

Computer Science

Computer science is a subject related to almost every area of study known to modern man, the arts (computer-generated art, verification of historical documents, syntax study of languages including translation, etc.), social sciences (correlating experimental data, simulation, artificial intelligence studies), natural sciences (has always had application in this area), business and government (the largest single groups of users), education (computer-aided instruction, artificial intelligence), medicine, etc.

The department offers a wide selection of programs, ranging from broad to specialized:

- 1) Master's Degree Program
- 2) Undergraduate B.Sc. Degree Programs:
 - B.Sc, Honours in Computer Science
 - B.Sc, Major in Computer Science
 - B.Sc, Major in Computer Science with a Concentration in Imaging and Digital Media
 - B.Sc, Major in Computer Science with a Concentration in BioInformatics
- 3) Minor in Computer Science
- 4) A multidisciplinary Bachelor of Arts with a Major in Information Technology (details to be found in the Multidisciplinary Programs section at the end of the calendar)
- 5) Certificate Programs
 - Certificate in Computer Science
 - Certificate in Software Technology

MASTER'S DEGREE PROGRAM

Entrance Requirements

The minimum requirements for admittance to the Master's program are an undergraduate degree with a major in Computing Science or equivalent, and a high upper-second class standing.

Candidates with high academic standing in an undergraduate degree other than computer science, who have some computer science background either academic or professional, may be admitted as graduate preparatory students. Preparatory students will be asked to complete up to one year of undergraduate courses to enhance their background.

In addition to academic performance, a combination of factors is taken into consideration in assessing the eligibility of a candidate for admission into graduate programs.

One important such a factor is the availability of a faculty member competent and willing to supervise the academic program of studies and research of the candidate; a candidate will not be admitted to the program under any circumstance unless such a faculty member exists.

Other factors include the performance of the candidate and the assessment provided by his/her referees as a measure of the likelihood that the candidate can successfully complete the course of studies and research.

Program Requirements

Graduate students should familiarize themselves with the University and divisional calendar and regulations. Some of the information herein is adapted from these regulations, but is not intended as a replacement.

Completing the degree normally requires four one-term 500-level courses, registration and participation in the Graduate Seminar (CSC 597 and CSC 598), together with a Master's thesis (CSC 599). Courses are chosen by students in consultation with their supervisor. All courses prescribed for a student's approved program of study are designated as primary. Courses additional to the student's

Parallel computational models and their properties will be presented. Other typical topics include: complexity classes, and the parallel computation thesis.

Computer Science 509ab Pattern Recognition 3-3-0

This course addresses the statistical pattern classification theory and techniques. The material presented is concerned with discrimination and classification in the context of computer vision applications. The course topics include Bayesian decision theory, maximum likelihood estimation, nonparametric techniques, stochastic methods and unsupervised clustering.

Computer Science 510 Model-Based Testing of Reactive Systems 3-3-0

Testing is the primary hardware and software verification technique used by industry today. Usually, it is ad hoc, error prone, and very expensive. In recent years, however, many attempts have been made to develop more sophisticated, formal testing methods. The goal of this course is to provide an in-depth exposure of this emerging area. Students are expected to participate in the presentation of the lecture material and perform independent research.

Computer Science 512 Mathematical Models in Image Processing 3-3-0

Cross-listed with CSC412. Image processing is a rapidly growing field. As such, it requires and necessitates a number of mathematical models and domains to achieve efficient processing algorithms. Designing a successful processing technique invariably relies on having a successful model for images themselves. The mathematical techniques needed could range from Partial differential equations, Differential geometry, Morse theory, Topology, Algebraic topology, Wavelets, Statistical techniques, Calculus of variations, Numerical methods, Graph theory, and Optimization. The objective of this course is to discuss in depth a number of selected mathematical topics (and their use in image processing) that are of interest to the students at the moment the course is given.

Computer Science 513 Image Analysis 3-3-0

Cross-listed with CSC413. Image analysis is concerned with the development of machine algorithms in order to mimic the biological organism's ability to see and understand images and videos. The course content includes: camera models and calibration, image enhancement, features extraction and representation, shape from shading, stereo and texture, optical flow, motion analysis, high level vision and case studies.

Computer Science 514 Computer Networks & Distributed Algorithms 3-3-0

Cross-listed with CSC 414. The course presents computer networks at a functional level, with strong emphasis on programming distributed applications over a network. Discussion will be based on open networking and application standards such as the TCP/IP protocol suite and the Portable Operating System Interface (POSIX). The concept of distributed algorithms together with the associated challenges and examples are then presented. Programming distributed applications (in C or C++) is an integral part of the course. Students are expected to work on a large, distributed, and practically meaningful application as part of the course.

Computer Science 515 Concurrent & Real-Time Systems 3-3-0

Concurrent systems are complicated; they consist of many components which may execute in parallel, and the complexity arises from the multitude of combinations in which they interact. Concurrency by its very nature introduces phenomena not present in sequential systems, such as deadlock and livelock. This course provides an introduction to an algebraic language (such as CSP) for specification, analysis, and verification of concurrent and real-time systems. It then goes on and presents the use of such a language as a formal method for concurrency at different stages in the development process.

