UNIVERSITY OF TORONTO
FACULTY OF ARTS & SCIENCE

Appendix B

Major Program Modification Proposals for the 2011-2012 Calendar

Ecology and Evolutionary Biology, BSc, Specialist
Department of Ecology & Evolutionary Biology (EEB)

The Department currently offers a major program in Ecology and Evolutionary Biology and considers that student demand, as well as the availability of interdisciplinary courses already offered by the Department and cognate units, will result in a successful specialist program. The University of Toronto is at the forefront in discovery, innovation, and teaching of biodiversity and conservation issues. Ecology and Evolutionary Biology, which conducts research and teaching in biodiversity and conservation, is at the core of this effort.

The Department was established in 2006 and has been reviewing its offerings in line with its new mission. This new program proposal has resulted from this planning process. EEB currently sponsors three Specialist programs, with a total of about 100 students in all years: Ecology, Evolutionary Biology, and Behaviour. These three programs have existed for many years (and prior to 2006-07 were administered by Department of Zoology). The differences among these programs were small in terms of required courses for each. The faculty in EEB had a day-long retreat in Oct. 2009 to discuss undergraduate curriculum issues, where it was agreed to reduce the number of EEB Specialist programs from three to one (Ecology and Evolutionary Biology) with three defined streams (areas of concentration in Ecology, Evolutionary Biology, Behaviour).

Description and Calendar Entry

Ecology and Evolutionary Biology is a broad discipline that seeks to understand the origins, diversity, and distribution of organisms. This Specialist program provides an in-depth understanding of ecological and evolutionary patterns and processes, as well as the diversity of life forms (microbes, fungi, plants, animals). Concepts are taught using a broad array of approaches, including molecular studies, laboratory experiments, computer and mathematical modelling, and field studies. An integral part of the experience is to conduct independent research projects in the laboratory and/or field. There is a strong emphasis within the program on hands-on laboratory and fieldwork that complement the conceptual framework developed in lectures. Students in this program have the opportunity to concentrate in ecology, evolutionary biology, or behaviour.

Ecology and Evolutionary Biology Specialist program:
(12 full courses or their equivalent, including at least 4.0 FCEs at the 300+-level, 1.0 of which must be at the 400-level)

First Year (3.0 FCEs): BIO120H1; BIO130H1; CHM138H1; CHM139H1; MAT135Y1/137H1/157H1

1. 2.0 FCEs: BIO220H1 (ecology and evolutionary biology); BIO230H1 (molecular and cell biology); BIO260H1/HMB265H1 (genetics); BIO251H1/BIO270H1 (plant/animal form and function)

2. 0.5 FCE in statistics from: EEB225H1 (recommended); PSY 201H1; STA 220H1/250H1/257H1; GGR270H1

3. 0.5 FCE in core evolution from: EEB 318H1, 323H1, 362H1

4. 0.5 FCE in core ecology from: EEB 319H1, 321H1, 328H1, 370H1

5. 0.5 FCE in organismal biology from: EEB 263Y1, 266H1, 267H1, 268H1, 330H1, 331H1, 337H1, 340H1, 356H1, 360H1, 382H1, 384H1, 386H1, 388H1, 389H1

6. 0.5 FCE in environmental biology: ENV234H1

7. 1.0 FCE at 300+ series, from: EEB 318H1, 319H1, 321H1, 322H1, 323H1, 324H1, 328H1, 330H1, 331H1, 337H1, 340H1, 356H1, 360H1, 362H1, 365H1, 370H1, 375H1, 382H1, 384H1, 386H1, 388H1, 389H1, 428H1, 440H1, 459H1, 460H1, 465H1, 466H1, 494H1, 495H1, 496H1; EHJ 351H1, 352H1; ENV334H1

Sub-total = 8.5 FCEs

8. 1.0 to 2.5 FCEs in at least two of the three following categories: (1) one field course (0.5 FCE) from EEB 401H1, 403H1, 404H1, 405H1, 406H1, 407H1, 409H1, 410H1/ FOR306H1; (2) one seminar (0.5 FCE) from EEB 494H1, 495H1, 496H1; and/or (3) one independent research project course (1.0 FCE) from EEB 498Y1/499Y1 and concurrent research issues course EEB488H1 (0.5 FCE).

Sub-total = 9.5 or 11 FCEs (depending on options chosen in #8)

9. Select the remaining FCEs for a total of 12.0 FCEs (at least 1.0 must be 300+ series if 1.0 FCE is completed in #8 above) from: BIO271H1; all EEB courses (excluding EEB202H1/214H1/215H1/216H1); EHJ 351H1, 352H1; ENV334H1; JHE 353H1, 355H1; and no more than 1.0 FCE from the following (note that some courses may require prerequisites that are not listed within this program): ANT 336H1, 333Y1, 335Y1, 430H1, 436H1; CSB 328H1, 340H1, 349H1, 350H1, 352H1, 353H1, 430H1, 431H1, 452H1, 458H1, 472H1, 474H1; ENV 315H1, 346H1; FOR 200H1, 307H1, 413H1, 416H1, 417H1; GGR 201H1, 203H1, 205H1, 206H1, 305H1, 307H1, 308H1, 403H1, 409H1; GLG 202H1, 351H1, 436H1; JGE 347H1, 348H1; PSY100H1, 260H1, 270H1, 280H1, 290H1, 390H1, 397H1, 492H1, 497H1 (note that many PSY courses have limited enrolment)

Total = 12 FCEs

NOTE: Students may wish to concentrate in ecology, evolutionary biology, or behaviour. Recommended EEB, EHJ and JHE courses for these concentrations are as follows:

Ecology: EEB 255H1, 319H1, 321H1, 328H1, 365H1, 370H1, 375H1, 428H1, 440H1, 465H1, 495H1; EHJ351H1
Evolutionary Biology: EEB 323H1, 324H1, 362H1, 440H1, 459H1, 460H1, 466H1, 494H1; EHJ352H1; JHE 353H1, 355H1
Behaviour: EEB 322H1, 496H1

Academic Context

Ecologists and evolutionary biologists recognize that all life has evolved and that an understanding of the factors influencing the origin and maintenance of biological diversity − from genomes to ecosystems − underlies all life sciences and is critical to our stewardship of life on this planet. Society needs to make informed decisions about sustainable development, global temperature change, control of invasive species, the preservation of genetic diversity and ecosystem integrity, and the control of emerging infectious diseases. These are fundamentally evolutionary and ecological problems.

Students in this program will obtain an in-depth understanding of the diversity of life forms (microbes, fungi, plants, animals) and the diverse aspects of organismal biology in the natural world. Their studies will include a broad array of approaches – including molecular studies, laboratory experiments, computer and mathematical modeling, and field studies – as well as opportunities to conduct independent research projects in the laboratory and/or field. There is a strong emphasis within the program on hands-on laboratory and fieldwork that complement the conceptual framework developed in lectures. Graduates of this program will come to realize that the ecological and evolutionary underpinnings of life present a host of scientific problems that are both intellectually challenging and critical to our future.

