Graphics
COMP 4752 - Introduction to Computational Intelligence
COMP 4753 - Artificial Intelligence
COMP 4754 - Database Systems
COMP 4756 - Image Processing
COMP 4759 - Computer Networks
COMP 4762 - Introduction to Computational Molecular Biology
COMP 4766 - Introduction to Autonomous Robotics
COMP 4767 - Information Visualization and Applications
COMP 4768 - Software Development for Mobile Devices
COMP 4770 - Team Project
COMP 4780 - Honours Project
COMP 4800 to 4825 - Special Topics

The following is a selection of special topics courses which have been offered in the past:

COMP 4715 - Formal Specification of Programming Languages
COMP 4726 - Performance Evaluation of Computer Systems
COMP 4727 - Elements of Parallel Computation
COMP 4736 - Numerical Computations
COMP 4737 - Numerical Algorithms for Supercomputers
COMP 4745 - Introduction to Computational Geometry
COMP 4746 - Distributed Computing
COMP 4747 - The Complexity Class NP
COMP 4748 - Introduction to the Science of Complexity
COMP 4755 - Information Retrieval
COMP 4760 - Pattern Recognition and Neural Networks

2.4 Course Numbering Scheme

There are five areas of computer science offered in the 3000 and 4000 level computer science courses. The meaning of the third digit of a course number indicates a particular area of computer science as follows:

- 1** Programming languages
- 2** Computer systems
- 3** Numerical computations
- 4** Theoretical aspects
- 5 or 6** Applications (e.g., artificial intelligence, computer graphics, database systems, robotics, computational geometry, image processing, computer networking, computer-aided design, computational intelligence, information visualization)
- 7** Project course
- 8** Honours project
- 9** Directed readings

2.5 Undergraduate Programs involving Computer Science

2.5.1 Major in Computer Science (BA or B.Sc.)

1. See General Regulations (BA or B.Sc., as appropriate) in university *Calendar*.
2. Forty-five credit hours in computer science courses are required for a major:
 - (a) Computer Science 1710, 2710, 2711, 2742, 2760, 3715, 3716, 3719, 3724, 3725, 3754 and 4770.
 - (b) At least six additional credit hours at the 4000 level in computer science courses.

(c) Three additional credit hours in a computer science course at the 3000 level or beyond.

- Additional courses required of the major are Mathematics 2000, 2050, 2320 and Statistics 1510 or 2510.

Notes:

- Students are encouraged to take Business 2000, Mathematics 3000 and Statistics 2560.
- A computer science major is encouraged to take a concentration of courses in another discipline in order to gain a broader background. An optional minor is available with a bachelor of science degree.

2.5.2 Honours in Computer Science (BA or B.Sc.)

Students must make a separate application for entrance into an honours program. This is normally done at the beginning of the third year of studies. Acceptance into an honours program is based on a student's academic performance in the computer science courses taken up to the time of application.

1. In order to graduate with an Honours degree, a candidate shall obtain:

(i) A grade of "B" or better, OR an average of 75% or higher (whichever is to the candidate's advantage) in the minimum number of courses in the Honours subject (or subjects) prescribed by the Department (or, in the case of joint Honours, Departments) concerned, excluding the 1000-level courses.

AND

(ii) An average of at least 2.75 points on the total number of courses required for the degree.

Note: Students who wish to fulfil the requirements of Clause (i) above using repeated or substituted courses must obtain approval of the Head of the Department and the Committee on Undergraduate Studies. No more than three such repeated or substituted courses will be permitted.

See also General Regulations for Honours Degree (BA or B.Sc., as appropriate) in university *Calendar*.

2. Sixty-three credit hours in computer science courses are required for an honours degree:

(a) Computer Science 1710, 2710, 2711, 2742, 2760, 3715, 3716, 3719, 3724, 3725, 3754, 4770 and 4780.

- (b) Eighteen additional credit hours in computer science at the 4000 level.
- (c) Six additional credit hours in computer science at the 3000 level or beyond.

3. Additional courses required of the honours student are Mathematics 2000, 2050, 2320, and Statistics 1510 or 2510.

Notes:

- Students are encouraged to take Business 2000, Mathematics 3000 and Statistics 2560.
- A computer science honours student is encouraged to take a concentration of courses in another discipline in order to gain a broader background. An optional minor is available with a bachelor of science degree.

2.5.3 Honours in Computer Science (Software Engineering) (B.Sc. Only)

Completion of the honours in computer science (software engineering) program does not qualify persons to hold the designation "Professional Engineer" as defined by various provincial acts governing the engineering profession.

1. See Section (1) under **Honours in Computer Science (BA or B.Sc.)**.
2. Sixty-three credit hours in computer science courses are required for the honours degree in computer science (software engineering), including:
 - (a) Computer Science 1710, 2710, 2711, 2742, 2760, 3715, 3716, 3718, 3719, 3724, 3725, 3754, 4719, 4759, 4770 and 4780.
 - (b) Twelve additional credit hours in computer science courses at the 4000 level.
 - (c) Three additional credit hours in computer science courses at the 3000 level or beyond.
3. Additional courses required are: Mathematics 2000, 2320, 2050, and Statistics 1510 or 2510 .

Note:

- The honours project (Computer Science 4780) must be in the area of software engineering.

2.5.4 Computer Industry Internship Option (CIIO)

The CIIO provides an opportunity for qualified students to obtain rewarding placements that help them develop practical skills in a real work setting before graduation. The CIIO is available to computer science majors (B.Sc. and B.Sc. Honours only) who will typically apply between their third and fourth year of studies.

Admission Requirements

In order to be considered for admission to the CIIO, an applicant must:

- be a declared computer science major
- have successfully completed Computer Science 1710, 2710, 2711, 2742, 2760, 3716 and any two other core 3000-level computer science courses (selected from 3715, 3719, 3724, 3754, 3725), and
- have at least one computer science course left to complete after the internship.

Note: Admission to the CIIO is also subject to academic performance.

Internship Duration

Subject to the availability of job openings, a student may choose either an 8, 12 or 16 consecutive month internship period.

Internship Guidelines

- Internship employment is normally organized by the Co-op Education Coordinator (hereafter referred to as the Coordinator); however, students who have been accepted to the CIIO may also obtain their own internship placements. All placements are subject to the approval of the Coordinator and of the Head of the Department of Computer Science.
- Students who have applied to the internship program give permission to the Coordinator to supply prospective employers with copies of their resume and transcript.
- After being placed with an employer, students are not permitted to drop their internship without prior approval from the Coordinator and the Head of the Department of Computer Science. Students who drop an internship without permission, who fail to honour an agreement to work with an employer, or who conduct themselves in such a manner as to cause their discharge from the placements, will normally be awarded a fail grade for the internship period and may not be permitted to reapply.

Note: Students should also refer to the University Regulations - General Academic Regulations (Undergraduate).

Expectation of Work

Within a month of starting the internship, students are required to submit a list of their work term objectives. They are also required to submit a **progress report** due the last day of classes of each semester in which they are working. The work term objectives and progress reports are to be submitted to the Coordinator.

At the end of the internship period, students are required to submit a **final report** which will include a description of their internship projects and activities as well as their original objectives and accomplishments. The final report is to be submitted to the Coordinator by the last day of classes

of the semester in question. A completed Employer Evaluation form should be submitted to the Coordinator at the end of the internship period.

Registration, Assessment of Performance, and Assignment of Grades

Students must register for the course Computer Science 3700 every semester during their internship. Computer Science 3700 is a non-credit course open only to students who have been accepted into the internship program.

During the internship the employer and intern will complete student performance evaluations every four months and will submit them to the Coordinator. The final assessment of total work performed is the responsibility of the Coordinator, and will be based upon both input from the employer and the intern's final report.

At the end of the internship, each intern will be assigned one of the following grades after the final assessment of their performance:

- (a) *Pass with Distinction (PWD)*: Indicates EXCELLENT PERFORMANCE in both the work report and work performance.
- (b) *Pass (PAS)*: Indicates that PERFORMANCE MEETS EXPECTATIONS in both the work report and the work performance.
- (c) *Fail (FAL)*: Indicates FAILING PERFORMANCE in the work report or the work performance.

Also, the following will be noted on the transcript of the intern:

- Requirements for the Computer Industry Internship Option have been completed.
Internship Duration: - months.
- A grade of NC (No Credit) for Computer Science 3700 will be awarded in all semesters of the internship option prior to the final semester.

CIIO and Honours Program

In case a student is enrolled in both the honours program and the CIIO, the requirements of both must be met. Upon approval from the honours project supervisor, within the department, the employer and the head of the Department of Computer Science, an internship project may be submitted as a component of an honours project. These arrangements must be made within the first semester of the internship placement.

2.5.5 Minor in Computer Science (BA or B.Sc.)

For a minor in computer science, a student must complete at least 24 credit hours in computer science courses, including:

1. Computer Science 1710, 2710, 2711.
2. At least six credit hours selected from Computer Science 3715, 3716, 3719, 3724, 3725 and 3754.
3. Three additional credit hours in computer science courses at the 3000 level or above.

2.5.6 Computer Science and Statistics Joint Major (B.Sc. Only)

In addition to Mathematics 1000, 1001 and Computer Science 1710, the following courses numbered 2000 or higher are required:

- (a) Computer Science 2710, 2711, 2742, 2760, 3715, 3716, 3719, 3724, 3725, 3754, 4734 and 4770.
- (b) Statistics 2510 and 2560.
- (c) Mathematics 2000, 2050, 2051, 2320, 3340, Statistics 3410, 3411, 3520, 3521, 3540, and 4590.
- (d) Nine further credit hours in statistics courses numbered 3000 or higher including at least a three credit hour course numbered 4000 or higher, excluding Statistics 4581.

2.5.7 Computer Science and Statistics Joint Honours (B.Sc. Only)

1. See General Regulations for the Honours Degree of Bachelor of Science in university *Calendar*.
2. In addition to Mathematics 1000 and 1001, the following courses numbered 2000 or higher are required:
 - (a) Mathematics 2000, 2050, 2051, 2320, 3340, Statistics 3410, 3411, 3520, 3521, 3530, 3540 and 4590.
 - (b) Twenty-four further credit hours in statistics courses including at least 12 credit hours in courses numbered 4000 or higher, but not including Statistics 4599 and 4581.
 - (c) Computer Science 1710, 2710, 2711, 2742, 2760, 3715, 3716, 3719, 3724, 3725, 3754, 4734 and 4770.

- (d) Six additional credit hours in computer science courses at the 4000 level, not including Computer Science 4780.
- (e) Either Computer Science 4780 or Statistics 4599.

2.5.8 Computer Science and Pure Mathematics Joint Major (B.Sc. Only)

In addition to Mathematics 1000, 1001 and Computer Science 1710, the following courses numbered 2000 or higher are required:

- (a) Computer Science 2710, 2711, 2742, 2760, 3715, 3716, 3719, 3724, 3725, 3754 and 4770.
- (b) Six additional credit hours in computer science numbered 3000 or higher.
- (c) Mathematics 2000, 2050, 2051, 2130, 2320, 3000, 3202, 3260, 3320, 3340 and Statistics 2510.
- (d) Nine additional credit hours in courses numbered 3000 or higher offered by the Department of Mathematics and Statistics.

2.5.9 Computer Science and Pure Mathematics Joint Honours (B.Sc. Only)

1. See General Regulations for the Honours Degree of Bachelor of Science in university *Calendar*.
2. At least 51 credit hours in computer science are required including the following:
 - (a) Computer Science 1710, 2710, 2711, 2742, 2760, 3715, 3716, 3719, 3724, 3725, 3754 and 4770.
 - (b) Excluding Computer Science 4780, 15 additional credit hours from courses numbered 3000 or higher, at least nine credit hours of which must be in courses at the 4000 level.
3. The following courses in mathematics and statistics are required:
 - (a) Mathematics 1000, 1001, 2000, 2050, 2051, 2130, 2320, 3000, 3001, 3202, 3210, 3260, 3320, 3340, Statistics 2510;
 - (b) Either Mathematics 4000 or 4001;
 - (c) Excluding Mathematics 4399, 15 additional credit hours in courses offered by the Department of Mathematics and Statistics numbered 3000 or higher including at least nine credit hours from courses numbered 4000 or higher and at least nine credit hours in pure mathematics courses;

- (d) An honours dissertation in one of the departments, with the topic chosen in consultation with both departments.

2.5.10 Computer Science and Applied Mathematics Joint Major (B.Sc. Only)

The following courses are required:

- (a) Computer Science 1710, 2710, 2711, 2742, 2760, 3715, 3716, 3719, 3724, 3725, 3754, 4770, plus six further credit hours in computer science courses numbered 3000 or higher.
- (b) Mathematics 1000, 1001, 2000, 2050, 2051, 2130, 2320, 3000, 3100, 3132, 3161, 3202, 3260, 4160, 4190, plus one of Mathematics 3210, 4131, 4132, 4162.

In addition, Statistics 2510 is highly recommended.

2.5.11 Computer Science and Geography Joint Major (B.Sc. Only)

Students shall complete the following requirements:

1. Computer science requirements: Thirty-nine credit hours in computer science courses are required: Computer Science 1710, 2710, 2711, 2742, 2760, 3715, 3716, 3719, 3724, 3725, 3754, 4751 and 4770.
2. Geography requirements: Thirty-nine credit hours in geography courses are required: Geography 1050, 2001, 2102, 2195, 2302, 2425, 3202, 3222, 3250, 3260, 4202, 4250, 4261.
3. Additional requirements: Mathematics 2000, 2050 and 2320.

2.5.12 Computer Science and Geography Joint Honours (B.Sc. Only)

1. See General Regulations for the Honours Degree of Bachelor of Science in university *Calendar*.
2. Forty-eight credit hours in computer science courses are required for the joint honours:
 - (a) Computer Science 1710, 2710, 2711, 2742, 2760, 3715, 3716, 3719, 3724, 3725, 3754, 4751 and 4770.
 - (b) Six additional credit hours in courses at the 4000 level not including Computer Science 4780.
 - (c) Three additional credit hours in courses at the 3000 level or beyond.