Computer Science 516 Volumetric Image Analysis & Visualization 3-3-0

Digital volumetric images are stacks of two dimensional image slices produced for instance by tomographic scanner. The goal of this course is to study the different techniques and algorithms for the analysis of volumetric images including a discussion about some sources of volumetric images, especially those occurring in medical imaging with different modalities (Radiology, Computed Tomography, Magnetic Resonance Imaging, Nuclear Medicine, Ultrasound, Positron Emission Tomography). The course will also address the different techniques used to display and visualize volumetric images including volume slicing, surface rendering, and volume rendering.

Computer Science 517 Database Software Design 3-3-0

Cross-listed with CSC 417. This course covers how one can implement a Database Management system. Major topics are Storage management, Query processing, and Transaction management. As a basic assumption, data will not all fit in main memory, so algorithms and data structures appropriate for effective disk storage and quick access must be used. For example, one may use index structures such as B-trees or hash tables. We cover parsing of queries and optimizing of query plans. Finally, we cover durability of transactions using logging, and concurrency control for isolation of transactions. Additional topics in distributed databases are also presented. Students are expected to prepare a research paper during the course, or pursue a larger applied project.

Computer Science 519 Special Topics in Computer Science 3-3-0

Cross-listed with CSC419. The course will present topics of current interest in Computer Science. The course content varies reflecting the interests of the faculty. Students are expected to participate in the presentation of the lecture material and engage in independent research.

Computer Science 596 Research Topics in Computer Science 3-3-0

This course provides an introduction to the primary and secondary sources of information in the computing science literature. Faculty discuss their own research objectives and present an overview of research issues in the major subject areas of Computer Science. Students are required to submit and present a paper on a topic that relates to their research.

Computer Science 597f Graduate Seminar 6-0-0

Students are expected to participate in the departmental seminars and give at a minimum two presentations (one outlining their thesis proposal, and another one about their thesis work). All Master's students are normally expected to enroll in this course in their first year in the program. Students will not receive credit for both this course and CSC 598 at the same time.

Computer Science 598f Graduate Seminar 6-0-0

Students are expected to participate in the departmental seminars and give at a minimum two presentations (one outlining their thesis proposal, and another one about their thesis work). All Master's students are normally expected to enroll in this course in their second year in the program. Students will not receive credit for both this course and CSC597 at the same time.

Computer Science 599 Master's Thesis 21-0-0

UNDERGRADUATE BSC DEGREE PROGRAMS

Entrance Requirements

To enter a Computer Science program, a student must normally have a Quebec collegial diploma (D.E.C.) in science, or the equivalent, including Differential Calculus, Integral Calculus (Calculus I and II), Mechanics and Electricity and Magnetism (Physics I and II). Students having these prerequisites can normally complete their undergraduate programs in three years. Students lacking the above Math and Physics courses will have to take their equivalent at Bishop's, in addition to their university requirements.

Students entering four-year degree programs (after completion of Ontario OSSD, grade 12 in other provinces or the US or equivalent) must register in their first year in the required collegial equivalent science courses: Mathematics 191 and 192; Physics 191, 192, 081 and 082; 6 credits of Humanities which includes 3 credits of English (ELA116 or other) and 3 credits in either English, History, Classical Studies, Philosophy, Religion, Liberal Arts.

Mature students are admitted to a 3-year program, in addition to the Math and Physics collegial courses (Math 191, 192, Phy 191, 081, 192, 082).

Students who possess another University degree recognized by the Quebec government, and who wish to obtain a B.Sc. in Computer Science, are normally admitted in a 30-credit-minimum program. However, they have to satisfy the program requirements: the Math and Physics prerequisites, 44 Computer Science credits, 12 Math credits and Phy 101.

All students in B.Sc. programs must complete the English Writing Proficiency requirement (EWP) before graduation. Three credits for the EWP requirement are added to the total credit requirement for each program below.

Program requirements:

Honours in Computer Science

A. Dissertation-based Honours

Total of 95 credits (plus the EWP):

at least 56 CSC credits:

44 required: CSC 101, CSC 111, CSC 116, CSC 121, CSC 204, CSC 211, CSC 217, CSC 275, CSC 303, CSC 309, CSC 405, CSC 499

15 electives must include at least 5 courses total and 9 credits from 300-level courses and above

15 MAT credits

12 required: MAT 105, MAT 106, MAT 107, MAT 108

3 elective

3 PHY credits PHY 101

18 free electives

B. Course-based Honours

The course-based stream does not require a dissertation (i.e. CSC407) but requires 3 more CSC courses. It is primarily designed for students wishing a specialization in Computer Science but are not interested in research and do not intend to pursue graduate studies

38 cr. CSC required CSC 101, CSC 111, CSC 116, CSC 121, CSC 204, CSC 211, CSC 217, CSC 275, CSC 303, CSC 309, CSC 405

24 cr. CSC electives must include at least 8 courses total, and 15 credits from 300-level courses

15 cr. MAT

12 required MAT 105, MAT 106, MAT 107, MAT 108

3 elective

3 cr. PHY PHY 101

15 cr. free electives

95 Total (plus EWP)

Honours in Computer Science, Concentration in Imaging and Digital Media - at least 117 credits total + EWP:

This program requires completing up to 30 credits at Université de Sherbrooke. This program will provide outstanding students with the possibility to participate in the cooperative program in place at Université de Sherbrooke.