Graduates of this program will be well positioned to pursue graduate studies in ecology and evolutionary biology, as well as careers in universities, colleges, primary and secondary schools, environmental consulting firms, environmental law, science journalism, national or provincial parks, hospitals, government and non-governmental agencies, resource management agencies, private industry, research labs, and public utilities.

Learning Outcomes

At the completion of this program students will:

- Recognize that all life has evolved and that an understanding of the central question of the origin and maintenance of diversity − from genomes to ecosystems − underlies all life sciences and is critical to our stewardship of life on this planet.
- Obtain an in-depth understanding of the diversity of life forms (microbes, fungi, plants, animals) and the diverse aspects of organismal biology in the natural world.
- Have been exposed to a broad array of scientific approaches – including molecular studies, laboratory experiments, computer and mathematical modeling, and field studies in different areas of the world – as well as have opportunities to conduct independent research projects in the laboratory and/or the field.
- Critically evaluate scientific information, use information to generate hypotheses, assess whether evidence supports their conclusions and the conclusions of others, and use this knowledge to solve problems.
- Possess a comprehensive understanding of evolutionary and ecological principles so that they can make informed decisions on pressing societal issues that include population growth, emerging diseases, global environmental change, and the conservation of biodiversity.
- Understand that evolution is the unifying, central concept in biology, and that organismal traits – ranging from genomes to social behaviour to aging – are the product of a complex interplay between the environment, contemporary selective forces, evolutionary history, and genetic and developmental constraints.
- Realize that adaptive evolution is a process that results from selection pressures imposed by the physical and biotic environment on individuals within populations; and that the ecological challenges of capturing resources for growth, successful reproduction, and avoiding enemies largely determine the ways organism’s function.
- Appreciate that determining the mechanisms of change in biological systems requires theoretical and empirical approaches and the application of scientific hypothesis testing across a range of phenomena in the biological hierarchy; and that the identification of recurrent patterns of variation across different spatial and temporal scales is more likely to lead to general conclusions than the ad hoc collection of facts from individual case histories.
- Understand that ecological and evolutionary change in biological systems requires a multidisciplinary approach and the integration of information from different levels in the hierarchy of life, from genes through organisms to ecosystems; and that a focus on any one level will often fail to provide comprehensive insight.

Degree Objectives

a. DEPTH OF KNOWLEDGE

This Specialist program provides students with an in-depth understanding of ecological and evolutionary principles, including laboratory and field-based coursework. Depth of knowledge within ecology and evolutionary biology is achieved through course prerequisites. The foundation courses in ecology and evolutionary biology – BIO120H1 and BIO220H1 – are prerequisites for all 300+ level courses in EEB. Each student in this program is required to complete a core course (0.5 FCE) at the 300-level in ecology (from EEB 318H1, 319H1, 321H1, 328H1, 370H1) and evolution (from EEB 318H1, 323H1, 362H1). Students are also required to complete an additional 2.0 to 3.5 FCEs at the 300+ level in ecology and evolutionary biology. At the 400-level each student completes two (or all three) of a field course, seminar, or independent research project in ecology or evolutionary biology. Field courses require each student to complete an individual project – develop hypothesis, design experiment, collect, analyse and interpret data – which is written up as a scientific article and presented orally. In seminar courses, students read and critique primary scientific literature, write grant proposals, write review papers, and actively participate in class discussions each week. In independent research project courses, students read the scientific literature, develop a research question, write a proposal including experimental design, conduct independent research, analyse and interpret data, present findings at regular lab meetings, receive feedback on their progress, write a paper in the format of a publication in a scientific journal, and communicate via an oral presentation or scientific poster. Through the sequence of these courses, students will demonstrate a command of increasingly advanced material, use the scholarly materials and research tools relevant to ecology and evolutionary biology, and learn to synthesize information from a broad range of prior learning and previous research.

b. COMPETENCIES

i. Critical and Creative Thinking

Students critically evaluate scientific information, generate hypotheses, assess whether evidence supports their conclusions and the conclusions of others, and use this knowledge to solve problems. This begins in the first year in BIO120H1 and BIO130H1 where students read the primary scientific literature, reason about data, and pose hypotheses and ways of testing these hypotheses. In second year students take BIO220H1, BIO230H1, statistics (e.g., EEB225H1/STA220H1), and genetics (BIO260H1/HMB265H1), which require reading and analysis of the primary literature and generating hypotheses (BIO220H1, BIO230H1) and creative problem solving (statistics and genetics). BIO120H1, BIO130H1, BIO220H1, and BIO230H1 each have hands-on laboratories; the required courses in statistics and genetics each have tutorials. Students in all 300+ level required courses participate in critical and creative thinking activities ranging from study and analysis of the primary literature to writing essays and lab reports where hypotheses generated and conclusions from data are developed, to oral presentations and discussions regarding conclusions drawn from the primary literature. The 300-level required core ecology and evolution courses have either weekly hands-on labs or interactive tutorials. Developed critical and creative thinking skills are required for success in the capstone courses at the 400-level (field course, seminar, research project).

ii. Communication

Students in this program will obtain proficiency at expressing their ideas effectively in writing, as well as be required to present their work orally in a field course, seminar course, or independent research course (and concurrent research issues course). Meaningful introductory writing instruction is provided in the laboratories in BIO120H1 and BIO220H1, as well as BIO130H1 and BIO230H1. ENV234H1, EEB263Y1, and 300+ series ecology and evolutionary biology courses require students to write essays, critiques of scientific papers, or formal lab reports from activities or experiments conducted in practicals. Results of independent research project courses and field courses are usually written up in the format of a publication in a scientific journal (and indeed some projects have been published in reputable journals), and are always communicated via an oral presentation or scientific poster.

iii. Information Literacy

Beginning in first year (BIO120H1, BIO130H1), students are introduced to effective strategies for finding, evaluating, and using scholarly information in the biological sciences, and this is expanded upon in BIO220H1, BIO230H1, and upper-year courses. Through the requirement of writing essays, reports, and/or critiques, students in this program become familiar with the primary databases used in ecology and evolutionary biology; learn to evaluate resources for relevance, authority, and reliability, including information available on the web; synthesize information from various sources; understand the importance of expressing thoughts in their own words and citing sources appropriately; and learn to analyze information to test a particular hypothesis. The seminar courses, project courses, and field courses all provide important opportunities for students to use the scientific literature effectively.

iv. Quantitative Reasoning

Students in this program will be proficient at reasoning with mathematical and statistical concepts in order to present, interpret and discuss results of scientific experiments and enhance their understanding of ecological and evolutionary principles. Required courses in this program include calculus in first year (e.g., MAT135Y1) and statistics (e.g., EEB225H1 – Biostatistics for Biologists). The required BIO120H1/220H1 and upper-year EEB courses with practicals have laboratory activities in which students develop their quantitative skills by analyzing and interpreting data, including statistical inference. The required BIO260H1/HMB265H1 (genetics) provides training in the manipulation and interpretation of numeric data, including statistical analysis. Advanced courses introduce the theoretical underpinnings of ecology and evolution (e.g., EEB323H1, EEB370H1, EEB459H1). For example, EEB323H1 incorporates weekly problem sets covering various aspects of population genetic theory.

