Computer Science

Faculty

Madjid Allili,
B.Sc.(Algiers), M.Sc., Ph.D.(Sherbrooke);
Professor

Layachi Bentabet,
B.Sc.(Eng.National Polytechnic, Algeria),
M.Sc.(Elec.Eng. Institut national des sciences appliquées, Lyon),
Ph.D.(Sherbrooke); Professor

Stefan D. Bruda,
B.Sc.Eng., M.Sc., Ph.D. (Queen's);
Professor

Russell Butler
B.Sc. (Bishop's), M.Sc., Ph.D. (Sherbrooke);
Associate Professor
Chair of the Department

Rachid Hedjam
B. Eng.(University of Setif, Algeria), M.Sc. (Montreal),
Ph.D. (ÉTS);
Assistant Professor

Yasir Malik,
M.Sc. (Ajou University, S. Korea), Ph.D. (Sherbrooke);
Associate Professor

Program Overview

Computer science is a subject related to almost every contemporary intellectual discipline, the arts (computer-generated art, verification of historical documents, syntax study of languages including translation, etc.), social sciences (correlating experimental data, simulation, artificial intelligence studies), natural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) Undergraduate B.Sc. Degree Programs

I. B.Sc, Honours in Computer Science

II. B.Sc, Major in Computer Science

2) A multidisciplinary B.A. with a Major in Information Technology (BAIT)

3) Minor in Computer Science

4) Certificate Program in Computer Science

Note: *A Co-op program is offered for all students in programs 2) and 3) above. Please refer to the Co-op section.*

5) Master's Degree Program (see *Graduate Programs* section)

I. Thesis Option

II. Project Option

Undergraduate B.Sc. Degree Programs

Computer Science Honours (120 credits) HONCSC

A. Dissertation-based Honours

18 credits:	Program prerequisites *(please refer to Table II in the Faculty section of the Calendar)*
57 CS credits:	39 required: CS 201, CS 211, CS 216, CS 304, CS 310, CS 311, CS 317, CS 321, CS 403, CS 409, CS 455, CS 499 18 electives: must include 12 credits from 400-level courses and above
12 MAT credits:	12 required: MAT 108, MAT 200, MAT 206, MAT 207
3 PHY credits:	PHY 101
3 credits:	Arts and Science requirement *(please refer to the Faculty section of the Calendar)*

27 credits of free electives

B. Course-based Honours

The course-based stream does not require a dissertation (i.e. CS 499) but requires 3 more CS 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:

18 credits:	Program prerequisites *(please refer to Table II in the Faculty section of the Calendar)*
60 CS credits:	33 required: CS 201, CS 211, CS 216, CS 304, CS 310, CS 311, CS 317, CS 321, CS 403, CS 409, CS 455 27 electives: must include 15 credits from 400-level courses
12 MAT credits:	12 required: MAT 108, MAT 200, MAT 206, MAT 207
3 PHY credits:	PHY 101
3 credits:	Arts and Science requirement *(please refer to the Faculty section of the Calendar)*

27 credits of free electives

General Notes for Honours

- after a minimum of 1 semester, a student with a grade of at least 80% in required courses may request entry to the Honours program
- the dissertation stream requires, in addition, departmental permission.
- students must maintain an average of 80% in required courses to stay in the program

Computer Science Major (120 credits) MAJCSC

18 credits:	Program prerequisites (please refer to Table II in the Faculty section of the Calendar)
45 CS credits:	30 required: CS 201, CS 211, CS 216, CS 304, CS 310, CS 311, CS 317, CS 321, CS 403, CS409 15 electives
9 MAT credits:	6 required: MAT 108, MAT 200 3 elective (MAT 19X cannot count as MAT elective)
3 PHY credits:	PHY 101
3 credits:	Arts and Science requirement (please refer to the Faculty section of the Calendar)

42 credits of free electives

Information Technology Major (B.A.) MAJITC

Information Technology (IT) is defined by the Information Technology Association of America (ITAA), as the study, design, development, implementation, support or management of computer-based information systems, particularly software applications and computer hardware. IT deals with the use of electronic computers and computer software to convert, store, protect, manage, transmit and retrieve data, securely.