This program includes a number of new courses in computer science and mathematics needed in Imaging and Digital Media. Students completing the requirements of this program will also qualify for a minor in Mathematics. With a judicious choice of electives, students can easily fulfill requirements for a major in Mathematics.

Stages: 3 (possibility of 4)

CSC credits:

at least 35 required CSC 101, CSC 111, CSC 116, CSC 121, CSC 204, CSC 211, CSC 217, CSC 218, CSC 303, CSC 309

2 courses from the following:

CSC 207, CSC 215, CSC 216, CSC 306, CSC 310, CSC 405, CSC 414, CSC 471

Imaging credits:

28 required CSC 208, CSC 302, CSC 412, CSC 413, IMN 117, IMN 317, IMN 459, IMN 467, IMN 517

6 electives from the following:

IMN 538, IMN 637, IMN 638, IMN 659, IMN 697

MAT credits:

21 required MAT 105, MAT 106, MAT 107, MAT 108, MAT 213, MAT 217, MAT 301

3 electives from the following:

MAT 109, MAT 115, MAT 210, MAT 211, MAT 214, MAT 215, MAT 225

3 PHY credits: PHY 101

15 Free Options

NB: IMN (Imagerie et Médias Numériques) are imaging and digital media courses available in French at Université de Sherbrooke.

Conditions and restrictions for Honours programs

- a student is initially admitted to the Major program
- After a minimum of 1 semester, a student with a grade of at least 80% in required courses may request entry to the Honours degree
- the dissertation stream and the Concentration in Imaging and Digital Media require, in addition, departmental permission.
- students must maintain an average of 80% in required courses to stay in the program

Major in Computer Science

There are three possible streams that may be followed to obtain a Computer Science Major. The first is a broad program that provides the traditional courses expected in a CSC major, and can easily be combined with any other Major. The second is specialized in Imaging and can only be enrolled into by approval of the department. The third is specialized in BioInformatics, and combines courses both from CSC as well as Biological Sciences.

1) Major in Computer Science — Total of 95 credits (plus EWP):

at least 47 CSC credits:

32 required: CSC 101, CSC 111, CSC 116, CSC 121, CSC 204, CSC 211, CSC217, CSC 303, CSC 309

15 electives must include at least 5 courses total

12 MAT credits

6 required: MAT 105, MAT 108

6 elective

3 PHY credits PHY 101

33 free electives

2) Major in Computer Science, Concentration in BioInformatics – at least 95 credits + EWP

The following program is based on a 3-year format for students possessing a CEGEP DEC in Sciences. Students admitted into a 4-year program will have to take the following prerequisites and their associated labs, normally in their first year (U0): Bio 191, Che 191, 192, Mat 191, 192, Phy 191, 192, Ela 116 (or another English literature course), a Humanities elective course (chosen from 100-level offerings in Cla, Eng, His, Lib, Rel and Phi), and an additional free elective.

a) CSC/MAT/PHY courses (at least 29 credits):

23 credits required:

CSC 111, CSC 204, CSC 205, CSC 217, CSC 327

MAT 105

PHY 101

6 credits taken from any of:

CSC 201 (or BIO 252), CSC 207, CSC 208, CSC 216, CSC 310

MAT 108

BMS 325

b) BIO/BCH/CHE courses (36 credits plus associated labs):

30 credits required, plus associated lab credits:

BIO 110, BIO 115, BIO 116, BIO 118, BIO 212, BIO 314 (or BCH 341), BIO 360 / BCH 370, BCH 211

CHE 105, CHE 106

6 credits taken from any of:

BIO257, BIO291, BIO367,
BCH336, BCH461

c) Bioinformatics (9 credits):

BCH 342

Stage in Bioinformatics (6 credits)

d) 21 credits of free electives

MINOR IN COMPUTER SCIENCE

25 CSC credits

11 required: CSC 101, CSC 111, CSC204

15 electives (must include at least 5 courses total, and must not include CSC 102)

General Notes/Restrictions:

1. Only one of CSC304, CSC307 or CSC499 may be taken for credit, unless with a special departmental authorization
2. Computer Science courses that are double-listed in Math cannot be counted toward fulfilling the Math electives required for the Computer Science Honours/Major.
3. Students are encouraged to take their free electives from outside the department.