3. A total of 51 credit hours in geography is required for the joint honours:
 - (a) Geography 1050, 2001, 2102, 2195, 2226, 2302, 2425, 3202, 3222, 3226, 3250, 3260, 3303, 4202, 4250, 4261 and the former 4291.
4. Additional requirements:
 - (a) Mathematics 2000, 2050 and 2320.
 - (b) An honours dissertation (either Computer Science 4780 or Geography 4999). The topic for dissertation must be chosen with the prior approval of the heads of both departments.

2.5.13 Computer Science and Physics Joint Major (B.Sc.)

1. Chemistry 1050 and 1051 (or Chemistry 1010, 1011 and 1031).
2. Computer science requirements: Thirty-nine credit hours in computer science are required for the joint major: Computer Science 1710, 2710, 2711, 2742, 2760, 3715, 3716, 3719, 3724, 3725, 3731, 3754 and 4770.
3. Physics requirements: Physics 1050 (or 1020) and 1051 plus at least 30 additional credit hours in physics including: Physics 2053, 2055, 2750 (or 2056), 2820, 3220, 3400, 3500, 3550, 3750 and 3900.
4. Additional requirements: Mathematics 1000 and 1001, Mathematics 2000, 2050, 2320, 3260 and Statistics 2510 and Physics 3810 or Mathematics 3202.

2.5.14 Computer Science and Physics Joint Honours (B.Sc.)

The following courses are prescribed:

1. Chemistry 1050 and 1051 (or Chemistry 1010, 1011, and 1031).
2.
 - (a) Computer Science 1710, 2710, 2711, 2742, 2760, 3715, 3716, 3719, 3724, 3725, 3731, 3754 and 4770.
 - (b) Nine additional credit hours in computer science numbered 3000 or higher, including at least six credit hours in courses numbered 4000 or higher.
3.
 - (a) Physics 1050 (or 1020) and 1051.
 - (b) Physics 2053, 2055, 2750 or 2056, 2820, 3220, 3400, 3500, 3750, 3820, 4500, 4820, and 3230 or 3900.
4. Physics 490A/B or Computer Science 4780.

5. Physics 3810 or Mathematics 3202.
6. (a) Mathematics 1000 and 1001.
(b) Mathematics 2000, 2050, 2320 and 3260.

Statistics 2510 is recommended.

The topic for the honours project or thesis, Computer Science 4780 or Physics 490A/B, must be chosen with the prior approval of both departments.

2.5.15 Computer Science and Economics Joint Major (B.Sc. Only)

The following courses are prescribed:

1. Computer Science requirements:

Forty-two credit hours in computer science courses are required: Computer Science 1710, 2710, 2711, 2742, 2760, 3715, 3716, 3719, 3724, 3725, 3731, 3753, 3754, and 4770.

2. Economics requirements:

A total of 42 credit hours in economics courses is required: Economics 2010, 2020, 2550, 3000, 3001, 3010, and six credit hours from either 3550 and 3551, or 4550 and 4551 are obligatory.

The remaining 18 credit hours shall be chosen from among the various economics courses in consultation with the head of the department or delegate, and will include at least nine credit hours in courses at the 4000-level.

3. Additional requirements:

Mathematics 1000, 1001, 2000, 2050, 2320 and Statistics 2510.

2.5.16 Bachelor of Commerce (Co-operative) Concentrations

(See university *Calendar*, Faculty of Business Administration Program Regulations for further information.)

Information Systems

Students electing an information systems concentration should complete the following courses:

1. Six of Business 5700, 5701, 5702, 5703, 6700, 6701, 7700, and 7701,
and either
2. Geography 2195, 3260, and one of 3202, 4202, 4261
or
2. three of the following: Computer Science 1710, 2500, 2710, 2760, 3710, 3715, 4761, 4767.

Operational Research

Students electing an Operational Research concentration should complete the following eight courses:

1. Business 5401, 5402, 6400, 7400; and
2. any four of: Computer Science 1710, Computer Science 2710, Mathematics 1001, Mathematics 2050, and any Business Information Systems course (or courses) at the 5000 level or above.

2.5.17 Joint Degrees of Bachelor of Arts and Bachelor of Commerce (Co-operative)

Any student who is admitted into the Bachelor of Commerce (Co-operative) program may simultaneously complete the requirements for a Bachelor of Arts program. The Major or Minor program for the Bachelor of Arts can be Computer Science. See the university *Calendar*, Faculty of Arts or Faculty of Business Administration Degree regulations for further information.

2.6 General Degree Regulations

Students are encouraged to familiarize themselves with the general degree requirements (as printed in the *Calendar*) pertaining to their chosen faculty (arts or science).

In order to graduate within the Faculty of Science a candidate shall have:

- (a) satisfied the conditions of General Academic Regulations (Undergraduate);
- (b) obtained an average of at least 2.0 points per credit hour in the 78 credit hours in Science required for the degree;
- (c) obtained an average of at least 2.0 points per credit hour in the minimum number of credit hours in the major subject (or, in the case of joint majors, subjects) required for the major program (or, in the case of joint majors, programs).

In order to graduate with the General Degree of Bachelor of Arts, a candidate shall obtain:

- (a) an average of 60% or higher on the minimum number of courses prescribed for the Major program, excluding 1000-level courses, and
- (b) an average of 60% or higher on the minimum number of courses prescribed for the Minor program, excluding 1000-level courses, and
- (c) an average of 2.0 points or higher per credit hour on the courses in Arts disciplines.

2.7 Second Degree in Computer Science

A student can receive a bachelor's degree in computer science from Memorial, even if they already have a bachelor's degree in another subject from Memorial. In order to receive a second degree majoring in computer science, a student must consider the following regulations:

- (a) Section 5.2.3 of the General Academic Regulations (Undergraduate) of the university *Calendar* states: "A student will not be awarded the same bachelor's degree more than once by this university." Thus if a student already has a Bachelor of Science degree, then he/she must complete the computer science degree requirements for a Bachelor of Arts degree. If he/she already has a Bachelor of Arts degree, then he/she must complete the requirements for a Bachelor of Science degree. A student can also convert a general degree into an honours degree (e.g., B.Sc to B.Sc Honours).
- (b) Section 5.3.3 of the General Academic Regulations (Undergraduate) of the university *Calendar* states: "Every student for a second bachelor's degree shall complete at least 30 credit hours at this university beyond those required for the first degree. These credits must be applicable to the degree sought." This means that a second degree can be obtained with a minimum of 30 extra credit hours. However, meeting the requirements for the computer science degree can result in taking more than 30 extra credit hours.

Any student wishing to take computer science as a second degree should consult with the Department of Computer Science to ensure that all regulations can be met as efficiently as possible.

2.8 Deferred Exams

(a) Final Examinations

Please refer to Section 5.7.3 of the General Academic Regulations (Undergraduate) of the university *Calendar*.

(b) Midterms and Term Tests

Please note that deferred midterms and term tests are at the discretion of the instructor of each course.

2.9 Supplementary Exams

(a) Supplementary examinations will be allowed in certain of the computer science courses which have written final examinations. In each course, students will be informed as to the possibility of a supplementary examination during the first week of classes. This information will be provided in writing, as part of the evaluation scheme for the course.

(b) Supplementary examinations will be similar in length and degree of difficulty as the original final examination.

(c) Students who wish to write supplementary examinations must apply in writing to the department within one week of release of grades. Forms are available in the general office (EN-2021).

(d) A student who has clear or conditional standing may write a supplementary examination in a course if the final grade obtained is 45-49F and if his or her term mark is at least 50%.

(e) In order to pass the course, the student must pass the supplementary examination. If the student passes the supplementary examination, then a new grade will be calculated using the same weighting scheme as used in the course, but with the result of the supplementary examination replacing that of the original final examination. Any additional course requirements, including a requirement to pass the laboratory component of a course, will continue to apply.

(f) If the new course grade is higher than the original, it will replace the original grade on the student's transcript, *subject to the condition that the final mark will not exceed the student's term mark*. The student's transcript will indicate that the course result was earned as the result of a supplementary examination.

- (g) Supplementary examinations will be written no later than the first week of the semester immediately following the one in which the course was failed. Normally they will coincide with the writing of deferred examinations. Grades for supplementary examinations will be submitted to the Office of the Registrar within one week following the commencement of classes for that semester.
- (h) A student may write a supplementary examination for any one registration in a course only once; if the course result following the supplementary examination is a fail, then the course must be repeated in order to obtain credit.

3 Description of Undergraduate Courses

Throughout the following sections, an asterisk * placed after a course number indicates that there are other prerequisites. Parenthesized characters with the course number indicate the semester(s) in which the particular course is **usually** offered: F(fall), W(winter), S(spring).

3.1 1000-Level Courses

| | |
|-----------|---|
| COMP 1510 | An Introduction to Programming for Scientific Computing |
| COMP 1550 | Introduction to Multimedia Application Development |
| COMP 1600 | Basic Computing and Information Technology |
| COMP 1700 | Introduction to Computer Science |
| COMP 1710 | Object-Oriented Programming I |

COMP 1510 (F, W)

An Introduction to Programming for Scientific Computing

Students Interested

Those who are interested in learning a programming language, as well as various numerical methods relevant to scientific computing. Numerical methods to solve selected problems from Physics, Chemistry and Mathematics will be covered.

Objectives of the Course

To introduce students to basic programming in the context of numerical methods, with the goal of providing the foundation necessary to handle larger scientific programming projects.

Prerequisite(s)

Mathematics 1000

Successor(s)

COMP 2500, Mathematics 2130*

Representative Workload

| | |
|---------------------------|-----|
| Lab Quizzes / Assignments | 40% |
| Midterm | 30% |
| Final Exam | 30% |

Representative Course Outline

- Computer terminology and fundamental concepts, problem solving, floating point arithmetic
- Programming in Fortran 90
 - data types, expressions, I/O, formatted I/O, if statement, logical operators, loops, arrays, subprograms
- Numerical methods to solve selected problems from Physics, Chemistry, and Mathematics.
- Programming in C
 - data types, expressions, I/O, formatted I/O, if statement, logical operators, loops, arrays, subprograms

Comments or Notes

- Students can receive credit for only one of Computer Science 1510 or Computer Science 2602.

- Students who have received credit for the former Applied Mathematics 2120 cannot receive credit for Computer Science 1510.
- In addition to three one-hour lectures, there is a structured laboratory as scheduled in the university timetable. Laboratory sections will meet for the first **two and one-half** hours of the laboratory slots. **Attendance at these laboratory sessions is compulsory.**

COMP 1550 (F)

Introduction to Multimedia Application Development

Students Interested

In this course multimedia is used as the subject of learning as part of an introductory course to programming, with ActionScript used as the programming language. The course is intended to appeal to students who would not normally be interested in taking core courses using the Java language (which form the standard path for Computer Science majors). Students doing a Bachelor of Arts degree may find this course of interest, as well as other students interested in learning about programming computers. The course emphasizes the concept of immediate applicability of the programming concepts introduced, particularly for introductory multimedia projects.

Objectives of the Course

This course is an introduction to programming and computer science with an emphasis on the development of multimedia applications. The course introduces the fundamental principles of programming, including object-oriented and event-driven programming. Successful students will understand how to use and create classes and methods and combine them with multimedia libraries to produce animations, handle input from keyboard and mouse, and import sounds and videos to produce multimedia applications which can be directly deployed on the Internet.

Prerequisite(s)

None

Successor(s)

None

Representative Workload

| | |
|--------------------------|-------|
| Assignments and Lab Work | 30% |
| Test 1 | 12.5% |
| Test 2 | 12.5% |
| Final Project | 20% |
| Final Exam | 25% |

Representative Course Outline

- Introduction to Computers and Programming
 - definitions of multimedia, digital data fundamentals, multimedia authoring overview
- Flash and ActionScript
 - overview of Flash, ActionScript syntax overview
- Using objects

- getting familiar with variables, using objects, calling methods, using properties and events
- Datatypes
 - primitive datatypes, expressions, strings, string manipulation
- Conditionals
 - if statement, boolean expressions, switch-case statements, nesting conditionals
- Loops
 - while, do and for loops
- Arrays and Functions
 - using arrays, defining functions
- Object References
 - understanding objects vs. object references, calling objects from the library, defining the document class
- Implementing Object Classes
 - implementing methods, understanding constructors, instance fields, and local variables
- Events and Event Handling
 - ActionScript event basics, assessing objects through event handlers, event listeners
- Style and HCI Guidelines for Multimedia
 - fonts, color, user feedback, splash screens, loaders and publishing
- Labs
 - overview: general introduction to the Flash interface and the timeline, getting started with the drawing tools, using symbols and managing the library
 - ActionScript overview, core language fundamentals, creating buttons, basic animation
 - exploring properties, methods, events and advanced built-in animation
 - understanding display lists and timeline control instructions
 - writing functions
 - object-oriented programming
 - programming motion effects
 - drawing vector graphics and raster graphics
 - programming rich text
 - handling sound and video
 - loading assets and publishing

COMP 1600 (F, W, S)

Basic Computing and Information Technology

Students Interested

Any students who will benefit from information management and analysis skills and concepts.

Objectives of the Course

To introduce information management and data analysis using internet, spreadsheet and database technology. The concepts and technologies are typically introduced using applied problems in a business context, but the essential concepts and skills are applicable to any field.