v. Social and Ethical Responsibility

Students in this program will develop values of personal and academic integrity, and engage in critical reflection upon questions of responsibility to oneself and society. Lectures, essays, and lab exercises in 100- to 400-level courses in this program address many pressing societal issues, including global environmental change (e.g., BIO120H1, BIO220H1, ENV234H1, ENV334H1, EEB428H1), ethical implications of molecular genetics and its use in human populations (BIO130H1, BIO230H1), ethical concerns surrounding the collection, analysis and interpretation of human population genetic data (EHJ352H1), the conservation of biodiversity (e.g., EEB 255H1, 362H1, 365H1, 465H1, 466H1, and the many organismal biology courses), population growth (e.g., EEB319H1, EHJ351H1), emerging diseases (e.g., BIO220H1), eugenics and controversies over evaluating human intelligence (e.g., EEB322H1), and the sustainable management of resources (e.g., EB319H1, 321H1, 365H1, ENV334H1).

c. AN INTEGRATIVE, INQUIRY-BASED ACTIVITY

All required courses in this program have either labs or tutorials which are inquiry-based and integrate with lecture material.

All students in this program will engage in an advanced integrative, inquiry-based activity by completing any two (or all three) of: (1) field course (from EEB 401H1, 403H1, 404H1, 405H1, 406H1, 407H1, 409H1, 410H1/ FOR306H1); (2) seminar (EEB 494H1, 495H1, 496H1); and (3) independent research project (EEB 497H1/498Y1/499Y1, and concurrent research issues seminar course, EEB488H1). Field courses require each student to complete an individual project – develop hypothesis, design experiment, collect, analyse and interpret data – which is written up as a scientific article and presented orally. In seminar courses, students read and critique primary scientific literature, write grant proposals, write review papers, and actively participate in class discussions each week. In independent research project courses, students read the scientific literature, develop a research question, write a proposal including experimental design, conduct independent research, analyse and interpret data, present findings at regular lab meetings, receive feedback on their progress, write a paper in the format of a publication in a scientific journal, and communicate via an oral presentation or scientific poster.

Departmental/College Resource Implications	
Estimated Enrolment per Academic Year in this program (please explain)	On average there are about 100 students (in total, all years) in the current three Specialist programs (Ecology, Evolutionary Biology, and Behaviour). It is expected that with the deletion of these three programs, the total enrolment in the Ecology and Evolutionary Biology Specialist will remain at about 100 students.
New courses necessary to mount for this program	None
Additional Instructor(s) Requirements	None
Teaching Assistant(s) Requirements	Use existing TA resources.
Laboratory Equipment Requirements	Use existing laboratory equipment.
Computing Resources Requirements	Use existing computing resources; but we note that a limitation to the course capacity in several courses that require computers during practicals/labs (e.g., EEB 225H1, 321H1, 365H1, 466H1) is the available of spaces in computer labs (such as CQUEST computer labs in Ramsay Wright building). Additional space in computing labs is highly desired.
The Department of Ecology and Evolutionary Biology will provide the resources for this program.	

Computer Science, BSc, Specialist, Major and Minor

Department of Computer Science

The Department of Computer Science currently offers a Computer Science Specialist and Major in addition to a set of combined specialists and other programs.[1] As a result of curriculum review, the Department proposes a modification and consolidation of its existing programs to offer a single Computer Science Specialist, Major and a new Minor.

The current collection is now confusing for students, who find it hard to distinguish some programs from others or who find it difficult to make choices that match their goals. In addition, some students in upper years find their choices constrained by a very specialized program. The consolidated Specialist, Major and Minor programs will avoid duplication amongst existing program offerings while still maintaining a strong emphasis on providing breadth and depth. Students will have greater flexibility in course choice in upper years. The Department has a collection of concentrations: sets of optional courses that can direct students toward expertise in particular areas of computer science in upper years, such as game design, theory of computation, and human-computer interaction. These concentrations are being more fully developed and would be explained on the Department website.

The Department would offer advice on course choice and concentration within the program and the proposed calendar entries direct students to advice from staff and faculty. The Department would also offer explicit advice on preparing for graduate studies as students often seem to be unaware of the possibility of further study.

Arts & Science students are increasingly enrolling in double major programs and there is no longer a need for separate Combined Specialists in the Department. In addition, as the discipline of computer science has become prevalent in relation to many disciplines, the Department does not consider it feasible design Combined Specialist programs with every discipline where interaction might be fruitful, and experience shows that such programs are difficult to maintain. The Department has consulted with associated cognate Departments in the Faculty and they are supportive in the concept of providing double Majors rather than separate combined programs. The Department has also consulted with the Faculty of Applied Science and Engineering, as Applied Science and Engineering students are enrolled in computer sciences courses. The program modifications will not impact these students as the computer science courses will still be available.

Specialist Program

Description and Calendar Entry
The proposed Specialist program encompasses a common first two years and greater freedom for students to choose their 300- and 400-level courses. **Specialist (12 FCEs)** First year (2.5 FCEs): 1. CSC (108H1, 148H)/150H1, 165H/240H1; MAT 135Y1/137Y1/157Y1 Second year (3.5 FCEs): 2. CSC 207H1, 209H1, 236H1/240H1, 258H1, 263H1/265H1; MAT 223H1/240H1; STA 247H1/255H1/257H1

[1] Artificial Intelligence, BSc: Spec; Computer Science, BSc: Spec, Maj; Computer Science-Foundations, BSc: Spec; Computer Science-Information Systems, BSc: Spec; Computer Science-Software Engineering, BSc: Spec; Computer Science & Economics, BSc: Spec; Computer Science & Mathematics, BSc: Spec; Computer Science & Physics, BSc: Spec; Computer Science & Statistics, BSc: Spec; Human-Computer Interaction, BSc: Spec.

Notes:

1. Students with a strong background in an object-oriented language such as Python, Java or C++ may omit CSC 108H1 and proceed directly with CSC 148H1. There is no need to replace the missing half-credit; however, please base your course choice on what you are ready to take, not on “saving” a half-credit.
2. CSC 150H1 is an accelerated alternative to CSC 108H1 and CSC 148H1, intended for students with previous programming experience in a procedural language. If you take CSC 150H1 instead of CSC 108H1 and 148H1, you do not need to replace the missing half-credit; but please see Note 1.
3. CSC 240H1 is an accelerated and enriched version of CSC 165H1 plus CSC 236H1, intended for students with a strong mathematical background, or who develop an interest after taking CSC 165H1. If you take CSC240H without CSC165H, there is no need to replace the missing half-credit; but please see Note 1.
4. Students may not omit CSC 165H1 and proceed directly to CSC 236H1. Either (CSC 165H1 and CSC 236H1), or CSC 240H1 is required for program completion. CSC 165H1 is different from CSC 108H1 in this respect.
5. Consult the Undergraduate Office for advice about choosing among CSC 108H1, CSC 148H1, and CSC 150H1, and between CSC 165H1 and CSC 240H1.