**B.A. PROGRAM, MAJOR IN
INFORMATION TECHNOLOGY**

Please see the details of the program in the Multidisciplinary Programs section at the end of the calendar.

CERTIFICATE PROGRAMS – at least 32 credits

Description and objectives:

The Certificate Programs in Computer Science are designed for individuals who need to acquire a basic understanding of computers and programming and knowledge of the field in order to expand their area of interest and professional expertise. Topics include: Communications, Software Engineering, Graphics and Artificial Intelligence. This program will help students to take full advantage of the computer technology available in the workplace.

Prerequisites to programs:

Applicants insufficient Math background might be required to take an additional 3-credit Math course in their first semester (Math 190 or equivalent).

Program description

A) Certificate in Compute Science

Required courses : 19 credits

CSC 101ab	Foundations of Computer Science	4-3-3
CSC 111ab	Programming Methodology	4-3-3
CSC 116ab	Low Level Programming Language	4-3-3
CSC 204ab	Data Structures	3-3-3
CSC 211ab	Computer Organization	4-3-0

Elective courses: at least 15 credits (5 courses) from any other Computer Science courses except CSC 102.

B) Certificate in Software Technology

Required courses: 17 credits

CSC 101ab	Foundations of Computer Science	4-3-3
CSC 111a	Programming Methodology	4-3-3
CSC 204ab	Data Structures	3-3-0
CSC 303a	Principles of Programming Languages	3-3-3

and at least one of:

CSC 218a	C++ Programming	3-3-3
CSC 328ab	Object Oriented Software Construction	3-3-1
CSC 316ab	Special Topics in Software	3-3-0
CSC 319a	Special Topics in Computer Science	3-3-0

Elective courses: at least 15 credits (5 courses) from any other Computer Science course except CSC 102.

Undergraduate Courses

Computer Science 101ab Foundations of Computer Science 4-3-3

An introduction to Computer Science and selected applications suitable for both majors and science non-majors who want a broad overview of the field. The course provides a layered introduction covering hardware, system software and applications packages. The course includes elementary programming. Topics include Algorithmic foundations of Computer Science; The hardware world: number systems, boolean logic, computer circuits, Von-Neumann architecture; System software: assembly language, operating systems, high level languages, language translation; Models of Computation; Applications and Social Issues

Note: Registration priority is given to Science and IT students. CSC students must take this course in their first year.

Computer Science 102ab Essentials of Computing on the PC
for Non-Scientists 4-3-3

An introduction to the use of the personal computer as a tool by non-scientists. No prior knowledge of computers is assumed and a learning by doing approach is adopted wherever possible.

Survey of the basic hardware: I/O devices, secondary storage, communications. Systems Software: DOS and Windows fundamentals from the user perspective. Basic hands-on applications: Word Processing, Spreadsheets, Database Processing. Information and Decision Support systems, Communications and connectivity: E-mail and the Internet, Internet Resources. Workplace issues, privacy and security.

Note: This course cannot be taken for credit by Science students or by students who have received credit for either BCS114 or CSC101. This course is not offered on a regular basis.

Computer Science 103 Interactive Web Page Design 4-3-3

In this course, students will learn the basics of HTML, the language describing web pages, and CSS, another web page language. By constructing fill-in forms and employing short sections of script, students will learn how to enable users of the Internet to interact with their web pages: sending data to be stored, and receiving customized responses. The course will include simple database operations. Extensive laboratory work will result in students creating their own set of personal web pages on a publicly accessible server. The course is open to anyone interested in the subject.

Computer Science 107ab Databases and Dynamic Web Design 4-3-3

In this course students will build dynamic websites using SQL and PHP, learn the web technology that powers a million e-commerce sites, and enables dynamic, interactive tools and applications. Many real-world websites, and especially e-commerce sites, are complex and need flexibility. As a result, the use of databases as a back-end for websites is increasingly popular. Queries in the SQL language allow a high degree of selectivity, as well as easy storage and retrieval of large image and media content objects. This course expands upon the techniques of web programming introduced in Computer Science 103, to teach the use of databases in web design. Applications are primarily drawn from the e-commerce area, but are not limited to this domain. Students will learn how to design and administer a database, set up mailing lists, build discussion

forums, create a storefront, and even build a working shopping cart. This course emphasizes practical skills with hands-on-projects.

Prerequisite: Computer Science 103 or permission of the instructor.

Computer Science 111ab Programming Methodology 4-3-3

This course introduces algorithms, data structures and software engineering principles. The use of a high level language is the tool to develop these components. By the end of the course, a successful student should be 'fluent' in programming, and have a good base for data structures. The course provides the necessary programming skills needed for further studies in Computer Science. Java is used as the programming language.

Computer Science 113ab Digital Imaging for Photography 4-3-3

The students in this course will learn the basic concepts of digital imaging as applied to photography. They will explore the techniques of acquiring, manipulating and outputting digital photographic images using a variety of modern hardware and software equipment. This course will also discuss lighting, color representation, gray scale and color adjustments, image filtering and special effects creation. The course is organized around assignments and a final project where the students will apply the studied techniques to their own photographic processes.