Prerequisite(s)

Level III Advanced Mathematics, or Mathematics 1090 (which can be taken concurrently)

Successor(s)

None

Representative Workload

| | |
|-----------------|-----|
| Assignments (3) | 10% |
| Labs (8) | 10% |
| Midterm Exam | 40% |
| Final Exam | 40% |

Representative Course Outline

- Computing and networks
 - File management
 - File system structure and layout
 - Web and internet computing
 - Network connectivity
 - Security

- Data manipulation with spreadsheets
 - Spreadsheet design
 - Input/output and formulae
 - Cell addressing
 - Graphing
 - Table functions and conditional functions
 - Scenarios and related techniques

- Data manipulation and modelling with databases
 - Client/server review
 - E/R model
 - Primary/foreign keys
 - Report generation
 - Referential integrity, redundancy
 - Query design
 - Normal form
 - SQL and QBE syntax
 - Validation
- Optional Topics
 - Presentation and project integration software
- Labs
 - Lab 1 - Internet and remote services
 - Lab 2 - File systems and data representation
 - Lab 3 - Spreadsheet I
 - Lab 4 - Spreadsheet II
 - Lab 5 - Spreadsheet III
 - Lab 6 - Database I
 - Lab 7 - Database II
 - Lab 8 - Database III

Comments or Notes

- In addition to three one-hour lectures, there is a structured laboratory as scheduled in the university timetable. **Attendance at these laboratory sessions is compulsory.**
- Students can receive credit for only one of Computer Science 1600, Computer Science 2650, Computer Science 2801 or the former Business 2700.

COMP 1700 (F, W)

Introduction to Computer Science

Students Interested

This course is designed for potential computer science majors with little or no background in programming and would also be of interest to anyone who would like to find out what computer science is about.

Objectives of the Course

This course gives students an overview of some important areas of Computer Science including fundamental and topical issues in computers, languages, programming and applications.

Prerequisite(s)

Mathematics 1090 (or equivalent), or Mathematics 1000, either of which may be taken concurrently

Successor(s)

COMP 2500

Representative Workload

| | |
|--------------|-----|
| Assignments | 15% |
| Lab Quizzes | 15% |
| Midterm Exam | 20% |
| Final Exam | 50% |

Representative Course Outline

- Definition of Computer Science

- Introduction to Computer Systems
 - hardware
 - software
 - systems software
 - application software

- Algorithms
 - formal definition of an algorithm
 - representing algorithms
 - control structure: sequential, conditional and iterative operations
 - searching algorithms
 - sorting algorithms
 - efficiency of algorithms

- Hardware
 - number systems
 - boolean logic and gates
 - computer circuits

- Machine architecture
 - memory
 - input/output
 - arithmetic/logic unit
 - control unit
 - machine language

- Assembly language

- Networks

- Programming languages
 - history
 - Python language
 - translation process

- Selected topic(s)

COMP 1710 (F, W)

Object-Oriented Programming I

Students Interested

This course is required for all computer science majors and minors.

Objectives of the Course

This course is an introduction to fundamental programming techniques, primitive data types and operations, program control structures and the use of objects, classes and methods.

Prerequisite(s)

Mathematics 1000 (which can be taken concurrently), or Mathematics 1090 (or equivalent)

Successor(s)

COMP 2500, COMP 2710, COMP 2742, Mathematics 2130*

Representative Workload

| | |
|-------------|-----|
| Assignments | 20% |
| Lab Quizzes | 15% |
| Test(s) | 20% |
| Final Exam | 45% |

Representative Course Outline

- Fundamental programming constructs, including primitive data types, operations, expressions, assignment statements, input, output, selection and looping (13 hours)
- Algorithms and problem solving (7 hours)
- An introduction to object-oriented programming (5 hours)
- Fundamental data structures, including strings, arrays and array lists (5 hours)
- Searching (linear) and sorting (bubble sort) algorithms (2 hours)
- Using APIs (2 hours)

CC2001 modules

PF1 (9/9), PF2 (6/6), PF3 (4/14), PL4 (2/3), PL6 (4/10), AL3 (2/12), SE2 (2/5)

Comments or Notes

- In addition to three weekly one-hour lectures, there is a structured laboratory as scheduled in the university timetable. **Attendance at these laboratory sessions is compulsory.**

- Students who have previously completed Computer Science 2710 will not be permitted to register or receive credit for Computer Science 1710.

3.2 2000-Level Courses

COMP 2000 - Collaborative and Emergent Behaviour
COMP 2500 - Data Analysis with Scripting Languages
COMP 2602 - Computer Programming in Fortran
COMP 2650 - Problem Solving with Personal Computers
COMP 2710 - Object-Oriented Programming II
COMP 2711 - Introduction to Algorithms and Data Structures
COMP 2742 - Logic for Computer Science
COMP 2760 - Encountering the Computer: Society and the Individual
COMP 2801 - Introductory Computing for Business

COMP 2000 (F)

Collaborative and Emergent Behaviour

Students Interested

A grasp of computation as a significant paradigm for understanding both technology and modern models of natural phenomena, and its connection to other fields of human inquiry, is valuable for students generally. Existing courses related to computation are not generally accessible to majors in non-mathematical disciplines, and require extensive background in programming. This course is intended to be generally accessible to undergraduate students, and will be of particular interest to students in the Interdisciplinary Program in Communications.

Objectives of the Course

Collaborative and Emergent Behaviour is a survey of computation as a means of understanding, modeling, and describing artificial and natural systems. The emergence of complex behaviour from the interaction of simple rules governing individual components is illustrated and discussed, as well as the role of communication between system components. Selected systems to be studied will be drawn from different topic areas which may include the worldwide web, the mind (cognitive science), formal logic, autonomous robotics, chaos and fractals, and bioinformatics. Each topic will incorporate an associated laboratory experience.

Prerequisite(s)

None

Successor(s)

None

Representative Workload

| | |
|-----------------|-----|
| Topic Tests (5) | 25% |
| Lab Reports (6) | 25% |
| Final Exam | 50% |

Representative Course Outline

- Introduction
 - Complex systems: Examples of complex systems, concepts of causality, predictability, and determinism in classical natural science, introduction to chaos, randomness, and emergence as properties of dynamical systems.
 - Communication and logic: “Real-world” reasoning versus formal logic, semantic limitations, concept of “truth” and “knowable,” communication, representation, and encoding of information.

- Selected Topic Area: Chaos and fractals
Examples of self-similarity in nature and mathematics, dimensionality in fractal objects, feedback and stability of dynamical systems.
Lab work: Guided exploration of rules for generation of fractal objects and landscapes
- Selected Topic Area: Internet and Mobile Computing
Models of networks (client-server, peer-to-peer, etc.), web technologies and applications, protocols, layers, switching.
Lab Work: Use mobile devices to explore underlying functionality of network (connectivity, authentication, security)
- Selected Topic Area: Bioinformatics
Function and encoding (DNA/RNA/transfer RNA), pattern matching, sequence alignment, gene regulation and metabolism.
Lab Work: Sequence reconstruction and pattern matching (using GENBANK)
- Selected Topic Area: Robotics
Kinematics, perception, and control, swarm intelligence, communication and emergence.
Lab Work 1: Experiment with control parameters for a two-wheeled robot.
Lab Work 2: Experiment with rules for robot co-operation and group behaviour.
- Selected Topic Area: Cognitive Science
Metaphors for mind; mind as computation; limits on computational minds, human problem-solving, artificial intelligence.
Lab Work: Beat the computer at problem solving; can you pass a reverse Turing test?

Comments or Notes

- Students will be expected to attend six bi-weekly three-hour lab sessions and to submit a lab report at the end of each lab.

COMP 2500 (F)

Data Analysis with Scripting Languages

Students Interested

Data analysis is an essential part of many university courses and research projects. Computer systems have become indispensable for many data analysis tasks. Students and researchers can use scripting languages to perform many data analysis tasks more easily than with traditional programming languages. This course is of interest to people who need to readily perform data analysis. Scripting languages have been adopted by many research communities to aid in their research.

Objectives of the Course

This course will provide practical examples of using a scripting language to perform data analysis and data visualization. Common tasks of storing, searching, curve fitting, statistical analysis, plotting and data computation for data sets will be covered. Possible scripting languages include Perl, Python, and Ruby.

Prerequisite(s)

COMP 1510, COMP 1700, COMP 1710 or COMP 2602 (or equivalent)

Successor(s)

COMP 3550*

Representative Workload

| | |
|-------------|-----|
| Assignments | 30% |
| Tests | 30% |
| Final Exam | 40% |

Representative Course Outline

- Control statements and expressions of the scripting language
- Subroutines and modules
- String processing
- Managing data files and sets
- Processing collections of data with lists and dictionaries
- Graphing (plotting) and data visualization
- Statistical data analysis
- Numeric computations with packages
- Finding and using application specific packages (e.g., astronomy, physics, chemistry)

COMP 2602

Computer Programming in Fortran

Students Interested

This course is addressed to students in any discipline who are interested in studying algorithmic-problem solving and structured programming techniques. The Fortran programming language serves in this course as an effective tool for implementation of a computer solution, while promoting a solid programming style. Problems of numeric and non-numeric nature are examined using the most recent versions of software.

Objectives of the Course

The main objective of this course is to expose students to algorithmic-problem solving and to develop fundamental skills in computer programming, with emphasis on a transparent and disciplined programming style, code modularity and reusability of the components.

Prerequisite(s)

Mathematics 1000

Successor(s)

COMP 2500, COMP 3731*, COMP 3753*, Mathematics 2130*

Representative Workload

| | |
|--------------------------|-----|
| Assignments and Projects | 30% |
| In-class Tests | 25% |
| Final Exam | 45% |

Representative Course Outline

- Algorithmic problem solving
- Program development tools
- Programming languages and their qualities
- Computer Programming with Fortran
 - Structure of Fortran programs
 - Data types and declarations
 - Input and output operations
 - Conditional constructs
 - Iterative constructs
 - Format specification
 - Debugging techniques
 - Handling external files
 - Arrays and applications
 - Functions and subroutines

- Recursive procedures
- Character manipulation
- Derived data types
- Introduction to pointers
- Standard Fortran libraries
- Program documentation guidelines
- High performance computing

Comments or Notes

- Students can receive credit for only one of Computer Science 1510 or Computer Science 2602.

COMP 2650 (F, W)

Problem Solving with Personal Computers

Students Interested

Networked personal computers have become an integral part of the home and work environment in the modern era. They are “a primary engine” behind our daily endeavours. In order to fully succeed in academic careers and to meet growing expectations in the workplace, it has become critical to understand principles of computer operation and to utilize computer potential most effectively. This course offers an overview of opportunities that the digital information age presents. It may be of interest to students in any academic discipline.

Objectives of the Course

The main objective of this course is to provide students with proficiency in using personal computers, including an active use of algorithmic thinking and computer programming for solving common problems. This is being done through studying principles of operations of the computer hardware and through conceptual understanding of the computer software, from exploring capabilities of the existing software tools to learning methods of extending these capabilities. The course has a practical flavour.

Prerequisite(s)

Level III Advanced Mathematics, Mathematics 1000 (which can be taken concurrently), or Mathematics 1090 (or equivalent)

Successor(s)

None

Representative Workload

| | |
|----------------------|-----|
| Assignments | 20% |
| Lab Work and Quizzes | 20% |
| Midterm Exam | 20% |
| Final Exam | 40% |

Representative Course Outline

- User Level
 - Essentials of electronic communication
 - Computer networks and the Internet
 - Development of interactive web pages
 - Visual presentation software
 - Graphical display of data
 - Solving problems with electronic spreadsheets
 - Digital media and data representation

- System software and application software
- Computer and Information Processing
- Software Level
 - Fundamental programming constructs
 - Algorithm design and verification
 - Programming in visual environment
 - Enhancing user applications with macros
 - Object linking and embedding
 - Elements of Artificial Intelligence
 - Data Security and control
- Hardware Level
 - Principles of operation of the CPU
 - Machine Language and Assembly Language

Comments or Notes

- In addition to three one-hour lectures, there is a structured laboratory as scheduled in the university timetable. **Attendance at these laboratory sessions is compulsory.**
- Students can receive credit for only one of Computer Science 1600, Computer Science 2650, Computer Science 2801, or the former Business 2700.

COMP 2710 (F, W) Object-Oriented Programming II

Students Interested

This course is required for all computer science majors and minors.

Objectives of the Course

Continuing from Object-Oriented Programming I, this course studies object-oriented and event-driven programming. Additional topics include: recursion, basic analysis of algorithms, fundamental data structures such as simple linked structures and stacks, and fundamental computing algorithms such as binary search and quadratic time sorting. A brief overview of programming languages, virtual machines and language translations is also provided.

Prerequisite(s)

COMP 1710

Successor(s)

COMP 2711, COMP 3550*, COMP 3731*, COMP 3753*, co-requisite for COMP 2760, Mathematics 2130*

Representative Workload

| | |
|--------------|-----|
| Assignments | 20% |
| Lab Quizzes | 15% |
| Midterm Exam | 20% |
| Final Exam | 45% |

Representative Course Outline

- Object-oriented and event-driven programming (12 hours)
- Fundamental data structures - simple linked structures and stacks (6 hours)
- Recursion - recursive structures and recursive methods (3 hours)
- Basic analysis of algorithms - worst-case big-O bounds on running time; analysis of simple recursive methods (2 hours)
- Algorithms - binary search, insertion sort, selection sort, and worst-case running time for each (3 hours)
- Programming Language Overview - programming languages, virtual machines language, language translation, declarations and types (6 hours)

CC2001 modules

PF3 (5/14), PF4 (3/5), PF5 (4/4), PL1 (2/2), PL2 (1/1), PL3 (2/2), PL4 (1/3), PL6 (6/10), AL1 (2/4), AL3 (3/12)

Comments or Notes

- In addition to three weekly one-hour lectures, there is a structured laboratory as scheduled in the university timetable. **Attendance at these laboratory sessions is compulsory.**

COMP 2711 (F, W)

Introduction to Algorithms and Data Structures

Students Interested

This course is required for all computer science majors and minors.

Objectives of the Course

This course includes the study of standard ways of organizing and manipulating data in computer storage. Fundamental concepts in the design and analysis of algorithms are also discussed.