This program provides the necessary skills and knowledge to work/design/participate within organizations that manage large amount of data and provide services to a large number of users. Students will develop skills and knowledge in Information Technologies, Management practices and Organizations, with the required fundamentals of Computer Science.

Note: Students following this degree program are not eligible to add a Business program

Program prerequisites: (12 credits)

Please refer to Table II in the Faculty section of the Calendar.

Core curriculum (36 credits):

CS 201, CS 211, CS214, CS 325, CS 304, CS 307
BCS 220, BCS 320, BCS 313, BCS 422,
BMA 140, BMG 100

Secondary Core [1] (30 credits)

A minimum of 3 courses in Computer Science.

A minimum of 3 courses in Business, normally chosen from the following list:
BCS 317, BCS 416, BCS 340, BCS 420,
BCS 430, BCS 450, BMG 214, BMG 323,
BMK 211, BMK 214, BMK 321, BMK 323,
BMK 333

[1] Students are advised to consult the Calendar for prerequisites

Arts and Science requirements (3 credits)

Please refer to the Faculty section of the Calendar

Free electives (39 credits)

Co-operative Education Program B.Sc. Coop

The co-operative Education Program combines a student's academic program with integrated work experiences through full-time work terms and regular academic sessions. The work terms are designed to present the students with the opportunity to blend theory and practice and to gain relevant work experience.

Each co-operative work term is between 12 and 16 weeks in length, and the student will be registered in a 3-credit Co-operative Placement course (CS 391, CS 392 or CS 393). These course credits count as free electives. Each is graded on a pass/fail basis and this grade is not included in the student's cumulative average. The evaluation is the responsibility of the Departmental Chair and will be based upon the submission of a work term report and a job performance report submitted by the employer. Normal academic regulations apply to the conduct and evaluation of the courses.

The number of work terms needed depends on the number of credits the students need to complete upon admission at Bishop's. Student who have been granted 30 advance credits (or more) will be required to complete two work terms (6 credits). Other students who have been admitted into a regular 120-credit degree program will be required to complete three work terms (9 credits). These credits will be added to the student's program and do not count as computer science courses, computer science electives, or free electives. All work terms must be completed before the student's final academic semester and a student's last semester before graduation cannot be a work term. While every effort will be made to find a suitable placement for all students in the program, no guarantee of placement can be made since the employment process is competitive and subject to market conditions.

Admission to the Co-operative Education Program

Students must submit an application to be admitted to the program. Full-time students in any Honours or Major program offered in the Computer Science Department who have completed the online application package, who have successfully completed BMG191 and who have a minimum cumulative average of 70% upon application are admissible into the Co-op Program. Students in the Co-op must maintain their 70% average and be full-time in order to stay in the program.

Work Term Registration

Once a student has signed the Co-operative Education Agreement, the student may not drop the course associated with the work placement, except for exceptional circumstances. A student who decides to do so will not be able to stay in the Co-operative Education program.

Tuition and Fees

Each work term placement is a 3-credit course and students will pay tuition based upon their fee paying status (Quebec resident, Canadian out-of-province, International).

Work Term Evaluation

Successful completion of the work term is based upon the following:

- The receipt of a satisfactory job performance report from the employer
- The submission of a satisfactory work term report by the student.

The job performance report will be completed by the employer, using guidelines supplied by the Computer Science Department. It is the student's responsibility to ensure that the employer sends the completed evaluation to the Co-op Coordinator on or before the established deadline. Employer evaluations are confidential and are not reported on the student's transcript.

Computer Science Minor (24 credits) MINCSC

9 required: CS 201, CS 211, CS 304
15 electives from any CS course

Certificate in Computer Science (30 credits) CONCSC

Description and objectives:

The Certificate Program in Computer Science is 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: Programming, Software Engineering, Web Design, Networks, Graphics, Artificial Intelligence and others. This program will help students to take full advantage of the computer technology available in the workplace.

Prerequisites to programs:

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

Program Overview

Certificate in Computer Science

12 required credits: CS 201, CS 211, CS 304, CS 321
18 credits of CS electives

General Notes/Restrictions:

1. Only one of CS 404, CS 408 or CS 499 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 must fulfill their Arts and Science requirements and Humanities requirements outlined in the Faculty section of the Calendar.