Later years (6 FCEs):

3. CSC 369H1, 373H1/375H1
4. 1.5 FCEs from the following: any 400-level CSC course; BCB 410H1, 420H1, 430Y1, with not more than 1.0 FCE from CSC 490H1, 491H1, 494H1, 495H1, BCB 430Y1
5. 1.5 additional FCEs from the following: any 300+level CSC course; BCB 410H, 420H, 430Y; ECE 385H, 489H
6. 2 additional FCEs from the following list:
   CSC: any 300-/400-level
   BCB 410H1, 420H1, 430Y1
   ECE 385H1, 489H1
   MAT 224H1, 235Y1/237Y1/257Y1, any 300-/400-level except 329H1, 390H1, 391H1
   STA 248H1/261H1, any 300-/400-level

The choices in 4, 5 and 6 must satisfy the requirement for an integrative, inquiry-based activity by completing one of the following half-courses: CSC 404H1, 420H1, 454H1, 490H1, 491H1, 494H1, 495H1 This requirement may also be met by participating in the PEY (Professional Experience Year) program.

Choosing courses

This program offers considerable freedom to choose courses at the 300+level, and you are free to make those choices on your own. We are eager to offer guidance and both our Undergraduate Office and individual faculty members are a rich source of advice.

We have also created a collection of what we call “concentrations”: sets of courses that direct you toward expertise in particular areas of Computer Science, such as game design, theory of computation, human-computer interaction, and many more. These concentrations are meant to help your choice, not to constrain it, and each concentration has at least one faculty member who would be happy to discuss it with you.

To read more about the concentrations, please consult our web site at http://web.cs.toronto.edu/program/ugrad.htm

Preparing for graduate study in Computer Science

Strong students should consider the option of further study in graduate school (where the degrees offered are typically M.Sc. and Ph.D.). If you find yourself frequently receiving marks in the B+ range or better, you should consult with faculty members to learn more about graduate school and whether it would be a good option for you. You will want to ask for advice on your particular interests — and you will find faculty members are happy to talk to you — but there are also some course choices that should be considered by all students thinking of graduate study in Computer Science.

Here are some courses that any CS graduate student should have taken, even if their interests lie in other areas of Computer Science: CSC 324H1, 336H/350H1, 343H1, 373H1, 369H1, and 463H1. (Some of those courses are required in the Specialist program.)

The research areas of CS can be grouped in different ways, but commonly used groupings include: systems, theory, networks, software engineering, artificial intelligence, numerical analysis, human-computer interaction, and graphics. Your course work will probably have given you an idea of where your own interests lie, but you should not take courses exclusively in one area. You will benefit by having taken an advanced course requiring considerable software development and a theoretical course.

It will be especially beneficial to have done a project course (CSC 494H1/495H1), a capstone course (CSC 490H1/491H1), and/or a summer research project. It is good if this individual work is in the area where you eventually decide you'd like to do your own research, but that is not essential; what you need most is some experience doing work on your own, under the mentorship of an experienced researcher.

Academic Context

This program is designed for students making Computer Science their prime choice as undergraduates, though it is also designed to accommodate significant interests in other fields. The goal is to prepare you to go on to a professional career without further study, to pursue graduate work in Computer Science or a related field, or to take the skills and understanding acquired here, and apply them in other areas.

The required courses in the program, mostly taken in the first two years of a four-year degree, provide a solid grounding in the core of Computer Science (CS). To understand the selection of required courses, you need to know that CS has two main branches: “theoretical” or “mathematical”, and “applied” or “systems”. In first year, the theoretical branch is represented by CSC 165H1, which introduces the mathematical skills and outlook needed to appreciate rigorous arguments, while CSC 108H1 and 148H1 begin the applied branch with an introduction to programming and data structures. The branches are not completely distinct: you will find programming examples in CSC 165H1 and mathematical argument in CSC 108H1 and 148H1. However, the two approaches begin here.

In second year, the applied branch continues with CSC 207H1 and 209H1, and the theoretical branch with CSC 236H1. CSC 263H1 brings the two together, applying mathematical analysis to more advanced data structures. CSC 258H1 presents the physical machinery lying behind computation.

Computer scientists use more traditional mathematics too, and the program includes calculus (at least at the introductory level), linear algebra, and probability. These topics are needed for algorithm analysis, numerical methods and also various applied areas.

In your third and fourth years, you are largely responsible for your own course choices, with our advice. CS is relevant in such a large range of other disciplines and professional contexts, and can be used in such a wide variety of careers, that we try not to limit your choices. We offer considerable advice on course choice in the calendar and on our web site, and you can also consult faculty members and our Undergraduate Office for more individual help.

As you seek this advice, remember that you are required to finish your program with a set of 400-level CSC or BCB courses, and that you should build toward these courses — all of which have prerequisites — by your selection of earlier courses. Many advanced courses share prerequisites, so your 300-levels do not precisely determine your 400-levels, but you do need to look ahead. We are happy to help with that.

Learning Outcomes

After completing the required courses in the Specialist program, you will be able to program in at least two programming languages; argue about and analyze mathematically the correctness and performance of standard algorithms and data structures; understand how the physical structure of a computer and the facilities offered by its operating system support the execution of programs; work well with others on a team building a significant piece of software; and communicate effectively with co-workers and customers.

Beyond these basics, a Specialist graduate will understand one area of CS in considerable depth, having reached the point of taking more than one 400-level course in an area of research or professional activity, or else will have reached moderate depth in a couple of areas. In either case, you will be prepared either to pursue graduate studies — not necessarily in one of the depth areas, because acquiring deep knowledge is a transferable skill — or to follow a career in computing.

Degree Objectives

c. DEPTH OF KNOWLEDGE

By the end of second year, students acquire theoretical and applied knowledge across the basics of computer science. The courses they choose at the 300 and 400 levels provide depth in at least one area of CS. If their 400-levels are in a single area, then there will be not just depth but also a kind of very specialized breadth; if in two or more areas, then there will be more than adequate depth in a couple of areas.

An essential point here is that no 400-level course exists by itself; all have prerequisites, and students acquire depth in the process of completing the prerequisites as well as while taking the 400-levels themselves.

d. COMPETENCIES

i. Critical and Creative Thinking

Computer scientists need to read, criticize and create mathematical arguments, English specifications, program code, consumer demands, and proposals linking computing with a wide variety of other disciplines.

The introductory programming courses, CSC 108H1 and 148H1, are mostly about writing code rather than reading it, but even there students need to read specifications carefully and quickly learn to question and criticize what they are asked to do. When they reach the following software-based course, CSC 207H1, they are required to integrate both code and prose written by different team members, and begin to appreciate that careful reading and a willingness to understand different approaches to the same problem are essential abilities. In CSC 369H1, students must read and analyze a large software system. They also create and carry out experiments to critically analyze the behaviour of the system.