Prerequisite(s)

COMP 2710

Successor(s)

COMP 3710, COMP 3715, COMP 3716, COMP 3718*, COMP 3719*, COMP 3724*,
COMP 3754*, COMP 4766*

Representative Workload

| | |
|--------------|-----|
| Assignments | 20% |
| Lab Quizzes | 15% |
| Midterm Exam | 20% |
| Final Exam | 45% |

Representative Course Outline

- Analysis Tools - Pseudocode, Math Analysis, Asymptotic Notation and Analysis (2 hours)
- Recursion - Recursive Methods and Analysis (2 hours)
- Stacks, Queues, Hash Tables, Graphs, Trees (6 hours)
- Abstraction Mechanisms - procedures, functions and iterators; activation records and storage management (3 hours)
- Algorithmic Strategies - brute-force algorithms, greedy algorithms, divide-and-conquer, backtracking, branch-and-bound, heuristics, pattern matching and string/text algorithms, numerical approximation algorithms (7 hours)
- Fundamental computing algorithms - $O(n \log n)$ sorting, hash tables, binary search trees, graphs, minimum spanning tree, depth-first and breadth-first traversals, shortest-path algorithms, transitive closure (13 hours)

CC2001 modules

PF3 (5/14), PF4 (2/5) PL5 (3/3), AL1 (2/4), AL2(6/6), AL3 (7/12), DS5 (4/4)

Comments or Notes

- In addition to three weekly one-hour lectures, there is a structured laboratory as scheduled in the university timetable. **Attendance at these laboratory sessions is compulsory.**
- It is recommended that students complete Computer Science 2742 prior to registering for Computer Science 2711.

COMP 2742 (F, W)

Logic for Computer Science

Students Interested

This course is required for all computer science majors and minors.

Objectives of the Course

This course is an introduction to propositional and predicate logic with applications. The use of the system of boolean logic in reasoning and circuit design, as well as basic proof techniques and the resolution principle, for both propositional and predicate logic, will be covered. Concepts involving sets will be used to illustrate different types of proof techniques. The probable intractability of boolean logic and Goedel's incompleteness theorem will be presented.

Prerequisite(s)

COMP 1710

Successor(s)

COMP 3724*, COMP 3754*

Representative Workload

| | |
|--------------|-----|
| Assignments | 15% |
| Midterm Exam | 30% |
| Final Exam | 55% |

Representative Course Outline

- Logic as a reasoning system (2 hours)
- Propositional logic, connectives, truth tables, normal forms (3 hours)
- Validity, rules of inference, resolution (3 hours)
- Symbolic and truth table proofs, proof techniques (8 hours)
- Complexity of predicate logic proofs (3 hours)
- Predicate logic, predicates, existential and universal quantifiers (5 hours)
- The resolution principle for predicate logic (2 hours)
- Automated theorem proving (3 hours)
- Limitations of both predicate and propositional logic (2 hours)

CC2001 modules

DS2 (10/10), DS3 (12/12)

COMP 2760 (F)

Encountering the Computer: Society and the Individual

Students Interested

This course is required for all computer science majors and minors.

Objectives of the Course

This course examines social, ethical, legal and cultural issues surrounding the use of computers in modern society. These broader social issues are followed by an examination of the use of social and individual psychology in user interface design. Students will be expected to demonstrate an understanding of these issues both directly (through verbal and written discourse) and practically, as applied to the creation of actual software artifacts.

Prerequisite(s)

Two 1000-level English courses, or equivalent

Co-requisite(s)

COMP 2710

Successor(s)

COMP 4761*, COMP 4767*, COMP 4768*

Representative Workload

| | |
|------------------------------|-----|
| Case Studies and Assignments | 40% |
| Midterm Exam | 20% |
| Final Exam | 40% |

Representative Course Outline

- Module 1. Society and Computing (20 hours)
 - History of computing
 - Social context, social implications of computing
 - Analytical tools
 - Professional ethics
 - Risks and liabilities
 - Intellectual property
 - Privacy

- Module 2. Psychology and the Computer (4 hours)
 - Human models in computing
 - Social models

- Module 3. The Culture of Software Design (10 hours)
 - Review of design issues from Module 1
 - Graphical user interface tools
 - Cultural and social approaches to design
 - Design considerations

CC2001 modules

SP1 (1/1), SP2 (3/3), SP3 (2/2), SP4 (3/3), SP5 (2/2), SP6 (3/3), SP7 (2/2), SP8 (3/3),
HC1 (6/6), HC2 (2/2)

COMP 2801

Introductory Computing for Business

Students Interested

This course is intended primarily for students in the Faculty of Business Administration.

Objectives of the Course

Increasingly, computer users participate in the selection, development and use of computer applications. Decision makers at all levels need the knowledge and skills to effectively use, manage and interpret computer-based data and information. Therefore, the objectives of this course are to introduce students to important computing concepts in the classroom and to provide experience with selected application packages in hands-on laboratory exercises.

Prerequisite(s)

Level III Advanced Mathematics, Mathematics 1000 (which can be taken concurrently), or Mathematics 1090 (or equivalent)

Successor(s)

None

Representative Workload

| | |
|-------------------------|-----|
| Assignments (4-5) | 20% |
| Lab Quizzes / Labs (10) | 20% |
| Midterm Exam | 20% |
| Final Exam | 40% |

Representative Course Outline

- Introduction to information processing
 - Introduction to computer systems
 - a. Hardware
 - i. CPU
 - ii. Memory
 - iii. Secondary storage
 - iv. Input/Output devices
 - b. Software
 - i. System Software
 - ii. Application Software
 - System software and file management
 - Application software
 - a. Electronic spreadsheet

- b. Database management system
 - c. Presentation software
 - d. Integration
- Data representation
- Data Security
- Programming
 - Problem solving and algorithms
 - Basic programming concepts
 - Extending the functionality of MS-Office Software (Spreadsheets and Databases) using the basic programming concepts.
- Basic concepts of Web Page development
- Lab software to be used:
 - Windows Operating System
 - Web Browser
 - Web-based software
 - NotePad
 - MS Office
 - a. MS Excel
 - b. MS Access
 - c. MS PowerPoint
 - d. MS Word or any other word processor
 - e. Built-in programming language

Comments or Notes

- In addition to the weekly lecture component, there is a structured laboratory as scheduled in the university timetable. **Attendance at these laboratory sessions is compulsory.**
- Students can receive credit for only one of Computer Science 1600, Computer Science 2650, Computer Science 2801 or the former Business 2700.

3.3 3000-Level Courses

COMP 3550 - Introduction to Bioinformatics
COMP 3700 - Industrial Experience
COMP 3710 - Vocational Languages
COMP 3714 - Programming Languages and their Processors
COMP 3715 - Network Computing with WEB Applications
COMP 3716 - Software Methodology
COMP 3718 - Programming in the Small
COMP 3719 - Theory of Computation and Algorithms
COMP 3724 - Computer Organization
COMP 3725 - Computer Architecture and Operating Systems
COMP 3731 - Introduction to Scientific Computing
COMP 3753 - Computational Aspects of Linear Programming
COMP 3754 - Introduction to Information and Intelligent Systems

COMP 3550 (F)

Introduction to Bioinformatics

Students Interested

This course is designed as an interdisciplinary introductory course for both Computer Science and Biology students in bioinformatics, and as a bridge between both disciplines. The course is intended to be a course for a mixed audience of students with different backgrounds (e.g. computer science and biology). The course will focus on the fundamental concepts, ideas and related biological applications of existing bioinformatics tools. The purpose is to provide the students with hands-on experience on the major computational approaches applied to a wide variety of bioinformatics problems.

Biology students will appreciate the impact of these approaches for addressing biological questions and will gain insight on the limitations and strengths of these approaches. Computer Science students will appreciate the practical use of the concepts they have been taught in other courses, but most importantly, the challenges posed by biological questions, and the need for the robust algorithms that deal with the vary large, noisy datasets typically present in biology. Computer scientists and biologists will both recognize the large diversity of questions addressed by bioinformatics applications. Many industry and research jobs now require cross-disciplinary collaboration. With this course, students will start becoming aware of the interdisciplinary nature of bioinformatics and appreciate the contribution of people outside their field of study.

Objectives of the Course

Bioinformatics deals with the development and application of computational methods to address biological problems. The course will focus on the fundamental concepts, ideas and related biological applications of existing bioinformatics tools. This course will provide hands-on experience in applying bioinformatics software tools and online databases to analyze experimental biological data, and it will also introduce scripting language tools typically used to automate some biological data analysis tasks.

Prerequisite(s)

COMP 2500 and/or COMP 2710, and one Biology course at the 1000-level or above or permission of the course instructor

Successor(s)

None

Representative Workload

| | |
|--------------------------|-----|
| Assignments and Projects | 25% |
| Lab work and quizzes | 20% |
| Midterm Exam | 25% |
| Final Exam | 30% |

Representative Course Outline

- Introduction
 - What is Bioinformatics?
 - Why is Bioinformatics required?
 - Importance of interdisciplinary collaboration
- Sequences
 - Why compare sequences?
 - Sequence similarity
 - Where to look for information about a sequence?
 - Sequence alignment: Pairwise and multiple
- Genomics
 - How are genomes sequenced?
 - How are genomes annotated?
 - Genomic variation
 - Gene expression
 - How is gene expression measured?
 - Pre-processing the data: denoising and normalization
 - Differential analysis
 - Interpreting a list of genes
 - Gene functional annotation - Gene Ontology (GO)
 - Finding over-represented gene functions in gene lists
 - Other source of annotations
 - Gene function prediction
- Proteomics
 - Protein Interaction Networks
 - Protein Domains
 - How are proteins measured and identified?
- Transcriptomics
 - Motif finding
 - Determining binding preferences
 - Inferring regulatory networks
- Metabolomics
 - Detection and identification of metabolites
 - Human metabolome project

- Labs

Students will be expected to attend a weekly lab session, and to submit a lab report or to answer a lab quiz at the end of each lab.

- Script programming and using bioinformatics libraries (BioPerl)
- Sequences
 - Using BLAST, BLAT
 - Using alignment tools (e.g., ProbCons, M-Coffee)
- Working with sequenced genomes
 - Ensembl, BioMart, UCSC Genome Browser
 - Linking own data to a Genome browser
- Analysis of gene expression data using existing tools (e.g., Babelomics, GeneXPress, Gene Pattern)
- Annotating a list of genes with functional annotation
- Using over-representation or enrichment analysis tools (e.g., GSEA, DAVID, GenMAPP, GOMiner)
- Using gene function prediction systems (e.g., GeneMANIA, FuncBase, NBrowse, STRING, FunCoup)
- Using motif finding tools in a set of sequences (e.g. MEME, AlignACE)
- Using regulatory networks prediction systems (e.g. COALESCE, Allegro)

Comments or Notes

- Students can receive credit for only one of Computer Science 3550 or Biology 3951.
- In addition to the weekly lecture component, there is a structured laboratory as scheduled in the university timetable. **Attendance at these laboratory sessions is compulsory.**

COMP 3700 (F, W, S) Industrial Experience

Students Interested

Students who are admitted to the Computer Industry Internship Option (CIIO) are required to register for this non-credit course every semester during their internship.

Objectives of the Course

To provide an opportunity for qualified students to obtain rewarding placements that help them develop practical skills in a real work setting before graduation. The CIIO is available to computer science majors (B.Sc. and B.Sc. Honours only) who will typically apply between their third and fourth year of studies.

Prerequisite(s)

Admission to the Computer Industry Internship Option

Successor(s)

None

Representative Workload

- Within a month of starting the internship, students are required to submit a list of their work term objectives. They are also required to submit a **progress report** due the last day of classes of each semester in which they are working. The work term objective and progress reports are to be submitted to the (Co-op Education) Coordinator.
- At the end of the internship period, students are required to submit a **final report** which will include a description of their internship projects and activities as well as their original objectives and accomplishments. The final report is to be submitted to the Coordinator by the last day of classes of the semester in question. A completed Employer Evaluation form should be submitted to the Coordinator at the end of the internship period.

Representative Course Outline

Not applicable

Comments or Notes

- Students must register for the course Computer Science 3700 every semester during their internship. Computer Science 3700 is a non-credit course open only to students who have been accepted into the internship program.
- During the internship the employer and intern will complete student performance evaluations every four months and will submit them to the Coordinator. The final assessment of total

work performed is the responsibility of the Coordinator, and will be based upon both input from the employer and the intern's final report.

- At the end of the internship each intern will be assigned one of the following grades after the final assessment of their performance:
 - a) *Pass with Distinction (PWD)*: Indicates EXCELLENT PERFORMANCE in both the work report and work performance.
 - b) *Pass (PAS)*: Indicates that PERFORMANCE MEETS EXPECTATIONS in both the work report and the work performance.
 - c) *Fail (FAL)*: Indicates FAILING PERFORMANCE in the work report or the work performance.
- The following will be noted in the transcript of the intern:
 - Requirements for the Computer Industry Internship Option have been completed.
Internship Duration: - months.
 - A grade of NC (No Credit) for Computer Science 3700 will be awarded in all semesters of the internship option prior to the final semester.
- In case a student is enrolled in both the Honours program and the CIIO, the requirements of both must be met. Upon approval from the honours project supervisor, within the department, the employer and the head of the Department of Computer Science, an internship project may be submitted as a component of an honours project. These arrangements must be made within the first semester of the internship placement.
- For more information, see Section 2.5.4 of this handbook.

COMP 3710 (F) **Vocational Languages**

Students Interested

Students and data processing professionals, with a good knowledge in at least one modern high-level programming language, who intend to learn other languages currently in vogue.

Objectives of the Course

This course is intended to provide students with a working knowledge of a variety of high-level programming languages. When given a wide range of languages with which to solve a particular problem, the student will be able to choose the most appropriate language for implementing the solution. Emphasis will be on the general semantic characteristics and the underlying decisions implicit in the design and implementation of these languages.

Prerequisite(s)

COMP 2711

Successor(s)

None

Representative Workload

| | |
|----------------|-----|
| Assignments | 20% |
| In-class Exams | 30% |
| Final Exam | 50% |

Representative Course Outline

- A selection of languages from:
 - **C** - A modern “systems programming” language of the type which is largely superseding assembler languages
 - **Prolog** - A descriptive language used in artificial intelligence based on representing knowledge with facts and rules
 - **C++** - The most widely used object-oriented programming language that is compatible with C
 - **Perl** - A scripting language
 - **Python** - A scripting language
 - **Javascript** - A browser scripting language

COMP 3714 (W)

Programming Languages and their Processors

Students Interested

This course is of interest to students who want a study of the syntax and semantics of programming languages.