Computer Science 116a Low Level Programming Language 4-3-3

The goal of this course is to understand how a computer works at a low level. Programming assignments will be primarily in an assembly language, with programming in C for comparison. Topics: binary and hexadecimal number systems, a representative processor architecture, addressing modes, procedure calls, parameter passing, the use of stacks, in particular for recursive procedure calls. Comparison of the reference processor with other processors.

Computer Science 121b Advanced Programming Techniques 4-3-0

The course is intended to be a sequel to introductory programming with emphasis placed on the architecture of software. It will go in depth into object-oriented techniques, reusability, data abstraction, class design, and implementation, design and structure of class libraries. Topics to be covered include: polymorphism, encapsulation, overloading, inheritance and delegation, types of inheritance (Inheritance for Extension, Specialization and Specification), composition, aggregation and design of collections. Static and dynamic types, downcasting, exception handling. The second half of the course will be devoted to software design patterns, with particular emphasis on the observer, iterator, visitor and selected creational patterns. Course work will involve significant programming projects. The teaching language will be Java.

Prerequisite: CSC 111

Computer Science 201 Computer Ethics 3-3-0

Ethics is a branch of philosophy. Computers introduce arguably unique ethical issues in the way their use affects society. Technically minded professionals often give little attention to ethical issues. This course explores the basis for ethical reasoning, and examines ethical issues such as invasion of privacy, mischief including viruses, piracy and liability of software. It also considers broader issues of impacts on the individual and society, control of the technology, and the question of the difference between human understanding and rule-based processing of data. Students will be expected to participate in class discussions and role-playing scenarios, and to write a term paper.

Computer Science 204b Data Structures 3-3-0

An advanced course designed to expose the student to the latest programming theory and software engineering principles. Topics covered include modularization, data encapsulation, information hiding, data abstraction, and other object oriented software construction techniques will be discussed. Parallel design of algorithms and data structures, analysis of algorithms (including "big O" notation and software verification methods.) Standard data structures such as stacks, queues, trees and graphs will be examined. Programming examples are done in Java.

Prerequisite: Computer Science 111

Computer Science 205ab Data Mining for Scientists 4-3-3

Data is now created faster than humans are able to understand it and use it. There may be patterns hiding within this data with potentially useful information. This course will teach students, including Biology and Biochemistry students as well as those from Computer Science, how to discover these patterns for the purpose of solving problems, gaining knowledge, and making predictions. Topics covered in this course include data preparation, clustering, classification, association rules for mining and linear regression. This course includes assignments and a final project where the students are required to perform mining on real datasets drawn from the biological and physical sciences.

Prerequisites: PHY 101 (or equivalent)

See Physics 274ab

Students may not take this course for credit if they have received credit for Physics 274.

Computer Science 206b Functional and Logic Programming 3-3-3

There is much more than imperative programming. This course introduces two other programming paradigms, functional and logic. Topics normally include: functional programming languages, such as Lisp and Haskell; higher order functions, lazy evaluation, abstract and recursive types, structural induction, symbolic expressions; logic programming languages, such as Prolog; operational interpretation of predicates and terms, proof search, unification, backtracking; typical applications.

Corequisite: Computer Science 204b

Computer Science 207b Using and Designing Data Bases 4-3-3

This course presents data modeling (Entity-Relationship model, UML, etc.), relational algebra, normalization, SQL language. Implementation of databases using the relational model is discussed. Object-oriented modeling and implementation is also introduced. Other topics include: Concurrency control, transaction processing, client-server systems, distributed databases, and web-based delivery of data.

Prerequisite: Computer Science 204

Note: Students may not take this course for credit if they received credit for either BCS 214 (Jan 98 and onward) or CSC 274 (prior to 2003).

Computer Science 208b Scientific Programming 3-3-3

Scientific Programming is a course for students who want to learn more about the computing that goes on behind computational science. Students will learn the basic mathematical tools and computational techniques including the design and analysis of algorithms for solving mathematical problems that arise in many fields, especially science and engineering. Emphasis is placed on both the actual implementation and on the numerical and algebraic methods. The programming projects assigned in this course will make substantial use of C and C++ for numerical computations and Maple for symbolic computations.

Prerequisite: CSC 204, Math 191, Math 192

Note: See Mat 279b and PHY 278b. Students may not take this course for credit if they have received credit for Mat 279 or Phy 278.

Computer Science 211a Computer Organization and Logic Design 4-3-3

This is a theoretical course on computer organization and architecture. Different computer components and how they function are studied in detail. By the end of the course, students should be able to build (in theory) a small computer without interface. Topics covered are: boolean algebra and gates, combinational circuits (decoders, multiplexers, PLAs), logic design (flip-flops, shift registers, counters, sequential circuits), the ALU, memory (RAM, ROM, secondary storage), I/O Devices and the control unit (hardwired, microprogrammed). For those interested students, a follow-up course, largely consisting of lab experiments, is CSC212b.