On the theoretical side, in CSC 165H1, students learn to analyze mathematical arguments at an introductory level and to write about the experience. The activity of writing leads for many of them to a deeper understanding of how to think about the proof process.

We often see original software projects carried out by our students in first and second year, but for most, the chance to create does not arise until later years. Creative thinking does not come from any specific set of courses, because the choice of 300- and 400-level courses is up to the student: but we think that every path through our program encounters the need for originality at some point, because, at the very least, our 400-level courses meet this requirement.

ii. Communication

Our students learn to communicate in at least three languages: mathematics, code and English. They begin learning to write computer programs in CSC 108H, and to read them in CSC 207H1. In CSC 165H1, we teach them to analyze mathematical arguments at an introductory level. Those two communicative skills are developed further in all later CSC courses.

The ability to read and write effectively in English is important to our students and our graduates. This is often a surprise to beginners, but it is a point insisted on by employers, and from the beginning we teach and require those skills. In CSC 165H1, we teach not just mathematical ideas but the discussion of mathematical ideas through written exercises including wikis or student journals. In CSC 207H1 student programming teams must produce user documentation and in addition written discussions by email and on group wikis throughout the programming process. In CSC 369H1, students write substantial documents that explain and justify their design decisions.

Under our new program, it becomes difficult by third year to point to courses required of all students, but the use of writing is growing and becoming more firmly embedded in courses such as CSC 300H1 (with essays), 301H1 (software development plans), 404H1 (user guides and online documentation), 420H1 (research reports), and 454H1 (business plans), as well as our individual project and capstone courses.

iii. Information Literacy

A great many of the resources in our field are not in books or even in libraries. From the beginning, we expect our students to install software, find example code, and look for explanations that are nowhere available on paper or even on reasonably static web sites. We direct them to sources of the code and information they require, show them how to use these resources during in-class demonstrations, and let them know we are there to help if they have trouble.

In Computer Science, and in the computing world generally, intellectual property rights are of course a major issue. We do not try to teach these topics in a separate module or course; instead, they arise naturally as we promote the use of legitimately free software, work with companies such as Microsoft to make available licensed student versions of their tools, and remind students that they in fact own the work they submit to us for evaluation.

As an example, the required course CSC 369H1 teaches operating system topics using a simplified version of an open-source OS that many of our students are familiar with. The current version includes changes made by our faculty members under the open source license. The point of the course is neither the specific OS itself nor open source licenses, but those topics are inevitably mentioned in the course. One cannot teach Computer Science without encountering this issue.

iv. Quantitative Reasoning

CSC 165H1 ("Mathematical Expression and Reasoning for Computer Science") is sufficient to provide this competency, but we do also require courses in MAT and STA as well as in CSC. Analysis of algorithms is entirely a quantitative endeavour — how much of a computing resource is needed to reach a conclusion? — and by the end of second year our students are certainly adept at both exact computation and order-of-magnitude estimation.

v. Social and Ethical Responsibility

Computer scientists are professionals, and the work they do affects their employers, their employers' customers, consumers of all kinds of products, and regular people everywhere. Choices made in designing, coding and testing software, and even in choosing mathematical models, affect the success, reliability and safety of software and hardware products, and the health, happiness and prosperity of users.

Computer scientists are also creators of intellectual products, and need to be aware of intellectual property issues. At the same time, we and our students are consumers of intellectual property, subject to the usual temptations but equipped with more than the usual skills, and the question of whether to pay for software tools or to seek free equivalents often comes up in our classes. We try to help with the practical questions of the search for software, but also to teach to the underlying issues.

We talk about these questions with our students from the beginning of their studies here — not in separate modules on ethics but in daily interactions in class. We teach software development by doing it while students participate in making choices, and the choices range from the technically low-level to the user experience: “Which way makes the program run faster?” “What does it cost?” “Is it acceptable for execution to take that long?" “What’s a good error message?” “How should the warning appear?” “What’s the user likely to be thinking?” “Which part of the program most needs to be reliable?” “What if the user is colour-blind? blind? deaf?”

This begins already in CSC 108H1 and continues in CSC 148H1. By CSC 207H1 there are classes explicitly about eliciting instructions from and reporting to customers and managers, and about interacting with team members to ensure designs are faithfully represented in implementations.

CSC 165H1, 236H1 and 263H1 focus on proofs, code correctness and algorithm analysis, where the human user is much less in sight; but even there, the goals of correctness and performance are not merely abstract.

There are optional upper-level courses that build further on social and ethical topics. CSC 300H1 ("Computers and Society") by itself offers a sufficient education in the ethical responsibilities of computer scientists.. CSC 318H1 (an introductory course on human-computer interaction) helps students learn to meet the human rather than insist on the priority of computing. CSC 404H1 (“Introduction to Video Game Design”) discusses the social impact of video games.

e. AN INTEGRATIVE, INQUIRY-BASED ACTIVITY

Some of our 400-level courses already require significant investigative work, followed by a reporting process consisting of a written report and often an oral presentation. Courses in this category include the "project" courses (CSC 494H1 and 495H1), the "capstone" courses (CSC 490H1 and 491H1), and some courses on specific topics, such as CSC 404H1 (“Introduction to Video Game Design”), CSC 420H1 (“Introduction to Image Understanding”), and CSC 454H1 (“The Business of Software”). We are encouraging instructors of our other 400-level courses, many of which already build on earlier studies and require student investigation, to move toward meeting the criteria for Integrative, Inquiry-Based Activities, but the process is far from complete.

A considerable fraction of our students — somewhere between a quarter and a half — take part in the Professional Experience Year (PEY) program. Doing a PEY has an effect on participants that pulls together their academic studies in the discipline, applies it to interesting or at least useful problems, and teaches the benefits of professional communication. For them, the PEY is an integrative, inquiry-based activity.

Departmental/College Resource Implications	
Estimated Enrolment per Academic Year in this program (please explain)	About 200 in Specialist and Major combined — the same as at present.
New courses necessary to mount for this program	None.
Additional Instructor(s) Requirements	None.
Teaching Assistant(s) Requirements	Same as at present.
Laboratory Equipment Requirements	Same as at present.
Computing Resources Requirements	Same as at present.

No new resources are required as the program because it replaces existing programs.