Objectives of the Course

Issues to be studied include virtual machines, translation, compiling, code generation and interpreters.

Prerequisite(s)

COMP 3719 and COMP 3724

Successor(s)

COMP 4717

Representative Workload

| | |
|--------------|-----|
| Assignments | 15% |
| Midterm Exam | 30% |
| Final Exam | 55% |

Representative Course Outline

- Review of typical elements of (imperative) programming languages
- Compilers and interpreters
- Specification of syntax and semantics of programming languages
- Expressions and assignments, side effects, control structures, data and procedural abstractions, parameter passing mechanisms, bindings, scopes, type systems
- Recursive-descent technique used for illustration of different aspects of syntax analysis, code generation and error recovery
- Language interpreters for low-level and high-level languages

COMP 3715 (F, W)

Network Computing with WEB Applications

Students Interested

This course is required for all computer science majors.

Objectives of the Course

To study how distributed applications (e.g., client/server Web applications) are constructed using the Internet. Topics covered include: the socket interface for network communication, client/server applications, browser scripting using Javascript, content generation for web applications (e.g., jsp, php), html/css documents, and the use of cryptography to handle security.

Prerequisite(s)

COMP 2711

Successor(s)

COMP 4759* , COMP 4768* , COMP 4770*

Representative Workload

| | |
|---------------------------|-----|
| Assignments | 10% |
| Project (Web application) | 25% |
| Midterm Exam | 15% |
| Final Exam | 50% |

Representative Course Outline

- Structure of Internet, Routing, DNS (1 hour)
- Layered Network Architecture (1/2 hour)
- TCP/IP (1/2 hour)
- Reliable communication (1 hour)
- Common protocols: FTP, HTTP, DHCP, etc. (1 hour)
- Socket Interface in C, Java for UDP and TCP (2 hours)
- Client/Server applications (3 hours)
- Threads and concurrency (3 hours)
- WEB content: HTML, CSS; Client-side scripting: Javascript (7 hours)
- Server content: XML, XSLT; Server-side web services, three tier architecture; One of: PHP, JSP, etc. (9 hours)
- Fundamentals of cryptography, Secret-key algorithms, Public-key algorithms, Protocols (SSL, PKI, PGP, CA, etc.), Digital signatures (3 hours)

CC2001 modules

NC1 (2/2), NC2 (7/7), NC3 (3/3), NC4 (3/3), NC5, NC7

COMP 3716 (F) **Software Methodology**

Students Interested

This course is required for all computer science majors.

Objectives of the Course

The objective of this course is to create a software system from requirements capture (through analysis, object-oriented design, implementation and deployment), perform requirements capture with use case analysis, create an object-oriented domain analysis model from the use cases, transform the domain model into software classes and apply design patterns in selecting and creating the software.

Prerequisite(s)

COMP 2711

Successor(s)

COMP 4718, COMP 4719*, COMP 4768*, COMP 4770*

Representative Workload

| | |
|-----------|-----|
| Project 1 | 15% |
| Project 2 | 15% |
| Project 3 | 20% |
| Midterm 1 | 25% |
| Midterm 2 | 25% |

Representative Course Outline

- Software development process definitions (2 hours)
- Use cases and UML use case notation (4 hours)
- Applying use case for requirements capture (7 hours)
- OO analysis and UML notation (3 hours)
- Applying OO analysis to use cases (6 hours)
- Modeling dynamic behavior with sequence and activity diagrams (3 hours)
- Applying dynamic modeling (2 hours)
- Design patterns (7 hours)
- Optional: Software tools, cvs, ant, junit (2 hours)

CC2001 modules

SE1 (8/8), SE2 (3/5), SE3 (3/3), SE4 (2/2), SE5 (4/4), SE6 (3/3), SE7 (3/3)

COMP 3718 (W)

Programming in the Small

Students Interested

This course is for students interested in the study of tools and techniques used in the construction of small software systems.

Objectives of the Course

The goal of this course is to demonstrate the tools and techniques used in the construction of small software systems. In software engineering, software development is characterized by programming in the large or programming in the small. Programming in the large deals with requirement analysis, system architecture design, module specification, module development, testing and maintenance - in other words, the entire software life cycle. Programming in the small deals with how a person creates the set of software components necessary to implement a module. Thus programming in the small is part of the larger software development process.

Prerequisite(s)

COMP 2711 and Mathematics 2320

Successor(s)

None

Representative Workload

| | |
|------------------|-----|
| Assignments | 20% |
| Software Project | 25% |
| In-class Exam | 20% |
| Final Exam | 35% |

Representative Course Outline

- Review/Introduction of the Java programming language
 - data and control structures
 - abstraction features and styles
 - pointers and memory management
 - common mistakes
 - classes

- An example of the complete system
 - the requirements
 - design decomposition techniques and strategies
 - the interface specification of submodules
 - the use of software libraries

- the actual source code
 - the supporting tool and files
 - testing
- Programming in the small
 - Parnas' information hiding
 - examples of decomposition according to Parnas
 - ADT and modules
 - testing and automatic testing
- Software development tools
 - source control systems: svn, git, hg, etc.
 - Integrated Development Environments: eclipse, netbeans, etc.
 - testing frameworks: junit
- The where, when and why of software libraries
- The design of utility software components
 - abstract data types
 - generic programming
 - libraries
- Strategies to the reading and modification of large programs
- Performance issues (time and memory)

COMP 3719 (F, W)

Theory of Computation and Algorithms

Students Interested

This course is required for all computer science majors.

Objectives of the Course

To study algorithm design relative to advanced data structures, to introduce non-standard (parallel/distributed) and abstract machine models and to introduce techniques for deriving intractability results (polynomial-time and general).

Prerequisite(s)

COMP 2711 and Mathematics 2320

Successor(s)

COMP 3714*, COMP 4711*, COMP 4712*, COMP 4719*, COMP 4740, COMP 4741, COMP 4742, COMP 4743, COMP 4748, COMP 4751*, COMP 4752*, COMP 4753*, COMP 4756, COMP 4761*, COMP 4762, COMP 4767

Representative Workload

| | |
|--------------|-----|
| Assignments | 25% |
| Midterm Exam | 40% |
| Final Exam | 35% |

Representative Course Outline

- Review of basic mathematical concepts and notation, alphabets, strings, languages, problems, asymptotic notation, complexity classes and their complements (3 hours)
- Efficient computations--the class P (12 hours)
 - the String Matching Problem; basic, finite state automata based and Knuth-Morris-Pratt algorithms
 - the String Recognition Problem; context--free languages and pushdown automata, dynamic programming, parsing
- The classes EXPTIME and NP (8 hours)
 - the Satisfiability Problem, deterministic and nondeterministic Turing machines
 - reducibility, NP-completeness and Cook's Theorem
 - the Tautology Problem, theorem proving, the class co-NP
- Beyond computability (10 hours)
 - the Church-Turing Thesis

- the Halting Problem, decidable and undecidable problems
- diagonalization proofs, undecidability of the Halting Problem
- reducibility, other undecidable problems

- Distributed algorithms, grid computing

(3 hours)

CC2001 modules

AL4 (3/3), AL5 (6/6)

Comments or Notes

- Credit cannot be obtained for both Computer Science 3719 and the former Computer Science 3711.
- Credit cannot be obtained for both Computer Science 3719 and the former Computer Science 3740.

COMP 3724 (F, W)

Computer Organization

Students Interested

This course is required for all computer science majors. Since this course addresses the how and why of computer organization, anyone who is interested in an indepth study of how computers are constructed should take this course.

Objectives of the Course

The objective of this course is to explore the how and why of computer organization. In order to accomplish this the course will examine the classical components of a computer system. These include: control, data path, memory, input and output. This course will also examine the common levels of abstraction used to reason about computer organization. These include: instruction set architectures, hardware components, register transfer level and logic design.

Prerequisite(s)

COMP 2711 and COMP 2742

Co-requisite(s)

Mathematics 2320

Successor(s)

COMP 3714*, COMP 3725, COMP 4711*, COMP 4712*, COMP 4723, COMP 4770*

Representative Workload

| | |
|----------------------|-----|
| Homework Assignments | 30% |
| In-class Exam | 20% |
| Final Exam | 50% |

Representative Course Outline

- Computer components: CPU, ALU, busses, memory, I/O devices (3 hours)
- Data representations: bits, integers, codes (3 hours)
- Integer arithmetic, bitwise operations (3 hours)
- Instruction Set Architecture: Addressing modes, Register sets, Instructions (6 hours)
- Machine-level representation of programs (6 hours)
- Boolean Algebra, Logic Design (3 hours)
- Busses (3 hours)
- Logical and register transfer level construction of CPUs and ALUs (3 hours)

CC2001 modules

AR1 (6/6), AR2 (3/3), AR3 (9/9), AR6 (7/7)

COMP 3725 (W)

Computer Architecture and Operating Systems

Students Interested

This course is required for all computer science majors.

Objectives of the Course

To introduce students to the low-level operations of computers. System design and the architectural implementations of these designs are explored. Memory management, operating systems, and I/O devices and their interactions are among the primary issues of this course.

Prerequisite(s)

COMP 3724

Successor(s)

COMP 4721, COMP 4754*, COMP 4759*, Special Topics in Computer Systems

Representative Workload

| | |
|-------------------------------|-----|
| Homework assignments | 20% |
| In-class Exams and/or Quizzes | 40% |
| Final Exam | 40% |

Representative Course Outline

- Review of computer components: CPU, ALU, Busses, Memory, I/O Devices (1 hour)
- Introduction to building upon simple circuits (1 hour)
- Multiprocessing and alternative architectures (3 hours)
 - single-cycle vs. multi-cycle circuits
 - pipelining
 - efficient resource allocation algorithms
- Memory management
 - organization and the memory hierarchy (5 hours)
 - shared-memory management (2 hours)
 - cache memory (1 hour)
 - virtual memory (2 hours)
 - memory management for multiprocessor systems (3 hours)
- Operating systems
 - operating system principles (2 hours)

- system processes and scheduling algorithms (3 hours)
- concurrency (resource management - linked to memory management) (3 hours)
- synchronization, interfacing and communication (3 hours)
- semaphores and monitors (2 hours)
- inter-process communication (1 hour)

CC2001 modules

AR4 (5/5), AR5 (3/3), AR7 (3/3), OS1 (2/2), OS2 (2/2), OS3 (6/6), OS4 (3/3), OS5 (5/5)

COMP 3731 (F)

Introduction to Scientific Computing

Students Interested

This course is of interest to students who wish to perform numerical computations in such areas as engineering, physics, statistics, modelling, graphics or operations research, and also to those students who wish to study the effectiveness of these solutions in a given computing environment.

Objectives of the Course

The development of algorithms for the numerical solution of mathematical problems, and the study of the numerical stability of these algorithms are the main objectives of this course. The efficiency of these algorithms with respect to speed and storage requirements is considered as well. Emphasis is also placed on the study of the sensitivity of selected problems to perturbations in the data. There is also a brief introduction to the development of numerical algorithms that take advantage of advanced computer architectures, such as pipeline processors, array processors and parallel processors.

Prerequisite(s)

Mathematics 2000 and Mathematics 2050, and one of COMP 2602 or COMP 2710

Successor(s)

COMP 4734, Special Topics in Numerical Computations

Representative Workload

| | |
|----------------|-----|
| Assignments | 50% |
| In-class Exams | 20% |
| Final Exam | 30% |

There will be about nine assignments given throughout the semester. Programming assignments are chosen to illustrate topics discussed in the lecture material. They emphasize the numerical dangers that may appear because of the finite precision of computers. Assignments can be written in any programming language unless specified otherwise. Nonprogramming problems are also assigned.

Representative Course Outline

- Errors in computations, computer arithmetic, stability of algorithms and conditioning of problems
- Principles of pipelining and parallel processing
- Evaluation of functions, computation of infinite alternating series
- Nonlinear equations
- Methods for systems of linear equations

- Norms, condition numbers
- Polynomial interpolation, curve fitting, least squares problem, cubic spline
- Numerical integration

Comments or Notes

- Credit cannot be obtained for both COMP 3731 and Mathematics 3132.

COMP 3753

Computational Aspects of Linear Programming

Students Interested

Those who wish to have an introduction to linear optimization problems that arise in many areas such as operations research.

Objectives of the Course

To analyze the Linear Programming (linear optimization) Problem, to investigate the recent developments in the theory necessary to solve this problem, to design efficient algorithms for its solution and to analyze the complexity of these algorithms and their numerical efficiency.

Prerequisite(s)

Mathematics 2050, and one of COMP 2602 or COMP 2710

Successor(s)

Special Topics in Numerical Computations

Representative Workload

| | |
|----------------|-----|
| Assignments | 50% |
| In-class Exams | 20% |
| Final Exam | 30% |

Programming assignments emphasize the numerical dangers that may appear because of the finite precision of computers. Some of the programming will use MATLAB.

Representative Course Outline

- Brief review of necessary linear algebra
- Introduction to the Linear Programming Problem (LPP)
- The simplex algorithm
- Sparse matrix techniques for the LPP problem
- Duality and postoptimality analysis
- Extensions to the simplex algorithm
- A brief introduction to interior algorithms for the LPP

COMP 3754 (F, W)

Introduction to Information and Intelligent Systems

Students Interested

This course is required for all computer science majors. The course is also of interest to other students who want exposure to applications such as Intelligent Systems and Information Management.

Objectives of the Course

To introduce students to application areas that are away from usual number-based and text-based processing. Students will learn the basic concepts and become aware of the historical developments and social and ethical issues related to application areas such as Intelligent Systems and Information Management. This exposure will help students to become knowledgeable about managing large volumes of data and dealing with problems that are well defined but whose algorithmic solutions are not feasible, or problems that are fuzzily defined.