List of Courses

CS 201 Foundations of Computer Science 3-3-0

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 algorithms, hardware, system software and applications. The course includes elementary programming using Python to illustrate the implementation of algorithms. Topics include Algorithmic foundations of Computer Science; The hardware world: number systems, Boolean logic, computer circuits, Von-Newmann architecture; System software: assembly language, and a choice of applications.

Note: Registration priority is given to Science and IT students. CS students must take this course in their first year. This course will use Grade 12 level mathematics. MAT 190 is a recommended prerequisite.

CSL 201 Foundations of Computer Science Laboratory 1-0-3

This is the practical laboratory for CS 201

CS 203 Interactive Web Page Design 3-3-0

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.

CSL 203 Interactive Web Page Design Laboratory 1-0-3

This is the practical laboratory for CS 203

CS 207 Databases and Dynamic Web Design 3-3-0

In this course, students will build dynamic websites using SQL and PHP, learn the wen technology that powers 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 SQL language allow a high degree of selectivity as well as storage and retrieval of large image and media content objects. This course expands upon the techniques of web programming introduced in CS 203 to teach the use of databases and 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, setup mailing lists, build discussion forums, create a store front and even build a working shopping cart. This course emphasizes practical skills with hands-on-projects. The course is offered with a lab (CSL 207)

Prerequisite: CS 203 or permission of the instructor

CSL 207 Databases and Dynamic Web Design 1-0-3

This is the practical laboratory for CS 207

Co-requisite: CS 207

CS 211 Introduction to Programming 3-3-0

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 simple data structures. The course provides the necessary programming skills needed for further studies in Computer Science.

CSL 211 Introduction to Programming Laboratory 1-0-3

This is the practical laboratory for CS 211

CS 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, Files Servers, Workstations, etc.), Network Protocols (TCP/IP, Apple Talk, Novel, etc.) and Network Infrastructure (Hubs, Routers, Gateways, Bridges, etc.).

CS 216 System Programming Languages 3-3-0

System programmers need to understand how a computer works at a low level. They program primarily in C, with some assembly language. This course covers number systems, the C programming language, and an assembly language for a representative processor architecture. Topics covered include addressing modes, the stack, function calls and argument passing.

CSL 216 System Programming Languages Laboratory 1-0-3

Practical work for CS 216 will consist of programming in C and MIPS assembly language.

CS 219 General Topics in Computer Applications 3-3-0
The course will present general Computer Science-related topics, of interest to both Computer Science as well as non-Computer Science students. The course content is expected to vary to reflect the interest of students and Faculty, as well as market innovations.

CSL 284 Unix System Administration Laboratory 1-0-0
This lab familiarizes students with the Linux and Unix environments covering system administration and user management. Students will start with isolated machines then learn how to interface a Unix system with a network. Advanced topics include the configuration and administration of email and Web servers, as well as techniques for the automation of system administrator tasks via scripting languages. All students will have root and console access to real machines, thus they will gain real networking experience.
Prerequisite: CS 211

CS 301 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-base processing of data. Students will be expected to participate in class discussions and role-playing scenarios, and to write a term paper.

CS 304 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: CS 211

CS 306 Functional and Logic Programming 3-3-0
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.
Prerequisite: CS 304 Allow concurrent

CS 307 Using and Designing Data Bases 3-3-0
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: CS 304
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).

CSL 307 Using and Designing Data Bases Laboratory 1-0-3
This is the practical laboratory for CS 307

CS 308 Scientific Programming 3-3-0
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.
Prerequisites: CS 304, MAT 191, MAT 192
Note: Students may not take this course for credit if they have received credit for MAT 279.