Major Program

Description
Major (8 credits) First year (2.5 FCEs): 1. CSC (108H1, 148H)/150H1, 165H1/240H1; MAT 135Y1/137Y1/157Y1 Second year (2.5 FCEs): 2. CSC 207H1, 236H1/240H1, 258H1, 263H1/265H1; STA 247H1/255H1/257H1 See notes 1-5 under the Specialist program (repeated here for the POSt Advisory Committee). <u>Notes:</u> 6. Students with a strong background in an object-oriented language such as Python, Java or C++ may omit CSC 108H1 and proceed directly with CSC 148H1. There is no need to replace the missing half-credit; however, please base your course choice on what you are ready to take, not on “saving” a half-credit. 7. CSC 150H1 is an accelerated alternative to CSC 108H1 and CSC 148H1, intended for students with previous programming experience in a procedural language. If you take CSC 150H1 instead of CSC 108H1 and 148H1, you do not need to replace the missing half-credit; but please see Note 1. 8. CSC 240H1 is an accelerated and enriched version of CSC 165H1 plus CSC 236H1, intended for students with a strong mathematical background, or who develop an interest after taking CSC 165H1. If you take CSC240H without CSC165H, there is no need to replace the missing half-credit; but please see Note 1. 9. Students may not omit CSC 165H1 and proceed directly to CSC 236H1. Either (CSC 165H1 and CSC 236H1), or CSC 240H1 is required for program completion. CSC 165H1 is different from CSC 108H1 in this respect. 10. Consult the Undergraduate Office for advice about choosing among CSC 108H1, CSC 148H1, and CSC 150H1, and between CSC 165H1 and CSC 240H1. Later years (3 FCEs). 3. 0.5 FCE from the following: any 400-level CSC course; BCB 410H1, 420H1, 430Y1 4. 1.0 additional FCE from the following: any 300+level CSC course; BCB 410H1, 420H1, 430Y1; ECE 385H1, 489H1 5. 1.5 additional FCEs from the following list, of which at least 0.5 FCE must be at the 300-/400-level: CSC: any 200-/300-/400-level BCB 410H1, 420H1, 430Y1 ECE 385H1, 489H1 MAT 223H1/240H1, 235Y1/237Y1/257Y1, any 300-/400-level except 329H1, 390H1, 391H1 The choices in 3, 4, and 5 must satisfy the requirement for an integrative, inquiry-based activity by completing one of the following half-courses: CSC 404H1, 420H1, 454H1, 490H1, 491H1, 494H1, 495H1 This requirement may also be met by participating in the PEY (Professional Experience Year) program. **Advice on choosing courses towards a Major in CSC** A Major program in any discipline may form part (but not the whole) of your degree requirements. The Major program in Computer Science is designed to include a solid grounding in the essentials of CS, followed by options that let you explore one or a few topics more deeply. You will also realize what areas you have not studied, and be ready to explore them if your interests change after completing the Major.

To give you freedom to choose your path through Computer Science, we have designed the Major to include a minimal set of required courses. There are some courses that we think you ought to consider carefully as you make those choices. CSC 373H1 is fundamental to many more advanced CSC topics, where designing appropriate algorithms is central. CSC 209H1 is a prerequisite to effective work in many application areas..

We have designed “packages” of related courses that are intended to accompany the Specialist program in Computer Science, and you may find them helpful in completing your Major too. Please see our web site at http://web.cs.toronto.edu/program/ugrad.htm

A significant role of the Major is to allow you to integrate your studies in CS and another discipline. For example, many CS students are also interested in statistics, economics, physics or mathematics. In those cases, it makes sense to enrol in a Major in one discipline and either a Major or a Specialist in the other. If your interests are evenly balanced, the obvious choice is to do two Majors, and that is what we assume here.

If you are doing a double Major (two Majors in related disciplines), you might want to consult your college registrar’s office for advice on satisfying the degree requirements with overlapping Majors.

A Major program is generally not enough to prepare you for graduate study in Computer Science, though a complete Specialist is not required. Please consult the advice about graduate study included with the description of the Specialist program in Computer Science.

CSC and Mathematics

The theoretical foundations of Computer Science are essentially a branch of mathematics, and numerical analysis (the area of CS that studies efficient, reliable and accurate algorithms for the numerical solution of continuous mathematical problems) is also a topic in applied mathematics. If you are interested in both CS and mathematics, a double major is a good choice.

In this double major, you should choose all the theoretical courses in the first three years: CSC 165H1, 236H1, 263H1, 373H1 and 363H1. If the "enriched" versions are available as alternatives, you should prefer them: CSC 240H1 in place of CSC 165H1 and 236H1, and CSC 265H1, 375H1 and 365H1 in place of CSC 263H1, 373H1 and 363H1 respectively. If you come to realize that your interests are mathematical after taking some of the non-enriched courses, it's not too late; you should ask us for advice. You should also take at least one of CSC 438H1, 448H1 and 465H1.

You should also make sure you take courses in numerical analysis -- CSC 336H1 and 351H1, and possibly CSC 446H1.

In the Major in Mathematics, you should prefer courses that are also in the Specialist program in Mathematics: MAT 157Y1, 240H1, 247H1 and so on. Ask the advisors in the Department of Mathematics which courses they would recommend if you're planning a career in mathematics. Don't be afraid to admit your interest in CS.

CSC and Bioinformatics/Computational Biology

Bioinformatics is a field that came into existence only in the 1990s but has become an extremely fruitful interaction between biological scientists and computer scientists. Deciphering the genome requires not just extremely clever biology but extremely clever computer science, drawing from the study of algorithms and data structures and from data mining.

To study bioinformatics, you should enrol in the Major program in Bioinformatics and Computational Biology sponsored by the Department of Biochemistry, and also in the Major in Computer Science. Your CSC Major should include a selection of courses something like this:

CSC 373H1/375H1
CSC 321H1/343H1
CSC 336H1/350H1
BCB 410H1, 420H1
CSC 411H1
Some of CSC 324H1, 363H1, 310H1, 412H1, 456H1

You should seek advice from both the Department of Biochemistry and the Department of Computer Science on how to distribute your courses across the two Majors.

CSC and Statistics
Here your CS course choices should be somewhat similar to those for CSC and Mathematics: take the theoretical CSC courses up to the 300 level, and prefer the higher-level MAT and STA courses. For example, take STA 257H1 and 261H1 rather than STA 247H1 and 248H1.

Within Computer Science, take courses in numerical analysis (CSC 336H1 and 351H1). Choose also from among information theory (CSC 310H), machine learning (CSC 321H1 and 411H1), and natural language processing (CSC 401H).

CSC and Economics
There is considerable opportunity for mutually supporting interests in CS and economics in the area of economic modelling, econometrics and numerical analysis. In CS, you might choose courses such as CSC 343H1 (databases), 358H1 (networks) and 369H1 (operating systems) to acquire the technical background for working with large systems and data sets, and CSC 350H1 and 351H1 (numerical analysis) to understand the difficulties of large numerical models.

If you are interested in financial modelling, you will also want to take CSC 446H1 to learn how to handle partial differential equations; to do that, you would want to have taken the necessary mathematical courses.

Applying ideas from economics to CS is a little harder, but certainly economic principles apply to databases (CSC 443H1) and networks (CSC 458H1). CSC 358H1 discusses how to model the processes involved in computer networks and in other customer-server systems. CSC 454H1 (business of software) would also benefit from your background in economics.

CSC and Linguistics
If you are interested in both CS and Linguistics, you should consider doing a Major in both. Your Major in Computer Science should focus on computational linguistics (CL), the sub-area of AI concerned with human languages (“natural languages”); researchers in this area are interested in developing programs that can “understand” and generate natural language. You should take our CL courses, CSC 401H1 and 485H1. (They can be taken in either order.) As preparation, you should also take CSC 324H1 (programming languages). Other courses you might find valuable are CSC 384H1 (AI), 343H1 (databases), and the theoretical courses CSC 373H1/375H1 and 363H1/365H1.

