Computer Science

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subject related to almost every contemporary intellectual discipline, the arts (computer-generated art, verification of historical documents, syntax study of languages including translation, etc.), social sciences (correlating experimental data, simulation, artificial intelligence studies), natural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) Master's Degree Program

2) Undergraduate B.Sc. Degree Programs:

   B.Sc, Honours in Computer Science

   B.Sc, Major in Computer Science

3) A multidisciplinary Bachelor of Arts with a Major in Information Technology (BAIT) (details to be found in the Multidisciplinary Programs section at the end of the calendar)

4) Minor in Computer Science

5) Certificate Programs

   Certificate in Computer Science

   Certificate in Software Technology

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

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 five one-term 500-level courses, registration and participation in the Graduate Seminar (CSC 597), 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 approved program are designated as secondary. Failure to attain a minimum of 65% in any of the primary courses may result in the student being required to withdraw from the program.

Under certain circumstances, it is permissible for a student admitted to the program to follow an approved graduate-level credit course at another university. All interested students should consult their supervisor and the chair of their department prior to registration in order to obtain further information on procedures and conditions of eligibility.

A thesis proposal should be completed as soon as possible and by the end of the second term in the program at the latest. Students are expected to present their proposal in the Graduate Seminar course and also expected to give more detailed seminars describing their work later.

Thesis topics are chosen after discussion with potential supervisors. The amount of flexibility allowed in pursuing a particular topic will vary according to the supervisor's needs and interests. Theses are defended before an examining committee consisting of two members of the department, and an external examiner.

Any candidate (full-time or part-time), after initial registration in a thesis must maintain this registration in all successive terms (including the term in which the student is examined) until his/her thesis is completed. Completion means submission of a final grade to the Division after modifications, any retyping involved, etc. Students should note that faculty approval to register in the thesis is given on the understanding that the student will be in regular contact with his/her supervisor, and that thesis research will be actively pursued in each term of registration.

Preparatory Students

Candidates who do not satisfy the admission requirements may be admitted to a qualifying program. If successful in this qualifying period and upon formal application to the Division, the student may eventually proceed to the Master's program. However, admission to the qualifying program does not imply automatic admission to the Master's program; at the end of the qualifying period the student will be required to apply for entry into the Master's program, at which time the department will determine the student's eligibility. If successful, the student will be informed of this decision by the Dean of the Division.

Credits taken to fulfill the requirements of the qualifying program may not be used for credit for the Master's degree. Courses taken extra to the program requirements of the qualifying year and which have been successfully completed may be considered for credit towards the Master's degree.

Graduate Courses

Master's students may take any four graduate courses as long as these are approved by their supervisor. Graduate students in the department may include in their programs relevant courses from other departments within the Division.

The department currently offers the following courses. The actual courses offered each term will be determined by student demand and the availability of faculty.

Computer Science 502ab Digital Topology & Mathematical Morphology 3-3-0
Digital topology deals with topological properties of digital images. Its concepts and results are used to specify and justify some important image processing algorithms, including algorithms for thinning, boundary extraction, object counting, and contour-filling. Mathematical Morphology is a set-theoretic method of image analysis allowing to extract image components that are useful for representation and description. It can provide boundaries of objects, their skeletons, and their convex hulls. It is also useful for many pre-and post- processing techniques, especially in edge thinning. The goal of this course is to provide a full exposure to these techniques, their mathematical backgrounds, and their different applications.

Computer Science 505ab Automata Theory & Computational Complexity 3-3-0
Cross-listed with CSC405. The course will address finite-state machines, context-free languages and pushdown automata, computability. A systematic study of the known relations between the most important resource bounded complexity classes, reductions, separation results and translation techniques is also included. Students are expected to prepare a research paper during the course.