CS 310 Introduction to Software Specifications 3-3-0
This course provides to all the students in CS degrees essential material on formal languages and automata, and also on program specification using logical predicates. The following topics will be addressed: introduction to techniques for specifying the behaviour of software, with applications of these techniques to design, verification, and construction of software; logic-based techniques such as loop invariants and class invariants; automata and grammar-based techniques, with applications to scanners, parsers, user-interface dialogs and embedded systems; computability issues in software specifications. These topics have been chosen because they are both theoretical and practical, and will be presented as such.
Prerequisite: CS 211
Prerequisite or Co-requisite: MAT 200

CS 311 Computer Organization and Logic Design 3-3-0
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, shirts registers, counters, sequential circuits), the ALU, memory (RAM,ROM, secondary storage), I/O Devices and the control unit (hardwired, microprogramed). For those interested students, a follow-up course, largely consisting of lab experiments, is CS 312
Prerequisites: CS 201, CS 211

CSL 311 Computer Organization and Logic Design Laboratory 1-0-3
This is the practical laboratory for CS 311

CS 312 Microcomputer Interfacing 3-3-0
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: CS 311 or equivalent programming and electronic knowledge.

CSL 312 Microcomputer Interfacing Laboratory 1-0-3
This is the practical laboratory for CS 312

CS 315 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.
Prerequisite: CS 211 or CS 216
Note: See PHY 365. Students may not take this course for credit if they have received credit for PHY 365.

CS 316 Artificial Intelligence 3-3-0
The course introduces students to the basic theoretical and practical aspects of Artificial Intelligence including Intelligent agents and environment, problem solving using classical search techniques and constrained search techniques, Knowledge representation using Logic order 0 and Logic order 1, Planning and Machine learning. The students will be exposed to solving different types of artificial intelligence problems mainly in search (e.g., shortest path, game), inference (reason about properties and relationships of individual objects), and learning (Machine learning). The course may include a team-based project.
Prerequisites: CS 304 and MAT 200

CS 317 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.
Prerequisites: CS 304 and MAT 200

CS 318 Advanced C++ Programming 3-3-0
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: CS 304

CS 321 Advanced Programming Techniques 3-3-0
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: CS 304 Allow concurrent

CSL 321 Advanced Programming Techniques Laboratory 1-0-3
This is the practical laboratory for CS 321

CS 325 Computer & Network Security 3-3-0
This course provides an introduction to security and privacy issues in various aspects of computing, including cryptography, software, operating systems, networks, databases, and Internet applications. It examines causes of security and privacy breaches, and gives methods to help prevent them.
Prerequisite: CS 214

CS 330 Programming Mobile Apps 3-3-0
This course will cover mobile application development for the Android operating system using Android Studio. The programming language is Java. Setting up and using Android Studio IDE will be covered in the introduction. Students will learn how to design and develop Android applications using best practices to account for the limited screen size and memory of mobile devices. Topics to be covered include layout design/management, communication between apps, Google Maps, 2D graphics, and mobile app specific software engineering patterns. The course concludes with monetization (ads, in-app purchases, etc) and app store optimization strategies. The goal is for each student to develop and release an app by the end of the course.
Prerequisite: CS 211

CSL 330 Advanced Programming for Mobile Apps Laboratory 1-0-3
This is a practical laboratory for CS 330
Co-requisite: CS 330

CS 375 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.
Prerequisites: CS 211, MAT 108 and MAT 207.
Note: See MAT 325 and PHY 375. Students may not take this course for credit if they have received credit for MAT 325 or for PHY 375.

CS 379 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 319. Students may not take this course for credit if they have received credit for PHY 319.

CS 391 Co-operative Placement I 3-0-0
Students will integrate theory and practice through a related work placement
Prerequisite: admission to the Co-op Education Program

CS 392 Co-operative Placement II 3-0-0
Students will integrate theory and practice through related work placement
Prerequisite: CS 391

CS 393 Co-operative Placement III 3-0-0
Students will integrate theory and practice through related work placement
Prerequisite: CS 392

CS 394 Stage in Bioinformatics 6-0-0
Students will integrate theory and practice through a related stage.
Prerequisite: CS 372 / BCH 342

CS 400 Independent Studies 3-0-0
Individual study and research under the guidance of an advisor and Department staff.
Prerequisite: Permission of the department

CS 401 Simulation Techniques 3-3-0
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.
Prerequisites: CS 304, 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.