CSC and Physics
If you want to study CS and physics, then as a physicist, you will be interested in how natural processes and human design can take us from the materials and laws of nature to useful computational machinery, and you will want to study CSC 258H (computer organization -- the way solid-state devices can be combined to build a machine that repeatedly executes instructions) and CSC 369H (operating systems -- the large software systems that organize the programs people write and run to present the appearance of a well-run self-policing machine).

As a computer scientist, you will wonder how accurately you can compute the results of calculations needed in simulating or
predicting physical processes. CSC 350H1 and 351H1 introduce you to numerical analysis, and CSC 446H1 applies it to partial differential equations, used to model many physical systems.

Both a computer scientist and a physicist will wonder how to write effective programs. CSC 263H1 and 373H1 teach you to choose appropriate data structures and algorithms, and CSC 363H1 helps you to understand whether a problem is computable, and if so, whether the computation takes a reasonable amount of time.

In fourth year, you may choose CSC 418H1, which depends on and also simulates the behaviour of light and mechanical systems. CSC 456H1 deals with high-performance computing of the kind used in scientific computing. CSC 420H1 might also be a good choice, though some preparation in artificial intelligence would be helpful.

2. Academic Context

This program is designed for students who want to become proficient in Computer Science while also pursuing significant interests in other fields. The goal is to prepare you to go on to a professional career without further study or to take the skills and understanding acquired here and apply them in other areas. To pursue graduate study in Computer Science, you would need some further preparation, and you should consult the advice accompanying our Specialist program.

The required courses in the Major, mostly taken in the first two years of a four-year degree, provide a solid grounding in the core of Computer Science (CS). To understand the selection of required courses, you need to know that CS has two main branches: “theoretical” or “mathematical”, and “applied” or “systems”. In first year, the theoretical branch is represented by CSC 165H1, which introduces the mathematical skills and outlook needed to appreciate rigorous arguments, while CSC 108H1 and 148H1 begin the applied branch with an introduction to programming and data structures. The branches are not completely distinct: you will find programming examples in CSC 165H1 and mathematical argument in CSC 108H1 and 148H1. However, the two approaches begin here.

In second year, the applied branch continues with CSC 207H1 and 209H1, and the theoretical branch with CSC 236H1. CSC 263H brings the two together, applying mathematical analysis to more advanced data structures. CSC 258H1 presents the physical machinery lying behind computation.

Computer scientists use more traditional mathematics too, and the program includes calculus (at least at the introductory level) and probability. These topics are needed for algorithm analysis, numerical methods and also various applied areas. If your interests are on the theoretical side, you may wish to include linear algebra (MAT 223H1) as one of your options.

In your third and fourth years, you are largely responsible for your own course choices, with our advice. CS is relevant in such a large range of other disciplines and professional contexts, and can be used in such a wide variety of careers, that we try not to limit your choices. We offer considerable advice on course choice in the calendar and on our web site, and you can also consult faculty members and our Undergraduate Office for more individual help.

As you seek this advice, remember that you are required to finish your program with a 400-level CSC or BCB course, and that you should build toward this course by including its prerequisites among your earlier courses. Many advanced courses share prerequisites, so your 300-levels do not precisely determine your 400-level choice, but you do need to look ahead. We are happy to help with that.

Learning Outcomes

After completing the required courses in the Major program, you will be able to program in at least two programming languages; argue about and analyze mathematically the correctness and performance of standard algorithms and data structures; understand how the physical structure of a computer and the facilities offered by its operating system support the execution of programs; work well with others on a team building a significant piece of software; and communicate effectively with co-workers and customers.

Beyond these basics, after completing the Major you will understand one area of CS in reasonable depth, having reached the point of taking a 400-level course in an area of research or professional activity. You will be prepared for a career in computing, possibly by applying your background in Computer Science to problems in a field from your other Major; or you will find yourself a very competent software developer or system administrator. If you find yourself wanting to learn more about Computer Science, taking a few courses beyond the minimum required will prepare you for graduate study.

Degree Objectives

f. DEPTH OF KNOWLEDGE

By the end of second year, students will acquire theoretical and applied knowledge across the basics of computer science. The courses they choose at the 300 and 400 levels provide depth in at least one area of CS. An essential point here is that no 400-level course exists by itself; we try to get our course prerequisites right, and students acquire depth in the process of completing the prerequisites as well as while taking the 400-level itself.

g. COMPETENCIES

i. Critical and Creative Thinking

Computer scientists need to read, criticize and create mathematical arguments, English specifications, program code, consumer demands, and proposals linking computing with a wide variety of other disciplines.

The introductory programming courses, CSC 108H1 and 148H1, are mostly about writing code rather than reading it, but even there students need to read specifications carefully and quickly learn to question and criticize what they are asked to do. When they reach the following software-based course, CSC 207H, they are required to integrate both code and prose written by different team members, and begin to appreciate that careful reading and a willingness to understand different approaches to the same problem are essential abilities.

On the theoretical side, in CSC 165H1, students learn to analyze mathematical arguments at an introductory level and to write about the experience. The activity of writing leads for many of them to a deeper understanding of how to think about the proof process.

We often see original software projects carried out by our students in first and second year, but for most, the chance to create does not arise until later years. Creative thinking does not come from any specific set of courses, because the choice of 300- and 400-level courses is up to the student: but we think that every path through our program encounters the need for originality at some point, because, at the very least, our 400-level courses meet this requirement.

ii. Communication

Our students learn to communicate in at least three languages: mathematics, code and English. They begin learning to write computer programs in CSC 108H, and to read them in CSC 207H1. In CSC 165H1, we teach them to analyze mathematical arguments at an introductory level. Those two communicative skills are developed further in all later CSC courses.

The ability to read and write effectively in English is important to our students and our graduates. This is often a surprise to beginners, but it is a point insisted on by employers, and from the beginning we teach and require those skills. In CSC 165H1, we teach not just mathematical ideas but the discussion of mathematical ideas through written exercises including wikis or student journals. In CSC 207H1 student programming teams must produce user documentation and in addition written discussions by email and on group wikis throughout the programming process.

Under our new program, it becomes difficult by third year to point to courses required of all students, but the use of writing is growing and becoming more firmly embedded in courses such as CSC 300H1 (with essays), 301H1 (software development plans), 404H1 (user guides and online documentation), 420H1 (research reports), and 454H1 (business plans), as well as our individual project and capstone courses.

iii. Information Literacy

A great many of the resources in our field are not in books or even in libraries. From the beginning, we expect our students to install software, find example code, and look for explanations that are nowhere available on paper or even on reasonably static web sites. We direct them to sources of the code and information they require, show them how to use these resources during in-class demonstrations, and let them know we are there to help if they have trouble.

In Computer Science, and in the computing world generally, intellectual property rights are a major issue.