Computer Science 506ab Parallel Models and Algorithms 3-3-0
This course provides an introduction to the design and analysis of parallel algorithms and to the various models of parallel computation. The course will discuss parallel algorithms for problems such as: basic arithmetic, sorting, searching, selection, graph theory, matrix computations, combinatorial enumeration, optimization, computational geometry, and numerical analysis. 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 510ab 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 512ab 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 513ab Image Analysis 3-3-0
Cross-listed with CSC413. Image analysis is concerned with the development of machine algorithms in order mimic the biological organism's ability to see and understand images and videos. The course content include: 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 514ab 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 515ab 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 livestock. This course provides an introduction to an algebraic language (such as CSP) for specification, analysis, and verification of conccurrent and real-time systems. It then goes and presents the use of such a language as a formal method for concurrency at different stages in the development process.

Computer Science 516ab 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 517ab 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 transcactions. 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 519ab 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 527ab Advanced Topics in Algorithms 3-3-0
Cross-listed with CSC 427. The course covers some advanced aspects of algorithms and complexity. It studies the topic of NP- complete problems. Some specialized algorithms in several areas will be discussed, such as Bioinformatics, Computational Geometry and Network Flow.

Computer Science 596ab 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 599ab Master's Thesis 24-0-0

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 equiva-

lent) 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:

I) Honours in Computer Science

Total of 95 credits (125 credits for a 4-year programme) plus the EWP:

A. Dissertation-based Honours

at least 59 CSC credits:

44 required:	CSC 101, CSC 111, CSC 116, CSC 121, CSC 204, CSC 211, CSC 217, CSC 275/CSC208, 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. CSC499) 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:

At least 62 CSC credits:

38 required	CSC101, CSC111, CSC116, CSC121, CSC204, CSC211, CSC217, CSC275/CSC208, CSC303, CSC309, CSC405
24 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

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 requires, in addition, departmental permission.
- students must maintain an average of 80% in required courses to stay in the program

II) Major in Computer Science

Total of 95 credits (125 credits for a 4-year programme) plus the 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

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.

CO-OPERATIVE EDUCATION PROGRAMME

The Co-operative Education Programme combines a student's academic programme 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 (CSC 391, CSC 392 or CSC 393). These course credits are in addition to regular degree programme requirements. **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.

Students in a 90-credit programme will be required to complete 2 work terms (6 credits). Students in a 120-credit programme will be required to complete 3 work terms (9 credits). These credits will be added to the student's programme 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 programme, no guarantee of placement can be made since the employment process is competitive and subject to market conditions.

Admission to the Co-operative Education Programme

Full-time students in the Computer Science Department with a minimum cumulative average of 70% after completing 15 credits (1 semester) are admissible into the Co-op Programme provided they submit their résumé (curriculum vitae) and a letter of intent for review. Admission is limited to full-time students in any Honours or Major programme offered in the Computer Science Department. Students in the Co-op must maintain their 70% average in order to stay in the program.

Please consult the Co-operative Education section on the Bishop's University website for important dates, deadlines and updates at www.ubishops.ca/coop

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). In addition, an administrative fee *of $200 per placement* will be charged to help cover the cost of monitoring and evaluating work placement. Payment is to be made through the Business Office.

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.

MINOR IN COMPUTER SCIENCE

26 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, CSC308 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.

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 Computer Science

Required courses : 19 credits

CSC 101	**Foundations of Computer Science**	**4-3-3**
CSC 111	**Programming Methodology**	**4-3-3**
CSC 116	**Low Level Programming Language**	**4-3-3**
CSC 204	**Data Structures**	**3-3-3**
CSC 211	**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 101	**Foundations of Computer Science**	**4-3-3**
CSC 111	**Programming Methodology**	**4-3-3**
CSC 204	**Data Structures**	**3-3-0**
CSC 303	**Principles of Programming Languages**	**3-3-3**
and at least one of:		
CSC 218	**C++ Programming**	**3-3-3**
CSC 328	**Object Oriented Software Construction**	**3-3-1**
CSC 316	**Special Topics in Software**	**3-3-0**
CSC 319	**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-Newmann 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 103ab 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 116ab 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 121ab Advanced Programming Techniques 4-3-3
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 201ab 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.

Computer Science 204ab 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 206ab 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.
Corequisite: Computer Science 204b

Computer Science 207ab 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 208ab 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.
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, shirts 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 214ab 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.).