CS 402 Computer Graphics 3-3-0
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: CS 304, MAT 108

CSL 402 Computer Graphics Laboratory 1-0-3
This is the practical laboratory for CS 402

CS 403 Principles of Programming Languages 3-3-0
The objective of this course is to introduce, analyze 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: CS 304 and CS 310

CS 404 Project 3-0-3
This course is normally taken by CS 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.
Prerequisites: approval of the dept., 80% in CS courses

CS 405 Data Mining 3-3-0
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, 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 models combination. This course includes assignments and a final project where the students are required to perform mining on real datasets.
Prerequisite: PHY 101 (or equivalent)
Note: See PHY 374. Students may not take this course for credit if they have received credit for P HY 374.

CS 406 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: CS 310
This course will be offered on alternate years.

CS 408 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.
Prerequisites: approval of the dept., 80% in CS courses.

CS 409 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.
Prerequisite: CS 304

CS 410 Software Engineering 3-3-0
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. Students will be expected to work jointly on several large software projects.
Prerequisites: CS 304, CS 310, CS 321, CS 403 (allow concurrent)

CS 411 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, micro-programed 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.
Prerequisite: CS 311 or instructor's permission.

CS 412 Computer Games Design 3-3-0
This course will explore the theory and practice of video game design and programming. Students will learn the basic concepts and techniques for the design and development of digital games. The topics covered in this course will include the history and taxonomy of video games, the basic building blocks of a game, computer graphics and programming, use interface and interaction design, and the software architecture for video games. It is assumed that students have taken courses in programming (best if it includes C or C++) and data structures. A good background in algorithms and basic mathematics (matrix algebra, trigonometry, linear algebra, vector calculus) is an asset for this course.
Prerequisite: CS 304

CS 415 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.
Prerequisite: CS 304

CS 416 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.
Prerequisite: CS 304

CS 417 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.
Prerequisite: CS 304

CS 418 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 reflecting the interests of the students and the faculty. It will be offered by arrangement with the department.
Prerequisite: CS 304

CS 419 Android System Programming 3-3-0
This course will focus on system programming in the context of the Android Open Source Project (AOSP). We will learn how to port Android from standard platforms currently supported in AOSP to new platforms, focusing on the system layers that need to be changed in the porting process. The course begins (first 2-3 weeks) with an introduction to the Unix-type operating system to get up to speed with OS concepts needed for the course (processes, page tables, trap handling, etc.). The remainder of the course focuses on Android system programming, including boot-up, build layers, WIFI, graphics, recovery, and OTA updates. The final project involves adding new OS features to the AOSP and installing your new version of Android on a physical mobile device. There will be an emphasis on group work and collaboration with your classmates to meet the objectives ofthe course.
Prerequisites: CS 216, CS 304

CS 436 Web Mining 3-3-0
Web mining, an interdisciplinary field, is at the forefront of analyzing and harnessing the vast and diverse data available on the World Wide Web. This course offers students a comprehensive introduction to the principles, techniques, and practical applications of web mining, equipping them with the skills to extract valuable insights from web data. Topics covered in this course include, but are not limited to, association rules, machine learning, information retrieval, social network analysis, and web crawling. Additionally, students will have the opportunity to engage in substantial projects, enabling them to gain hands-on experience with web mining tools. By the end of the course, students will possess the knowledge and capabilities to effectively navigate the complex web data landscape and apply their skills in various real-world scenarios.
Prerequisite: CS 317 or permission of instructor
Note: Cross-listed with CS 536. Students who take this course for credit may not receive credit for CS 536.

CS 446 Intelligent Systems and Neural Networks 3-3-0
The course presents students with theoretical and practical knowledge on different models of intelligent systems including bio-inspired systems (Genetic algorithms, Swarm optimization), Artificial neural networks, Recommendation systems and Bayesian networks. Students will be exposed to the implementation of different types of intelligent systems such as: neural networks for supervised learning, genetic algorithms and swarm optimization algorithms for combinatorial problems (e.g. the knapsack problem, the traveling salesman problem), recommendation systems for recommendation problems (e.g., book recommendation, movie recommendation), and Bayesian networks for probabilistic reasoning for different applications (e.g. disease prediction, document classification). The course may include a team-based project.
Prerequisites: MAT 191, PHY 101, CS 304