Prerequisite(s)

COMP 2711 and COMP 2742

Successor(s)

COMP 4752*, COMP 4753*, COMP 4754*, COMP 4770*

Representative Workload

| | | |
|-----------------|-----|---|
| Projects | 30% | (2 Projects – 1 in Intelligent Systems and 1 in Information Management) |
| Midterm Exam(s) | 20% | |
| Final Exam | 50% | |

Representative Course Outline

- Introduction to computers as Universal Machines (1/2 hour)
- Introduction to important application areas (1/2 hour)
 - Intelligent Systems
 - Information Management
- Information Management (15 hours)
 - Definition of Information Management systems
 - History and motivation for information systems
 - Types of Information Management systems
 - File Management systems
 - Databases and Database Management systems
 - Comparison of File Management systems and Database Management systems

- Databases and Database Management systems
 - Components of database systems
 - DBMS functions
 - Database architecture and data independence
- Models of Database Management systems
 - Relational
 - Hierarchical
 - Network
 - Object-Oriented
- Data modeling
 - Data models based on the types of concepts that they provide to describe the database structure – i.e., conceptual data model, physical data model, and representational data model
 - Entity relationship model and UML, including their use in data modeling
 - Concepts of the OO model such as object identity, type
- Relational databases
 - Fundamental terminology used in the relational data model
 - Design of a simple system
 - Implementation of the designed database
 - Querying the database
- Social and ethical issues
 - Privacy
 - Integrity
 - Security
 - Preservation
 - Scalability
 - Efficiency
 - Effectiveness
- Intelligent Systems (15 hours)
 - Definition of Intelligent Systems
 - History of Artificial Intelligence
 - Definition of Intelligence
 - Philosophical Questions
 - * Turing Test
 - * Searle’s “Chinese Room” thought experiment
 - Fundamental definitions
 - * Optimal versus human-like reasoning

- * Optimal versus human-like behaviour
- Types of problems studied in the area of AI
 - Pattern matching
 - Game playing
 - Natural language processing
 - Expert Systems
- Ethical issues in AI
- Search and constraint satisfaction
 - Problem solving
 - Problem space
 - Techniques to solve the problems
 - * Basic Search techniques
 - * Heuristic Search
- Knowledge representation and reasoning
 - Introduction and overview of Propositional and Predicate Logic
 - Theorem Proving
- AI programming language: Basic Prolog programming

CC2001 modules

IS1 (1/1), IS2 (5/5), IS3 (4/4), IM1 (3/3), IM2 (3/3), IM3 (4/4), IM4, IM5

3.4 4000-Level Courses

COMP 4711 - Structure of Programming Languages
COMP 4712 - Compiler Construction
COMP 4715 - Special Topics in Programming Languages
COMP 4718 - Survey of Software Engineering
COMP 4721 - Operating Systems
COMP 4723 - Introduction to Microprocessors
COMP 4726 to 4729 - Special Topics in Computer Systems
COMP 4734 - Matrix Computations and Applications
COMP 4736 to 4739 - Special Topics in Numerical Computations
COMP 4740 - Design and Analysis of Algorithms
COMP 4741 - Formal Languages and Computability
COMP 4742 - Computational Complexity
COMP 4743 - Graph Algorithms and Combinatorial Optimization
COMP 4745 to 4749 (excluding 4748) - Special Topics in Theoretical Aspects
COMP 4751 - Computer Graphics
COMP 4752 - Introduction to Computational Intelligence
COMP 4753 - Artificial Intelligence
COMP 4754 - Database Systems
COMP 4756 - Image Processing
COMP 4759 - Computer Networks
COMP 4762 - Introduction to Computational Molecular Biology
COMP 4766 - Introduction to Autonomous Robotics
COMP 4767 - Information Visualization and Applications
COMP 4768 - Software Development for Mobile Devices
COMP 4770 - Team Project
COMP 4780 - Honours Project
COMP 4800 to 4825 - Special Topics

COMP 4711

Structure of Programming Languages

Students Interested

Those who wish to pursue the study of design aspects of programming languages.

Objectives of the Course

To develop an understanding of the organization of programming languages and the runtime behavior of programs and to introduce basic concepts of formal description of programming languages.

Prerequisite(s)

COMP 3719 and COMP 3724

Successor(s)

Special Topics in Programming Languages

Representative Workload

| | |
|----------------|-----|
| Assignments | 20% |
| In-class Exams | 40% |
| Final Exam | 40% |

Representative Course Outline

- Formal specification of syntax
- Data, domains and domain constructors
- Storage and assignments
- Control structures, selection and iteration
- Binding, environment and scope
- Procedures and parameters
- Sequencers and continuations
- Concurrency
- Types and type checking, type constructors
- Introduction to denotational semantics

COMP 4712 (F)

Compiler Construction

Students Interested

Those who are interested in theoretical foundations of compiler construction.

Objectives of the Course

To provide a theoretical basis for practical approaches to compiler construction. Primary emphasis is upon popular methods of syntax analysis and code generation. Some background in formal languages and automata theory is required.

Prerequisite(s)

COMP 3719 and COMP 3724

Successor(s)

Special Topics in Programming Languages

Representative Workload

| | |
|--------------------------|-----|
| Assignments and Projects | 20% |
| In-class Exams | 40% |
| Final Exam | 40% |

Representative Course Outline

- Formal languages and their grammars, Chomsky hierarchy, context-free languages, derivation trees, leftmost and rightmost derivations, ambiguity, grammar transformations, normal forms
- Definition of programming languages, BNF and EBNF, syntax and semantics, syntax-driven semantics, lexical, syntax and semantics analysis
- Lexical analysis, finite automata and regular expressions, construction of scanners, lex
- Bottom-up parsing, general and deterministic, LR(k) grammars and parsers, LR, SLR, LALR parsers, code generation for bottom-up parsers, yacc
- Top-down parsing, general and deterministic, LL(k) grammars and parsers, action symbols and code generation for top-down parsing, recursive descent parsing
- Attribute grammars, inherited and synthesized attributes, attribute evaluation schemes

COMP 4715

Special Topics in Programming Languages

Students Interested

Those who want a more in-depth study in the area of programming languages.

Objectives of the Course

To give students exposure to current and topical information in the area of programming languages.

Prerequisite(s)

Special topics courses are not offered on a regular basis, but whenever departmental resources permit. For this reason the prerequisites can vary each time they are offered.

Successor(s)

None

Representative Workload

Variable

Representative Course Outline

Variable

Comments or Notes

- All relevant information on a special topics course is posted on the Computer Science website and the bulletin board outside the Computer Science General Office several weeks before the beginning of the semester in which the course is being offered.

COMP 4718

Survey of Software Engineering

Students Interested

This course is of interest to those students wanting exposure to the design and development of a large-scale software system.

Objectives of the Course

This course introduces methods and tools for developing, managing and maintaining large-scale software systems and provides students with practical experience in the development of a software prototype that goes through the life cycle of large-scale software development.

Prerequisite(s)

COMP 3716

Successor(s)

None

Representative Workload

| | |
|----------------|-----|
| Assignments | 35% |
| Project Report | 15% |
| Tests | 20% |
| Final Exam | 30% |

Representative Course Outline

- Software products and development models
- Project management and cost estimation
- Software requirement engineering (system requirements, software requirements, definition, specification, validation)
- Software prototyping
- Formal specification
- Software design methodologies (object-oriented vs. function-oriented design, user interface design, real-time systems)
- Software validation (program verification and validation, testing, techniques)
- Programming techniques and environments (reliability and reuse CASE)
- Software maintenance, configuration management, software re-engineering

Project Outline

- Use the concepts and techniques of software engineering to develop a software prototype of your own interests (recommended: a potential commercial system).
- Hand in a report of the project. In the report, you need to use your software prototype to show your understanding and practice of software development. Your report must cover the specified stage(s) of the software development life-cycle. As a guideline for you to schedule the project, the report will be broken down into four stages that will be specified respectively in assignments. The report will be marked according to its readability and your discussions on:
 - What should be done?
 - What methods or techniques are available?
 - What did you do and how did you do it?
 - What you did not do and why?
 - (optional) What feedback did you get from the succeeding steps and what modifications did you make?

COMP 4721 (F)

Operating Systems

Students Interested

This course is of interest to those who want to learn about the informal structures of operating systems in batch processing, multi-programming, multi-processing and time-sharing environments.

Objectives of the Course

The main objectives of the course are to learn how an operating system interfaces the hardware and software resources with the user's environment, to learn the compositions and connections of multilevel operating systems and to design substantial parts of an operating system.

Prerequisite(s)

COMP 3725

Successor(s)

Special Topics in Computer Systems, recommended prerequisite for COMP 4726

Representative Workload

| | |
|-----------------------|-----|
| Assignments (up to 6) | 25% |
| In-class Exams | 35% |
| Final Exam | 40% |

Representative Course Outline

- Introduction and overview
- History of operating systems
- Operating system structures
- Process management: process concept, concurrent processes, CPU scheduling, scheduling algorithms
- Process co-ordination: critical section, process synchronization, semaphores, monitors, critical regions, process communication
- Deadlocks: deadlock prevention, avoidance, detection and recovery
- Memory hierarchy: cache memory, associative memory
- Memory management: swapping, fixed and variable partitions, relocation, paging and segmentation (external and internal fragmentation)
- Virtual memory: page replacement algorithms, thrashing
- Secondary storage management
- File management: file-system organization, file operations, access methods, directory-structure organization
- Protection: access matrix, security, encryption
- Elements of distributed operating systems

- Selected case studies, e.g. UNIX

Comments or Notes

- Students can receive credit for only one of Computer Science 4721 or Engineering 8894.

COMP 4723

Introduction to Microprocessors

Students Interested

This course will be of interest to students who want to gain knowledge of embedded systems with microcontrollers.

Objectives of the Course

This course aims to give students an introduction to the architecture of microcontrollers. Students design and implement simple interface and control functions using C and assembly language. Basic electric circuit theory and electronics for the interfacing of microcontrollers with sensors and actuators will be covered.

Prerequisite(s)

COMP 3724

Successor(s)

Special Topics in Computer Systems

Representative Workload

| | |
|------------|--|
| Labs | 15% |
| Midterm | 15% |
| Project | 40% (including proposal, presentation and documentation) |
| Final Exam | 30% |

The project includes an initial presentation of the project proposal, with a brief write-up, a final project presentation and a project write-up.

Representative Course Outline

- Introduction to C
- Basic circuits
- Basic architecture of the ATMEL processors
- Assembly language programming
- On-chip components
- Serial peripherals
- Analog/digital control
- Electronics
- Applications

Comments or Notes

- Enrollment is limited depending on laboratory resources.
- Laboratory: Three hours per week for the first six or seven weeks, after which the laboratory time will be devoted to project development. **Attendance at laboratory sessions is compulsory.**

COMP 4726-4729
Special Topics in Computer Systems

Students Interested

Those who want a more in-depth study of the area of computer systems.

Objectives of the Course

To give students exposure to current and topical information in the area of computer systems.

Prerequisite(s)

Special topics courses are not offered on a regular basis, but whenever departmental resources permit. For this reason the prerequisites can vary each time they are offered.

Successor(s)

None

Representative Workload

Variable

Representative Course Outline

Variable

Comments or Notes

- All relevant information on a special topics course is posted on the Computer Science website and the bulletin board outside the Computer Science General Office several weeks before the beginning of the semester in which the course is being offered.

COMP 4734 (W)

Matrix Computations and Applications

Students Interested

Students in any discipline who are dealing with computer solutions of numerical linear algebra problems. Matrix computations may be found useful in disciplines such as engineering, statistics, physics, optimization, operations research, computational chemistry and signal processing.

Objectives of the Course

An introduction to the techniques of numerical linear algebra. Emphasis is placed upon developing the most recent and reliable algorithms. The stability of these algorithms as well as the sensitivity of the problems they solve will also be studied.

Prerequisite(s)

COMP 3731

Successor(s)

COMP 4735, Special Topics in Numerical Computations

Representative Workload

| | |
|--------------|-----|
| Assignments | 50% |
| Midterm Exam | 20% |
| Final Exam | 30% |

There will be about nine assignments given throughout the semester. Programming assignments are chosen to illustrate topics discussed in the lecture material and can be written in any programming language unless otherwise specified. Nonprogramming problems are also assigned.

Representative Course Outline

- An introduction to necessary topics of linear algebra
- Systems of linear equations; scaling; iterative refinement; estimating the condition number of a matrix
- Introduction to pipelining and parallel matrix computations
- The linear least squares problem
- The symmetric and unsymmetric eigenproblems
- The singular value problem of a matrix

COMP 4736-4739
Special Topics in Numerical Computations

Students Interested

Those who want a more in-depth study in the area of numerical computations.

Objectives of the Course

To give students exposure to current and topical information in the area of numerical computations.

Prerequisite(s)

Special topics courses are not offered on a regular basis, but whenever departmental resources permit. For this reason the prerequisites can vary each time they are offered.

Successor(s)

None

Representative Workload

Variable

Representative Course Outline

Variable

Comments or Notes

- All relevant information on a special topics course is posted on the Computer Science website and the bulletin board outside the Computer Science General Office several weeks before the beginning of the semester in which the course is being offered.

COMP 4740 (F)

Design and Analysis of Algorithms

Students Interested

This course is of interest to students wishing to formalize algorithmic problem solving techniques.

Objectives of the Course

To give students an overview of techniques for the design of efficient optimal-solution and heuristic algorithmic solutions.

Prerequisite(s)

COMP 3719

Successor(s)

None

Representative Workload

| | |
|-----------------|-----|
| Assignments (5) | 40% |
| Midterm Exam | 25% |
| Final Exam | 35% |

Representative Course Outline

- Optimal-Solution Algorithm Design Techniques - Solution Tree Search (Backtracking/Branch and Bound), Divide and Conquer, Dynamic Programming, Greedy Algorithms
- Heuristic Algorithm Design Techniques - Solution Spaces and Landscapes, Local Search, Simulated Annealing, Genetic Algorithms
- Advanced Data Structures - Set Maintenance and Search (B-Trees, Red-Black Trees, Binomial Heaps), String Processing (Keyword Trees, Suffix Trees), Disjoint Sets, Augmenting Data Structures

COMP 4741

Formal Languages and Computability

Students Interested

This course is of interest to those students seeking a deeper understanding of classical formal language theory and computability.