These topics arise naturally as we promote the use of legitimately free software, work with companies such as Microsoft to make available licensed student versions of their tools, and remind students that they in fact own the work they submit to us for evaluation.

iv. Quantitative Reasoning

CSC 165H1 ("Mathematical Expression and Reasoning for Computer Science") is sufficient to provide this competency, but we do also require courses in MAT and STA as well as in CSC. Analysis of algorithms is entirely a quantitative endeavour — how much of a computing resource is needed to reach a conclusion? — and by the end of second year our students are certainly adept at both exact computation and order-of-magnitude estimation.

v. Social and Ethical Responsibility

Computer scientists are professionals, and the work they do affects their employers, their employers' customers, consumers of all kinds of products, and regular people everywhere. Choices made in designing, coding and testing software, and even in choosing mathematical models, affect the success, reliability and safety of software and hardware products, and the health, happiness and prosperity of users.

Computer scientists are also creators of intellectual products, and need to be aware of intellectual property issues. At the same time, we and our students are consumers of intellectual property, subject to the usual temptations but equipped with more than the usual skills, and the question of whether to pay for software tools or to seek free equivalents often comes up in our classes. We try to help with the practical questions of the search for software, but also to teach to the underlying issues.

We talk about these questions with our students from the beginning of their studies here — not in separate modules on ethics but in daily interactions in class. We teach software development by doing it while students participate in making choices, and the choices range from the technically low-level to the user experience: "Which way makes the program run faster?" "What does it cost?" "Is it acceptable for execution to take that long?" "What's a good error message?" "How should the warning appear?" "What's the user likely to be thinking?" "Which part of the program most needs to be reliable?" "What if the user is colour-blind? blind? deaf?"

This begins already in CSC 108H and continues in CSC 148H. By CSC 207H there are classes explicitly about eliciting instructions from and reporting to customers and managers, and about interacting with team members to ensure designs are faithfully represented in implementations.

CSC 165H1, 236H1 and 263H1 focus on proofs, code correctness and algorithm analysis, where the human user is much less in sight; but even there, the goals of correctness and performance are not merely abstract.

There are optional upper-level courses that build further on social and ethical topics. CSC 300H1 ("Computers and Society") by itself offers a sufficient education in the ethical responsibilities of computer scientists. CSC 318H1 (an introductory course on human-computer interaction) helps students learn to meet the human rather than insist on the priority of computing. CSC 404H1 ("Introduction to Video Game Design") discusses the social impact of video games.

h. AN INTEGRATIVE, INQUIRY-BASED ACTIVITY

Some of our 400-level courses already require significant investigative work, followed by a reporting process consisting of a written report and often an oral presentation. Courses in this category include the "project" courses (CSC 494H1 and 495H1), the "capstone" courses (CSC 490H1 and 491H1), and some courses on specific topics, such as CSC 404H ("Introduction to Video Game Design"), CSC 420H1 ("Introduction to Image Understanding"), and CSC 454H1 ("The Business of Software"). We are encouraging instructors of our other 400-level courses, many of which already build on earlier studies and require student investigation, to move toward meeting the criteria for Integrative, Inquiry-Based Activities, but the process is far from complete.

A considerable fraction of our students — somewhere between a quarter and a half — take part in the the Professional Experience Year (PEY) program. Doing a PEY has an effect on participants that pulls together their academic studies in the discipline, applies it to interesting or at least useful problems, and teaches the benefits of professional communication. For them, the PEY is an integrative, inquiry-based activity.

Departmental/College Resource Implications	
Estimated Enrolment per Academic Year in this program (please explain)	About 200 in Specialist and Major combined — the same as at present.
New courses necessary to mount for this program	None.
Additional Instructor(s) Requirements	None.
Teaching Assistant(s) Requirements	Same as at present.
Laboratory Equipment Requirements	Same as at present.
Computing Resources Requirements	Same as at present.
Other	None.

No new resources are required as the program because it replaces existing programs.

Minor Program

Computer Science currently offers a Specialist and Major program. Many students, not enrolled in these programs, are currently taking computer science 200-level courses purely out of interest. These students have often expressed an interest in a Minor.

Description and Calendar Entry

Minor (4 credits)

CSC (108H1, 148H1)/150H1, 165H1/240H1, 207H1, 236H1/240H1

Notes:

11. Students with a strong background in Java or C++ may omit CSC 108H1 and proceed directly with CSC 148H1.
12. CSC 150H1 is an accelerated alternative to CSC 108H1 and CSC 148H1, intended for students with previous programming experience in a procedural language.
13. CSC 240H1 is an accelerated and enriched version of CSC 165H1 plus CSC 236H1, intended for students with a strong mathematical background, or who develop an interest after taking CSC 165H1.
14. Students may not omit CSC 165H1 and proceed directly to CSC 236H1. Either (CSC 165H1 and CSC 236H1), or CSC 240H1 is required for program completion. CSC 165H1 is different from CSC 108H1 in this respect.
15. Consult the Undergraduate Office for advice about choosing among CSC 108H1, CSC 148H1, and CSC 150H1, and between CSC 165H1 and CSC 240H1.

(Total of above: 2.5 credits. If you take fewer than 2.5 credits, you must take more than 1.5 from the next list, so that the total is 4 credits.)

1.5 credits from the following list, of which at least 1 credit must be at the 300-/400-level:
CSC 209H1, 258H1, 263H1/265H1
CSC: any 300-/400-level

Academic Context

The Minor in Computer Science provides the basics of,Computer Science, with an introduction to the two major branches of the discipline, the “theoretical” or “mathematical”, and the “applied” or “systems”. The theoretical branch is represented by CSC 165H1 and 236H1, and the applied branch by CSC 108H1, 148H1 and 207H1. The program requires too few courses to be a complete preparation for a career in computing, but it teaches you enough to allow you to understand where Computer Science can lead you and to choose courses that meet your interests, perhaps making connections with another discipline.

Learning Outcomes

After completing the required courses in the Minor program, you will be able to program in at least two programming languages; argue about and analyze mathematically the correctness and performance of some standard algorithms and data structures; work well with others on a team building a significant piece of software; and communicate effectively with co-workers and customers.

The optional courses cover a broad range, from software construction to mathematical analysis, with application areas that combine the two approaches. If you have another area of interest, you may be able to choose CSC courses that interact with that interest.

The one credit at the 300 level can be chosen from a very wide range of courses. Some courses have too many prerequisites to be met within the program requirements, but there are quite a number of variations possible without taking courses beyond the required minimum.

Departmental/College Resource Implications

Estimated Enrolment per Academic Year in this program (please explain)	The Minor may attract between 20 and 100 students based on non-computer science student participation in CS courses and on expressed interest.
New courses necessary to mount for this program	None.
Additional Instructor(s) Requirements	None.
Teaching Assistant(s) Requirements	Same as at present.
Laboratory Equipment Requirements	Same as at present.
Computing Resources Requirements	Same as at present.

The Department of Computer Science will provide the resources required for this Program from its existing budget.