Prerequisite: Computer Science 111ab, Computer Science 116ab

Computer Science 212b Microcomputer Interfacing 4-2-4

This course and integrated laboratory introduces the techniques used to interface a microcomputer to the real world with a robot as the main interface. Students will complete projects of increasing difficulty as they build and program a robot to accomplish a given task and brief the class on their design and findings. Topics will include: interactive programming, analog and digital inputs, use of the bus and registers to control output signals, simple electronic sensors, multiplexing and decoding and practical problem solving.

Prerequisite: Computer Science 211a or equivalent programming and electronic knowledge.

Computer Science 214 Introduction to Networks 3-3-0

This course introduces and discusses the components and architectures of computer networks. Topics to be covered include: Resources Sharing (Network Interface Circuitry, File Servers, Workstations, etc.), Network Protocols (TCP/IP, Apple Talk, Novell, etc.) and Network Infrastructure (Hubs, Routers, Gateways, Bridges, etc.).

Computer Science 215b Data Communications 3-3-0

This is a theoretical course on Data Communications. It covers the basic and physical aspects involved when data is transmitted from one point to another, such as analog vs. digital transmission, various forms of encoding analog and digital data into appropriate signals, error detection techniques, multiplexing, etc... As well as an introduction is given on networking techniques, differences between circuit and packet switching, routing techniques, and Local Area Networks. The course is of mathematical and physical nature.

Prerequisites: Computer Science 111ab, Computer Science 116a

Note: See PHY 265. Students may not take this course for credit if they have received credit for PHY 265.

Computer Science 216a Artificial Intelligence 3-3-3

A course aiming to introduce students to the basic concepts and techniques of Artificial Intelligence. Topics will include: Search strategies; knowledge representation; AI languages; Rule-based inference systems, expert systems; computer vision; planning and problem solving; natural language understanding.

Prerequisite: Computer Science 204, Computer Science 206b or Computer Science 303

Computer Science 217a Design and Analysis of Algorithms 3-3-0

This course is intended to make students familiar with most of the existing techniques for problem solving. It starts with an introduction to algorithms efficiency, solving recurrence relations and basic data structures. Then different techniques for algorithms design are discussed; the divide-and-conquer technique, the greedy technique and its applications to graph algorithms, dynamic programming, backtracking and branch and bound algorithms. With every technique presented, examples from different domains are studied and their algorithms analyzed. At the end, students are briefly introduced to the vast area of "difficult" problems, or NP-complete.

Prerequisite: Computer Science 204 and MAT 105

Note: see MAT 277. Students may not take this course for credit if they received credit for MAT 277.

Computer Science 218a Advanced C++ Programming 3-3-3

The C++ language has become an industry standard as an implementation language. The course aims at introducing the student to intermediate and advanced programming using C++, with particular emphasis on systems software and the use of the C++ object-oriented extensions in software engineering

C++ Programming basics (loops and decisions, arrays, structures, functions, pointers) Objects and Classes, Inheritance, Virtual functions, files and stream, I/O Structure and design of Class libraries, Standard Template Library, OOP Design basics. Although no prior experience in C is required, it is assumed that the student is already fluent in some other programming language and in the programming of data structures.

Prerequisite: Computer Science 204

Computer Science 272b Electric Circuits and Electronics 3-3-3

Review of D.C. circuits, Kirchoff's laws, network theorems. Network analysis for A.C. circuits, phasors. Diode circuits and filters. The physical basis of semiconductor devices including semiconductor diodes, junction transistors, and field-effect transistors. The operation of transistor amplifiers, digital electronics and integrated circuits will also be covered.

Note: See Phy 212. Students may not take this course for credit if they have received credit for Physics 212.

Computer Science 275b Numerical Methods 3-3-0

Numerical techniques for problem solving in mathematics, computer Science and Physics. Error analysis, roots of equations, QR-algorithm, interpolation, Numerical approaches to differentiation, integration and solutions of differential equations.

Prerequisite: Computer Science 111ab, Mathematics 107 and 108.

Note: See Mat 225 and Phy 275. Students may not take this course for credit if they have received credit for Mathematics 225 or for Phy 275.

Computer Science 301a Simulation Techniques 3-3-3

Computer simulation is defined and put into the context of other simulation methods. Two main techniques are studied, one involving automated spreadsheets (financial modelling) and the other queuing theory. A term project involving the simulation of an actual system is part of the course.

Prerequisite: Computer Science 204, PHY 101 (or equivalent)

Note: Students may not take this course for credit if they received credit for BMS 343. This course will be offered in alternate years.