Objectives of the Course

This course is an in-depth discussion of classical models of computation, their computational power and their use in the classification of problems into classes. In addition, the correspondence between the models of computation and the different types of grammars is established.

Prerequisite(s)

COMP 3719

Successor(s)

Special Topics in Theoretical Aspects

Representative Workload

| | |
|-----------------|-----|
| Assignments (5) | 40% |
| Midterm Exam | 25% |
| Final Exam | 35% |

Representative Course Outline

- Review of mathematical preliminaries: sets, binary relations, equivalence relations, partial orders, functions, finite and infinite sets, countable and non-countable sets, alphabet strings, string operations, languages, operations on languages
- Finite state automata, minimization, nondeterminism, closure properties of regular languages, regular expressions, pumping lemma
- Pushdown automata, context-free languages and grammars, equivalence, ambiguity, pumping lemma, parsing
- Turing machines, nondeterminism, multiple tapes, recursive and recursively enumerable languages
- The Chomsky hierarchy
- Decidability of problems concerning regular languages, context free languages and general languages
- Undecidability of the Halting Problem
- Reducibility and its application in proving undecidable and decidable languages
- The Post correspondence problem
- Oracle reductions and the arithmetic hierarchy
- The Recursion Theorem and its applications

Comments or Notes

- Students can receive credit for only one of Computer Science 4741 or the former Computer Science 3740.

COMP 4742

Computational Complexity

Students Interested

This course is of interest to students wishing to deepen their understanding of the nature of problem complexity.

Objectives of the Course

This course is an in-depth look at the theory of algorithms and algorithm complexity from a structural point of view. The emphasis will be placed on complexity classes containing problems of practical relevance such as P, NP and the parallel class of problems NC.

Prerequisite(s)

COMP 3719

Successor(s)

None

Representative Workload

| | |
|---------------|-----|
| Assignments | 40% |
| Inclass Exams | 20% |
| Final Exam | 40% |

Representative Course Outline

- Review of the basic formal models of computation and complexity
 - random access machines
 - Turing machines
 - oracle machines
 - alternating Turing machines
 - combinational circuits model
 - uniform and nonuniform complexity measures
 - resource bounded computations
- Complexity classes
 - resource bounded reducibility (Turing-Cook, polynomial time, logarithmic space)
 - the classes NP, P, NC, PSPACE, LOGSPACE and their complements
 - problems complete and hard for a complexity class
 - relationships between complexity classes
- The polynomial time hierarchy

- Randomized computations
 - randomized algorithms
 - randomized complexity classes
 - randomized sources

COMP 4743

Graph Algorithms and Combinatorial Optimization

Students Interested

This course is of interest to students wanting to deepen their understanding of graph and network optimization problems.

Objectives of the Course

To give students efficient algorithms for solving some graph and network optimization problems and to provide them with techniques to show the apparent intractability of others.

Prerequisite(s)

COMP 3719

Successor(s)

None

Representative Workload

| | |
|-----------------|-----|
| Assignments (5) | 40% |
| Midterm Exam | 25% |
| Final Exam | 35% |

Representative Course Outline

- Graph theory fundamentals
- Algorithms on graphs: graph connectivity and traversals, matching, shortest path, isomorphism, testing membership on families of graphs: bipartite, planar, of bounded tree-width
- Some NP-complete and hard problems on graphs, colourability, independent sets, vertex cover, clique
- Approximation algorithms for some graph theoretic problems
- Resource scheduling problems
- Greedy algorithms and scheduling problems, dynamic programming algorithm for scheduling of weighted intervals
- The maximum flow problem and the Ford-Fulkerson algorithm, maximum flow and minimum cuts, the preflow-push maximum-flow algorithm
- Applications of network flow

COMP 4745-4749 (excluding COMP 4748)
Special Topics in Theoretical Aspects

Students Interested

Those who want a more in-depth study in the area of theoretical computing.

Objectives of the Course

To give students exposure to current and topical information in the area of theoretical computing.

Prerequisite(s)

Special topics courses are not offered on a regular basis, but whenever departmental resources permit. For these reasons the prerequisites can vary each time the courses are offered.

Successor(s)

None

Representative Workload

Variable

Representative Course Outline

Variable

Comments or Notes

- All relevant information on a special topics course is posted on the Computer Science web-site and the bulletin board outside the Computer Science General Office several weeks before the beginning of the semester in which the course is being offered.

COMP 4751 (W)

Computer Graphics

Students Interested

Those interested in graphics in general, user interface tools and techniques, and state-of-the-art concepts in computer graphics.

Objectives of the Course

To introduce the students to the state-of-the-art concepts and trends in computer graphics including graphics standards. Furthermore, the underlying algorithms, as well as the basic techniques to develop them, will be presented.

Prerequisite(s)

COMP 3719 and Mathematics 2050

Successor(s)

COMP 4757

Representative Workload

| | |
|--------------|-----|
| Assignments | 40% |
| Midterm Exam | 20% |
| Final Exam | 40% |

Representative Course Outline

- Graphics systems: hardware and software
- Two-dimensional computer graphics
 - 2D primitives and rendering algorithms, primitive attributes
 - 2D transformations
- Graphics user interface: devices and dialogue design
- Three-dimensional computer graphics
 - 3D representations: curves, surfaces and solid objects
 - 3D transformations, projections and 3D viewing
 - Visualization: surface visibility, electronic colour, illumination and shading

Comments or Notes

- In addition to the three hours of classroom teaching, there will be a minimum three-hour laboratory each week to be scheduled by the department.

COMP 4752 (W)

Introduction to Computational Intelligence

Students Interested

This course will be of interest to students wishing to study computer implementations of adaptive mechanisms that facilitate intelligent behavior in complex and changing environments.

Objectives of the Course

To introduce students to the fundamentals of computational intelligence; in particular, the four major computational intelligence methods: artificial neural networks, evolutionary computation, swarm intelligence and fuzzy systems. The integration of these techniques for problem solving will be discussed.

Prerequisite(s)

COMP 3719 and COMP 3754

Successor(s)

Special Topics in Computational Intelligence

Representative Workload

| | |
|----------------------|-----|
| Assignments | 20% |
| Midterm Exam | 20% |
| Final Project Report | 30% |
| Final Exam | 30% |

Representative Course Outline

- Introduction to computational intelligence, including the background and history of evolutionary computation, neural networks, and fuzzy logic, and how they form the basis for the unified field of computational intelligence.
- Review of evolutionary computation theory and concepts: genetic algorithms, evolutionary programming, evolution strategies, genetic programming.
- Review of swarm intelligence, including particle swarm optimization and ant colony optimization.
- Review of basic neural network theory and concepts: supervised and unsupervised paradigms, network architectures.
- Back-propagation implementation, including learning algorithm, network architecture, and data preparation.
- Review of fuzzy systems theory and concepts: membership functions, fuzzy sets, fuzzy logic.
- Evolutionary design of artificial neural networks.
- Evolutionary design of fuzzy systems.

- Neuro-fuzzy systems.
- Fuzzy evolutionary algorithms.

COMP 4753

Artificial Intelligence

Students Interested

Those interested in exploring the idea that computers can be programmed to display “human-like” intelligence in situations other than scientific computing or data processing tasks.

Objectives of the Course

To introduce students to methods used in artificial intelligence programs to make the program behave intelligently, in particular heuristic programming versus algorithmic programming.

Prerequisite(s)

COMP 3719 and COMP 3754

Successor(s)

Special Topics in Artificial Intelligence

Representative Workload

| | |
|--------------|-----|
| Assignments | 20% |
| Midterm Exam | 20% |
| Project | 20% |
| Final Exam | 40% |

Representative Course Outline

- Introduction to AI, overview of AI application areas
- Propositional calculus
- Predicate calculus, inference rules, unification
- Search and problem solving: structures and strategies for state space search, control and implementation of state space search, heuristic search
- AI languages and Prolog: requirements for AI languages, introduction to Prolog
- Rule-based expert systems
- Vision: image processing, scene analysis
- Knowledge representation: network representations, structured representations, type hierarchies, inheritance
- Advanced representation in Prolog
- The General Problem Solver
- Natural language understanding: syntax analysis, context-free grammars, Augmented Transition Network parser, natural language applications
- Machine learning

COMP 4754

Database Systems

Students Interested

This course will be of interest to those who wish to have an introduction to the problems involved in dealing with large amounts of data.

Objectives of the Course

To introduce students to database processing, database management systems and database design considerations. Additional topics covered include the theory and methodologies essential for the relational database design, implementation, manipulation, optimization and management.

Prerequisite(s)

COMP 3725 and COMP 3754

Successor(s)

Special Topics in Database Systems

Representative Workload

| | |
|-------------------------|-----|
| Assignments and Project | 40% |
| Midterm Exam | 20% |
| Final Exam | 40% |

Representative Course Outline

- Introduction to databases and database management systems
- Entity relationship model
- Database design
- Requirement analysis
- Conceptual modelling
- Implementation design
- Physical design
- Modelling of data for various database models
- Dependencies, normal forms and other database design considerations
- Some sample query languages and query processing
- Query optimization
- Concurrency
- Recovery
- Security and integrity
- Distributed databases

COMP 4756 (W) **Image Processing**

Students Interested

This course will be of interest to those students who wish to learn techniques of specifying, designing and implementing digital image processing software systems.

Objectives of the Course

To provide the key analytical and algorithmic tools and concepts of digital image processing systems and to apply these tools and concepts to examples chosen from a wide variety of application areas.

Prerequisite(s)

COMP 3719

Successor(s)

Special Topics in Applications (Image Processing)

Representative Workload

| | |
|-----------------------------|-----|
| Assignments | 30% |
| Term Project (Presentation) | 20% |
| In-class Exam | 20% |
| Final Exam | 30% |

Representative Course Outline

- Image perception
- Image transformation
- Image enhancement
- Image filtering and restoration
- Image analysis
- Image data compression

Comments or Notes

- In addition to three weekly lectures, there will be a minimum three-hour laboratory per week to be scheduled by the department.

COMP 4759

Computer Networks

Students Interested

It is becoming more difficult to avoid hearing about the Internet and its promise to hook-up the world. The purpose of this course is to show how the Internet really works. It is concerned with exactly how one computer successfully communicates with a “net” of other computers.

Objectives of the Course

The main objective of this course is to study the protocols, the design of protocols, and the implementations of these protocols used to communicate between computers. Several standard protocols will be examined.

Prerequisite(s)

COMP 3715 and COMP 3725

Successor(s)

None

Representative Workload

| | |
|------------|-----|
| Projects | 40% |
| Tests | 20% |
| Final Exam | 40% |

Representative Course Outline

- Basic terms of communication networks
- The Berkeley Sockets
- The data line layer
- The physical layer
- Error/correcting and detecting codes
- Local area networks
- The network layer
- Transport, session and presentation
- Encryption
- Applications

COMP 4762

Introduction to Computational Molecular Biology

Students Interested

This course is of interest to those wishing to have an introduction to computational biology.

Objectives of the Course

In recent years, a growing number of genome projects worldwide has been producing ever-increasing amounts of data on the genetic makeups of a variety of organisms, from bacteria to plants to human beings. It is widely recognized that extracting useful information from this data will involve computation on a massive scale. This course will give an overview of computational problems and algorithms for these problems associated with a variety of analyses of biological molecular data. As such, this course will focus on the computational complexity (both time and space) of these problems and algorithms; it is not intended as a course on how to use existing computational biology software packages, and no prior knowledge of biology is required.

Prerequisite(s)

COMP 3719

Successor(s)

None

Representative Workload

| | |
|-----------------|-----|
| Assignments (4) | 30% |
| Term Project | 25% |
| Midterm Exam | 15% |
| Final Exam | 30% |

Representative Course Outline

- Introduction
- Pattern matching, pattern detection, and alignment
- Inferring evolutionary trees
- Sequence folding
- Meta-sequence analyses

COMP 4766

Introduction to Autonomous Robotics

Students Interested

Autonomous robotics is a very active research area in computer science. It addresses fundamental questions of how an agent can move, navigate, and carry out high-level tasks in unknown and noisy environments. This course will introduce students to algorithms and technologies that have emerged from research in autonomous robotics. These concepts are critical in any study of robotics systems. Further, they have wider applicability in various industries where properties of the real world must be sensed, modelled, and action upon.

Objectives of the Course

This course will introduce students to the fundamental constraints, technologies, and algorithms of autonomous robotics. The focus will be on computational aspects of autonomous wheeled mobile robots. The most important themes will be mobility, perception, and navigation. Assignments will require the implementation of controllers for robots using the Webots and the LEGO Mindstorms NXT robot kits.

Prerequisite(s)

COMP 2711, Mathematics 2000, Mathematics 2050, and Statistics 1510 or 2510

Successor(s)

None

Representative Workload

| | |
|-----------------|-----|
| Assignments (5) | 40% |
| Midterm Exam | 20% |
| Final Exam | 40% |

Representative Course Outline

- Introduction - Major paradigms in robotics
- Mobility - Methods of locomotion; kinematics; simple control systems
- Perception - Sensor technologies; stereo vision; modelling uncertainty of sensors and positional information
- Localization and Navigation - Environmental representation; Kalman and particle filtering; simultaneous localization and mapping (SLAM)
- Motion Planning - 2-D path planning; obstacle avoidance

COMP 4767 (F)

Information Visualization and Applications

Students Interested

Significant amounts of information are generated within many domains, including physical science, social science, and business. Understanding underlying patterns within such information is often difficult when only considering the raw data. Visualization takes advantage of human visual processing capabilities to enhance the ability of people to understand the underlying features of the information. This course will focus on the specific challenges with visualizing abstract information that does not have a direct relationship with the physical world. In particular, information visualization deals with the difficulties associated with representing such abstract information in a manner that supports interaction, exploration, and understanding. The broad application of information visualization to many diverse fields will make this course attractive to many students.