CS 450 Elements of Big Data 3-3-0
In this course, students will learn the fundamental theory and techniques of Big Data management and analytics, and to apply them to resolve problems in real-world applications. The principle is to learn strategic extraction and usage of relevant information in large datasets. In fact, the students will exploit recent concepts and trends to manage and analyze Big Data. For that purpose, recently designed algorithmic approaches and technologies will be covered to help the students to manage and analyze large datasets. For that aim, the covered topics will include, but will not be limited to streaming algorithms, distributed file-system architecture, resilient distributed datasets, similarity search, recommendation systems, and distributed machine learning. The students will work on large projects to practice the concepts presented in this course
Prerequisites: PHY 101 and CS 321
Cross-listed: CS 550

CS 454 Complements in Data Structures and Algorithms 3-3-0
The aim of this course is to cover many concepts in Data Structures, Algorithms, and Programming to make up deficiencies in Computer Science background for entering graduate students.
This course cannot be taken for credits by undergraduate students.

CS 455 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.
Prerequisites: CS 211, MAT 200

CS 457 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.
Prerequisite: CS 307

CS 462 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.
Prerequisites: CS 304, MAT 192, PHY 101 (or equivalent)

CS 463 Computer Vision 3-3-0
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
Prerequisites: CS 304, CS 318, MAT 192, PHY 101 (or equivalent)

CS 464 Network Programming 3-3-0
This 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). 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.
Prerequisite: CS 216

CS 467 Special Topics in Algorithms 3-3-0
The course builds on the techniques covered in CS 317 to present some specialized algorithms in several areas, including Bioinformatics, Computational Geometry, and Network Flow.
Prerequisite: CS 317 or permission of the instructor

CS 469 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.
Prerequisite: CS 304

CS 471 Graph Theory 3-3-0
An introduction to the combinatorial, algorithmic and algebraic aspects of graph theory.
Prerequisites: CS 304, MAT 200
Note: See MAT 421. Students may not take this course for credit if they have received credit for MAT 421.

CS 499F 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

Mathematics

Faculty

Madjid Allili,
B.Sc.(Algiers), M.Sc., Ph.D.(Sherbrooke);
Professor

Thomas Brüstle,
B.Sc., (Ludwig-Maximilians), M.Sc.,
Ph.D. (Zurich); Professor,
Maurice-Auslander Research Chair

François Huard,
B.Sc., M.Sc., Ph.D. (Sherbrooke); Professor

Trevor H. Jones,
B.Sc.H. (Acadia), M.Sc. (Dalhousie),
Ph.D. (University of New Brunswick);
Senior Instructor

Scosha Merovitz,
B.Sc.(Bishop's), M.Sc.(Dalhousie);
Special Instructor
Chair of the Department

David Smith,
B.Sc., M.Sc., Ph.D. (Sherbrooke);
Adjunct Professor

N. Brad Willms,
B.Math. M.M., Ph.D. (Waterloo);
Associate Professor

Program Overview

Mathematics is the language of the sciences, a language which allows scientists to quantify, model, understand and predict behaviour in an enormously diverse range of phenomena of interest. Simultaneously, Mathematics is often regarded as an art, as it is the creative study of patterns and of problem solving. Mathematics covers a wide range of disciplines including algebra, analysis, combinatorics and discrete mathematics, and differential equations. In first-year courses, mathematics students are joined by other science students, particularly from Physics and Computer Science. In the advanced courses, classes are very small, and some are given on an individual or tutorial basis.

The highest level of specialization is Honours, and Honours programs prepare students for direct entry into graduate work leading to a Master's or Ph.D. degree. All honours mathematics students have an opportunity to study independently and thus develop their reading and problem solving skills, and there is some chance to pursue special interests. The Majors programs provide students with an excellent general preparation for the career world, while not preventing entrance into graduate school (sometimes after a qualifying year). The Majors programs have sufficient electives to allow students to combine their major with a second major or at least a minor (the least specialized type of program) in another discipline. Students are encouraged to add a minor or major and many do so. Popular choices include computer science, physics, music, English, French, Spanish, drama, and philosophy. The Department of Mathematics offers several specialized, interdisciplinary programs, jointly with other departments, including Hispanic Studies and the School of Education.