Computer Science 302a Computer Graphics 4-3-0

This is an introductory course to the principles of interactive raster graphics. Topics include an introduction to basic graphics concepts, scan conversion techniques, 2-D and 3-D modeling and transformations, viewing transformations, projections, rendering techniques, graphical software packages and graphics systems. Students will use OpenGL graphics API to reinforce concepts and study fundamental computer graphics techniques. Prerequisites: Computer Science 204b, MAT 108a

Computer Science 303a Principles of Programming Languages 3-3-0

The objective of this course is to introduce, analyse and evaluate, on a comparative basis, the concepts on which programming languages and their implementations are based. Topics to be covered: definition of languages, syntax and semantics; Compilation techniques, top-down parsing, creating a parser; variables and binding, expressions, statements; data types; procedures scope, and run-time considerations; coroutines; implementation of block-structured languages; modularity and abstractions; concurrency exception handling and program correctness; functional programming object-oriented programming languages; logic programming and constraint languages.

The languages ML, Eiffel, Lisp, Scheme, Prolog, Haskell, 02, Java and Smalltalk will be used to illustrate the above concepts.

Prerequisites: Computer Science 204 and one other Programming Language course

Computer Science 304 Project 3-0-3

This course is normally taken by CSC students in their final year. The project must be approved in advance by the department. Students will be expected to submit a written report and to make a presentation.

Prerequisite: approval of the dept., 80% in CSC courses

Computer Science 306b Compilers and Interpreters 3-3-0

This course is intended as an introduction to the fundamentals of language translation and compiler construction. Topics will include language theory and syntax; grammars, finite state machines, non-deterministic push-down automata; a thorough treatment of parsing methods covering top-down, bottom-up and precedence parsers; Syntax directed translation; Run-time environments; optimization and error recovery; code generation. Students will be required to construct a working interpreter of a Pascal-like language.

Prerequisite: Computer Science 303a

This course will be offered on alternate years.

Computer Science 308 Project II 3-0-3

This course is normally taken in the final year of studies and may involve work on a theoretical topic or a practical implementation of a sizable software project. The topic must be approved in advance by the department. Students are expected to attend bi-weekly project meetings where they present and discuss their work. In addition, they will make a final presentation at the end of term and submit a report.

Prerequisite: approval of the dept. 80% in CSC courses.

Computer Science 309b Principles of Operating Systems 3-3-0

Basic concepts of computer hardware; program translation linking and loading; cooperating sequential processes; critical section problem, process synchronization primitives, parallel programming; introduction to multiprogramming; operating system nucleus; file systems; reliability and protection; system performance, measurement and evaluation. Memory Management. Paging and Virtual memory. Unix. Using and programming the Unix Shell, Unix implementation. Examination of the implementation of Unix clones Minix, Linux, Survey of state-of-the-art operating systems. Distributed Systems, Communication and synchronization in distributed systems. Theoretical issues and implementation.

Prerequisites: Computer Science 204b

Computer Science 310b Software Engineering 3-3-3

Software is an engineered product that requires planning, analysis, design, implementation, testing and maintenance. This course is a presentation of the techniques used in each step of the software product process. Topics: software requirements analysis and specifications; software design process, object oriented design; testing, reliability and maintenance; automated design tools, programming environments. Students will be expected to work jointly on several large software projects.

Prerequisites: Computer Science 204b, Computer Science 303a

Computer Science 311 Advanced Computer Architecture 3-3-0

The focus in this course is on basic principles, current practice, and issues in computer architecture and organization. At the end of the course students will have gained an understanding of how a computing system is organized, as well as why it is organized this way. The relation between hardware and the software that runs on it is emphasized, leading to an intuitive understanding of how the behavior of applications influences computer organization and design. Topics covered typically include (but are not limited to): instruction set design, microprogrammed versus hardwired processors, pipelining and superscalar processors, memory organization (cache, primary, virtual), I/O and interrupts, multiprocessors. Comparative critical and quantitative analyses of various systems that currently exist are presented.

Prerequisites: CSC 211 or instructor's permission.

All special topics courses require CSC111 and CSC204 as a prerequisite and/or any other course as specified by the instructor.

Computer Science 315ab Special Topics in Communications 3-3-0

The course will present topics of current interest or research directions in Computer Communications Networking and network programming. The course content is expected to vary to reflect the current interests of students and faculty. It will be offered by arrangement with the department.

Computer Science 316ab Special Topics in Software 3-3-0

The course will present topics of current interest or research directions in Software Science. The course content is expected to vary to reflect the current interests of students and faculty. It will be offered by arrangement with the department.

Computer Science 317ab Special Topics in Computer Applications 3-3-0

The course will present topics of current interest or research directions in Computer Applications. The course content is expected to vary to reflect the current interests of students and faculty. It will be offered by arrangement with the department.

Computer Science 318b Topics in Computer Science 3-3-3

The course will present topics of current interest or research directions in Computer Science. The course content is expected to vary reflecting the interests of the students and the faculty. It will be offered by arrangement with the department.