Objectives of the Course

The purpose of this course is to introduce students to the fundamental theories of human perception and information visualization. A specific focus will be placed on the design and implementation of applications that produce interactive visual representations of abstract information. Topics will include the human element of visualization, the mapping of data types to visual representations, encoding relationships present within the information, and supporting interaction and exploration within the visual representations. Packages such as the prefuse information visualization toolkit will be used to support the application development requirements of the class.

Prerequisite(s)

COMP 2760 and COMP 3719

Successor(s)

None

Representative Workload

| | |
|-----------------|-----|
| Assignments (4) | 40% |
| Midterm Exam | 20% |
| Final Exam | 40% |

Representative Course Outline

- Visual perception and the human element in information visualization
- Norman's stages of action framework
- Gestalt principles
- Information data types

- Visual encodings of abstract information
- Visual encodings of relationships within information
- Interacting with information visualization applications
- Understanding user goals and tasks
- Introduction to evaluation methods for information visualization

COMP 4768

Software Development for Mobile Devices

Students Interested

This course is of interest to students who wish to develop software in a networked mobile environment.

Objectives of the Course

The purpose of this course is to introduce students to the unique challenges of developing software in a networked mobile environment. Four fundamental aspects will be addressed in this course: software engineering for a mobile environment, network computing in a mobile environment, graphics programming for mobile devices, and human-computer interaction for mobile devices. In addition to these general topics, there will be components of the course that are specific to the iPhone/iPod Touch software development kit. These include the required programming language (Objective C), and the supported methods for networking, graphics display, and interaction.

Prerequisite(s)

COMP 2760, COMP 3715 and COMP 3716

Successor(s)

None

Representative Workload

| | |
|-------------------------|-----|
| Assignments (4) | 20% |
| Milestone-based Project | 40% |
| Presentation/Demo | 5% |
| Final Exam | 35% |

There will be one assignment for each major component of the course. The milestone-based project will be a group project with specific design/development milestones that must be met throughout the course. Each project will be required to contain some element from each of the four major components of the course. A short presentation of the final outcomes of the project will be required. A weekly lab will provide tutorials to assist students in becoming familiar with the tools in the SDK. There will be no marks assigned in the lab. Student groups will be given a choice from a selection of project topics. For each of these, the general requirements will be provided. Examples include multi-device ping-pong, network tic-tac-toe, instant messaging, and location-aware search.

Representative Course Outline

- iPod Touch/iPhone development
 - Xcode IDE
 - iPhone simulator
 - interface builder
 - testing and debugging
- Software Engineering for a mobile environment
 - extreme programming
 - agile modeling and design
- Network computing in a mobile environment
 - location services
 - context-aware mobile computing
- Graphics programming for mobile devices
 - OpenGL ES
 - game design for mobile devices
- Human-computer interaction for mobile devices
 - multi-touch
 - motion sensor

Comments or Notes

- In addition to three hours of lectures per week, there will be a laboratory of one and one-half hours per week.

COMP 4770 (W) Team Project

Students Interested

This course is required for all computer science majors.

Objectives of the Course

The objective of this course is to develop a working prototype of a software system as a team effort. A group of students will work on a project for a term, experiencing the advantages and difficulties of team projects.

Prerequisite(s)

COMP 3715, COMP 3716, COMP 3724 and COMP 3754

Successor(s)

None

Representative Workload

| | | |
|-----------------------|-----|---------------------|
| Requirements Document | 25% | (team effort) |
| Architecture Document | 15% | (team effort) |
| Module Document | 35% | (individual effort) |
| System Demonstration | 25% | (team effort) |

Representative Course Outline

The following milestones are required for a COMP 4770 project:

- Requirements Document
 - This document must describe the use cases and functional and non-functional requirements of the software system, as well as a schedule of the development tasks and the team member(s) responsible for each task.
 - Use cases are a requirements capturing technique.
- Architecture Document
 - The document must present the system's decomposition into modules and the assignment of responsibility of team members to subsets of the modules.
- Module Documents
 - These documents must include the complete description of the functionality and interface of each module. The module testing plans must be described and demonstrated.

- System Demonstration
 - The integrated system must be demonstrated by the team. The integrated system includes all software and data required to use the system. It contains an integration of all the modules.
 - A document describing the system testing must be submitted.

The milestones should be reached in the 4th, 6th, 9th and 13th week (respectively) of the semester.

Comments or Notes

- Students register for Computer Science 4770 as a regular three-hour per week course.
- Prior to the beginning of the term, registered students must submit a short CV indicating any experience or expertise relevant to the course.
- In the time slot assigned to the course, groups will meet to discuss their progress. A graduate student may be assigned to each group for help and advice.

CC2001 modules

SE8 (3/3)

COMP 4780 (F, W, S) Honours Project

Students Interested

This course is required of computer science majors who have been accepted into an honours program.

Objectives of the Course

To introduce computer science honours students to research activities, to familiarize these students with a special problem in computer science and to provide independent study on an advanced topic under the direct supervision of a member of the computer science faculty.

Prerequisite(s)

Admission to the honours program and permission of the Head of the Department. All required core courses and certain 4000 level computer science courses. Students normally register for this course during their last semester in the program.

Successor(s)

None

Representative Workload

- Students are expected to spend as much time on this course as is spent on any 4000-level computer science course. An honours student should meet regularly with his/her chosen supervisor to discuss the progress of the work and the future direction it should take.
- The topic is decided in consultation with the supervisor. The student is required to produce a written report on the project, to include the literature search on the topic and to present this work at a departmental seminar prior to the last week of the semester. The final grade received will depend on the written report, the seminar presentation and the discussions held between the student and supervisor, with the weighting of each of these to be at the discretion of the supervisor.
- No grade will be submitted for this course until all of the required work has been completed.

Representative Course Outline

Not applicable

Comments or Notes

- Registration for this course is by Course Change form only.

- An honours degree offers the student greater specialization in a particular area of computer science and as such, requires higher than average academic achievement. An honours degree is frequently a requirement for admission to an M.Sc. program. Any student interested in enrolling in an honours program is advised to consult with the head of department at his/her earliest convenience.
- For the degree **Honours in Computer Science (Software Engineering)**, the honours project must be in the area of software engineering.

COMP 4800-4825 Special Topics

Students Interested

Those who want a more indepth study in a specified area.

Objectives of the Course

To give students exposure to current and topical information.

Prerequisite(s)

Special topics courses are not offered on a regular basis, but whenever departmental resources permit. For this reason the prerequisites can vary each time they are offered.

Successor(s)

None

Representative Workload

Variable

Representative Course Outline

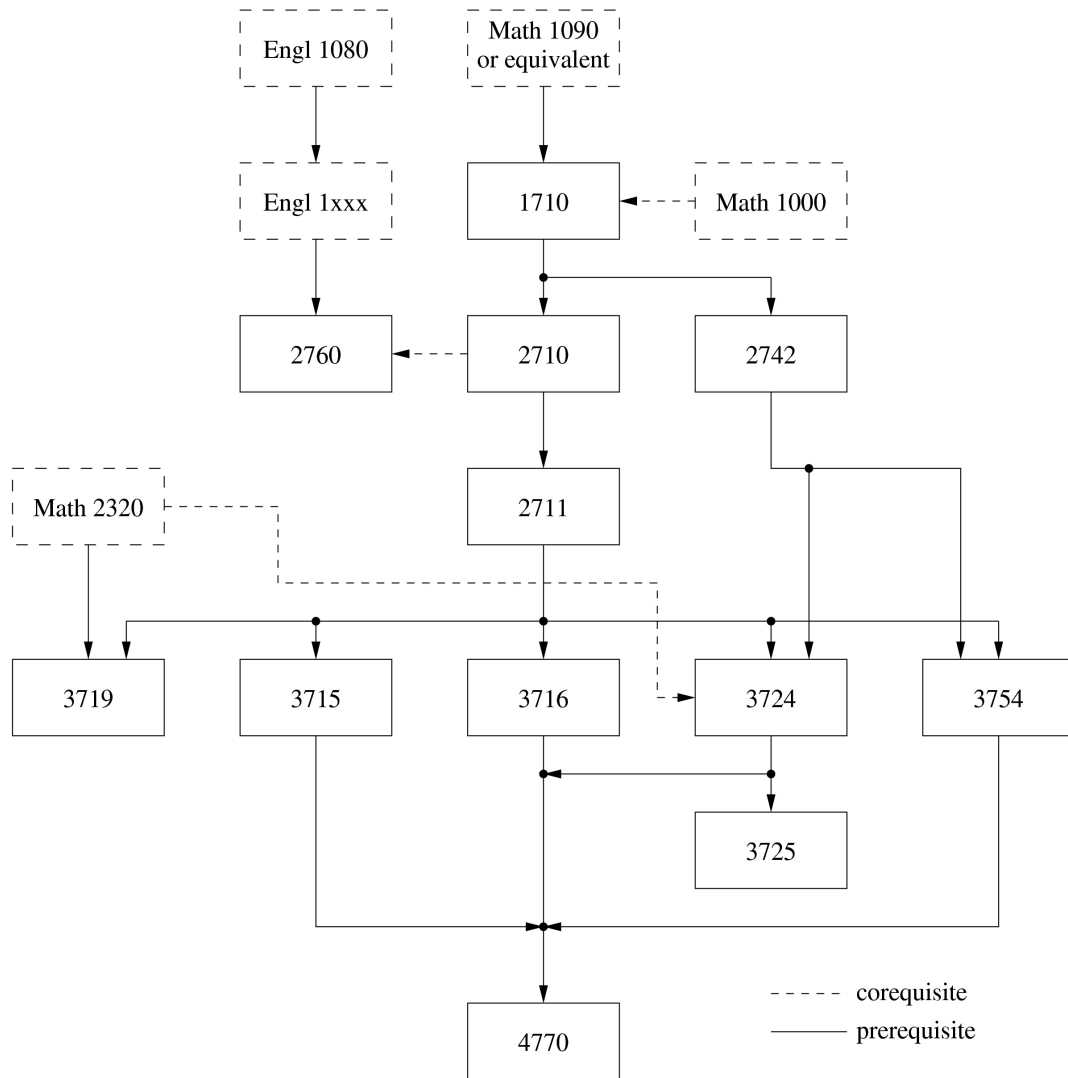
Variable

Comments or Notes

- All relevant information on a special topics course is posted on the Computer Science website and the bulletin board outside the Computer Science General Office several weeks before the beginning of the semester in which the course is being offered.

4 Appendix

4.1 Prerequisite Structure of Core Courses



Note:

- Students are advised to check the specific prerequisite(s) of each elective course.

INDEX of Computer Science Courses

| | |
|--|-----------|
| COMP 1510 - An Introduction to Programming for Scientific Computing | 38 |
| COMP 1550 - Introduction to Multimedia Application Development | 40 |
| COMP 1600 - Basic Computing and Information Technology | 42 |
| COMP 1700 - Introduction to Computer Science | 44 |
| COMP 1710 - Object-Oriented Programming I | 46 |
| COMP 2000 - Collaborative and Emergent Behaviour | 50 |
| COMP 2500 - Data Analysis with Scripting Languages | 52 |
| COMP 2602 - Computer Programming in Fortran | 53 |
| COMP 2650 - Problem Solving with Personal Computers | 55 |
| COMP 2710 - Object-Oriented Programming II | 57 |
| COMP 2711 - Introduction to Algorithms and Data Structures | 59 |
| COMP 2742 - Logic for Computer Science | 61 |
| COMP 2760 - Encountering the Computer: Society and the Individual | 62 |
| COMP 2801 - Introductory Computing for Business | 64 |
| COMP 3550 - Introduction to Bioinformatics | 68 |
| COMP 3700 - Industrial Experience | 71 |
| COMP 3710 - Vocational Languages | 73 |
| COMP 3714 - Programming Languages and their Processors | 74 |
| COMP 3715 - Network Computing with WEB Applications | 75 |
| COMP 3716 - Software Methodology | 77 |
| COMP 3718 - Programming in the Small | 78 |
| COMP 3719 - Theory of Computation and Algorithms | 80 |
| COMP 3724 - Computer Organization | 82 |
| COMP 3725 - Computer Architecture and Operating Systems | 83 |
| COMP 3731 - Introduction to Scientific Computing | 85 |
| COMP 3753 - Computational Aspects of Linear Programming | 87 |
| COMP 3754 - Introduction to Information and Intelligent Systems | 88 |
| COMP 4711 - Structure of Programming Languages | 92 |
| COMP 4712 - Compiler Construction | 93 |
| COMP 4715 - Special Topics in Programming Languages | 94 |
| COMP 4718 - Survey of Software Engineering | 95 |
| COMP 4721 - Operating Systems | 97 |
| COMP 4723 - Introduction to Microprocessors | 99 |
| COMP 4726 to 4729 - Special Topics in Computer Systems | 101 |
| COMP 4734 - Matrix Computations and Applications | 102 |
| COMP 4736 to 4739 - Special Topics in Numerical Computations | 103 |
| COMP 4740 - Design and Analysis of Algorithms | 104 |
| COMP 4741 - Formal Languages and Computability | 105 |
| COMP 4742 - Computational Complexity | 107 |
| COMP 4743 - Graph Algorithms and Combinatorial Optimization | 109 |
| COMP 4745 to 4749 (excluding 4748) - Special Topics in Theoretical Aspects | 110 |
| COMP 4751 - Computer Graphics | 111 |

| | |
|---|------------|
| COMP 4752 - Introduction to Computational Intelligence | 112 |
| COMP 4753 - Artificial Intelligence | 114 |
| COMP 4754 - Database Systems | 115 |
| COMP 4756 - Image Processing | 116 |
| COMP 4759 - Computer Networks | 117 |
| COMP 4762 - Introduction to Computational Molecular Biology | 118 |
| COMP 4766 - Introduction to Autonomous Robotics | 119 |
| COMP 4767 - Information Visualization and Applications | 120 |
| COMP 4768 - Software Development for Mobile Devices | 122 |
| COMP 4770 - Team Project | 124 |
| COMP 4780 - Honours Project | 126 |
| COMP 4800 to 4825 - Special Topics | 128 |