Computer Science 215ab 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 216ab Artificial Intelligence 3-3-0
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 217ab 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 218ab 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: Computer Science 204

Computer Science 219ab 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.

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 301ab 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.
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 302ab Computer Graphics 4-3-3
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 303ab 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 304ab 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 306ab 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 308ab 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 310ab 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; 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 311ab 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 superscalalr 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 318ab 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.

Computer Science 320ab Concurrent Programming 3-3-0
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 328ab 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 372ab Bioinformatics 3-3-0
This course will introduce students to Bioinformatics, which uses computer databases to store, retrieve and assist in understanding biological information.
Prerequisite: BCH 370, CSC 205
Note: See BCH 342. Students may not take this course for credit if they received credit for BCH 342

CSC 391ab 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

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

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

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

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 ab 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 412ab 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 413ab 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. 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 3-3-0
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 427ab 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.
Prerequisite: CSC217 or permission of the instructor

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 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

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.

Entrance Requirements

To enter a Mathematics program, a student will normally have completed either a Québec Collegial Diploma (DEC), or grade 12 in another Canadian province or the U.S.A., or the equivalent level of education internationally. Admitted students with a DEC will normally be registered in a three-year program of at least 94 credits, while students with a grade 12 diploma (or equivalent) will be registered in a four-year program of at least 124 course credits. A student's total credit requirement depends on the program and the type of degree chosen. However, all Bishop's students must complete the 3-credit, English Writing Proficiency (EWP) requirement which is included in the total credit requirement.

Students entering a Mathematics program at Bishop's from another Canadian University or College, or from accredited international post-secondary institutions, will have their transcripts of grades examined individually for possible transfer credit against a Bishop's program's requirements.

The requirements for Mathematics students differ depending on whether the degree being pursued is a B.Sc. or a B.A., and whether the student entered Bishop's after completing CEGEP in Québec, or after completing grade 12 (or the equivalent) in another province or elsewhere. Two or more Mathematics programs may not be combined in any Bishop's degree.

First-year Calculus requirement

All Mathematics students require six course credits of Calculus studies, normally in first year. Students with any Québec collegial diploma (DEC) are exempt from this requirement if they have completed a course in Differential Calculus and a course in Integral Calculus at CEGEP. If one or both of these courses were not completed at CEGEP, they must be completed at Bishop's and the credits will be added to the total credit requirement. Students entering four-year programs in Mathematics with a grade 12 diploma (or equivalent) must register in Mathematics 191 and 192 in their first year. These courses are included in the 124 (or higher) total credit requirement. Students transferring into Mathematics programs may use credit for Mat 198 to replace Mat 191, and Mat 199 to replace Mat 192. Credit for Mat 197 with a grade of 80% or higher will also be accepted to replace Mat 191. Mathematical Contexts Minor program students normally complete Mat 198 and 199 (instead of Mat 191 and 192, although theses are acceptable), and do not need to do so in their first year.

First-year Physics requirement

Mathematics students pursuing the Bachelor of Science (B.Sc.) degree require six course credits of introductory physics studies in first year. Students in the Bachelor of Arts (B.A.) degree program are exempt from this requirement. Students with any DEC are exempt from this requirement if they have completed two introductory Physics courses, Mechanics, and Electricity and Magnetism, at CEGEP. If one or both of these courses were not completed, they must be completed at Bishop's and the credits will be added to the total credit requirement. Students entering four-year B.Sc. programs in Mathematics with a grade 12 diploma (or equivalent) must register in Physics 191 and 192 in their first year. These courses are included in the 124 (or higher) total credit requirement.

The Programs

The Department of Mathematics offers the following degree programs:

Name of Program	Code	Levels of Specialization Available	Degree Types Available
Mathematics	MAT	Honours, Major or Minor	B.Sc. or B.A.
Mathematics Education	EDM	double Major with Education	B.Sc. or B.A.
Matemáticas en Español	n/a	Major in Mathematics combined with a Minor in Hispanic Studies	B.Sc. or B.A.
Mathematical Contexts	MAC	Minor	add to any