Computer Science 320b Concurrent Programming 3-3-3

Introduction to the principles and practice of concurrent programming. A vital topic in the design and implementation of operating systems, distributed systems, and distributed data structures. In addition, with the advent of commercially available multiprocessors, it is growing in practical significance as well. Topics to be covered: programming notation and logics for concurrent programming, concurrency and synchronization, specification and semantics of concurrent execution, safety and liveness properties, critical section problems, data parallel processes, semaphores, conditional critical regions, monitors, message passing: synchronous-asynchronous. Remote procedure call, rendezvous. Language Overviews: Turing Plus, Occam, Ada, SR, Linda.

Prerequisite: Computer Science 309b

This course will normally be offered in the summer as an extension course.

Computer Science 327b Special Topics in Algorithms 3-3-0

The course builds on the techniques covered in CSC 217 to present some specialized algorithms in several areas, including Bioinformatics, Computational Geometry, and Network Flow.

Computer Science 328b Object-oriented Software Construction 3-3-0

Introduction to Object Oriented concepts and their application in programming and design. Encapsulation, polymorphism, dynamic binding and inheritance. Overview of Programming (C++, Smalltalk, Eiffel, Java and Scoops) and their implementations. Object Oriented Design, UML and Software design patterns. A survey of related fields such as Object Oriented Databases and Concurrent OOPLs. Practical programming assignments will be implemented in Java, Eiffel and/or C++.

Prerequisite: Computer Science 204b

Computer Science 400ab Independent Studies 3-0-0

Individual study and research under the guidance of an advisor and Department staff.

Prerequisite: Permission of the department

Computer Science 405 Theoretical Aspects of Computer Science 3-3-0

The course will include several of the following topics: Computational models, Computational complexity; Finite-state machines; Context-free languages; Pushdown automata; Turing machines; Undecidable problems.

Prerequisite: CSC 204, MAT 105

Note: See MAT 275. Students may not receive credit for this course if they have already received credit for MAT 275, nor can they receive credit for both CSC 305 and CSC 405.

Computer Science 412 Image Processing 3-3-0

This course will introduce the area of Image Processing and present classical tools and algorithms in the field including: image perception, image acquisition and display, histogram techniques, image restoration, image enhancement, primitive operations for image analysis, segmentation, image transforms, and pattern and object recognition. Some examples of industrial applications of image processing and some important developments in image processing research will be also addressed. No student shall receive credit for both this course and CSC 312.

Prerequisites: CSC204, MAT192, PHY101 (or equivalent)

Computer Science 413 Computer Vision 3-3-3

This course is concerned with the computer acquisition and analysis of image data. Computer vision is the construction of explicit, meaningful descriptions of a physical object from images. Emphasis will be placed on: camera models and calibration, image representation, pattern recognition concepts, filtering and enhancing, segmentation, texture, motion from image sequences, deformable models, matching, stereovision, perceiving 3D from 2D images and tracking with dynamic models. The programming projects assigned in this course will make substantial use of the C and C++ programming languages. No student shall receive credit for both this course and CSC 313.

Prerequisites: CSC 204, CSC 218, MAT 192, PHY 101 (or equivalent)

Computer Science 414ab Computer Networks 4-3-3

This course elaborates on the material considered in CSC 215 (Data Communications). It presents computer networks at a functional level, with strong emphasis on programming distributed applications over a network. Discussion will be based on open networking and application standards such as the TCP/IP protocol suite and the Portable Operating System Interface (POSIX). Topics normally covered are TCP/IP architecture and programming, the client-server model, network file systems, streaming, tunnelling. Programming distributed applications (in C or C++) is an integral part of the course. No student shall receive credit for both this course and CSC. 314.

Prerequisite: CSC 218 and CSC 215 or equivalent experience in computer networks.

Computer Science 417ab Database Software Design 3-3-0

This course covers how one can implement a Database Management system. Major topics are storage management, Query processing, and Transaction management. As a basic assumption, data will not all fit in main memory, so algorithms and data structures appropriate for effective disk storage and quick access must be used. For example, one may use index structures such as B-trees or hash tables. We cover parsing of queries and optimizing of query plans. Finally, we cover durability of transactions using logging, and concurrency control for isolation of transactions. Additional topics in distributed databases are also presented.

Computer Science 419ab Special Topics in Computer Science 3-3-0

The course will present topics of current interest or research directions in Computer Science. The course content is expected to vary from year to year to reflect the current interests of students and faculty. It will be offered by arrangement with the department.

No student shall receive credit for both this course and CSC 319

Computer Science 471ab Graph Theory 3-3-0

An introduction to the combinatorial, algorithmic and algebraic aspects of graph theory.

Prerequisite: MAT105, CSC204

Note: See MAT321. Students may not take this course for credit if they have received credit for MAT321, nor can they receive credit for both this course and CSC371.

Computer Science 499 Honours Dissertation 6-0-0

The student is required to complete a theoretical or applied project. The subject is arranged with the student's supervisor during the first four weeks of term. A written dissertation is required as well as two seminar presentation.

Note: This course is open only to final year Computer Science Honour Students in the dissertation stream, and only by permission of the department.