Major Modifications Proposal Guidelines

1. **Program:** Electrical Engineering and Computer Science

2. **Degree Designation:** PhD

3. **Type of Modification:**

   This proposal presents a modification of an existing program with the aim of keeping the program relevant, current and sustainable. In particular, the following modifications are proposed.

   - Change of the name of the degree program from "PhD in Computer Science" to "PhD in Electrical Engineering and Computer Science."

   - Introduction of the following fields: Computer Engineering, Computer Science, Electrical Engineering, and Software Engineering.

   - Change of the admission requirements.

   - Change of the degree requirements.

4. **Effective Date:** Fall 2017

5. **Provide a general description of the proposed changes to the program.**

   A. Change the name of the degree program from "PhD in Computer Science" to "PhD in Electrical Engineering and Computer Science."

   B. Introduction of the following fields: Computer Engineering, Computer Science, Electrical Engineering, and Software Engineering.

   C. Changes of the admission requirements.

      (1) Instead of "Applicants must have an MSc degree equivalent to the MSc Computer Science degree (thesis option) at York University" it is required that "Applicants must have a Masters's degree in Computer Engineering, Computer Science, Electrical Engineering, Software Engineering or closely related field, which is equivalent to the MSc degree in Computer Science (thesis option) or the MASc degree in Electrical and Computer Engineering degree at York University."

      (2) A breadth statement is not required any more as part of the application.

   D. Changes to the degree requirements.

      (1) The breadth and depth requirements related to the graduate courses are dropped.

      (2) Each term, students are required to attend departmental seminars.

      (3) Each fall and winter term, students have to attend at least one professional development workshop.

6. **Provide the rationale for the proposed changes.**

A. A few years ago, the “Department of Computer Science and Engineering” was renamed to “Department of Electrical Engineering and Computer Science” to reflect the fact that expertise of its members spans from Electrical Engineering to Computer Science. Currently, the Graduate Program in Electrical Engineering and Computer Science has a PhD degree in Computer Science. However, the program has PhD students doing their research on topics ranging from Electrical Engineering to Computer Science, under the supervision of members of our graduate program with expertise ranging from Electrical Engineering to Computer Science. Therefore, it is proposed to modify the name of the degree program from “PhD in Computer Science” to “PhD in Electrical Engineering and Computer Science.” The norm is to have separate programs in Computer Science and in Electrical and Computer Engineering. However, there are a few exceptions in North America with a combined program, such as the University of California at Berkeley.

B. The proposed PhD in Electrical Engineering and Computer Science covers a wide spectrum consisting of Electrical Engineering, Computer Engineering, Software Engineering and Computer Science. It is sometimes essential for students to have their field of specialization indicated on their transcript in order to successfully secure a job or work permit. Therefore, it is proposed to introduce the following fields: Computer Engineering, Computer Science, Electrical Engineering, and Software Engineering. Note that none of the admission or degree requirements is specific to a particular field. The qualifying exam, the dissertation proposal and the dissertation cover topics within the field.

C.

(1) Since the proposed PhD in Electrical Engineering and Computer Science covers not only Computer Science, but also Computer Engineering, Electrical Engineering and Software Engineering, we propose to adjust the admission requirements so that students with a Master’s degree in Computer Engineering, Electrical Engineering, Software Engineering or a closely related field are also admissible. This reflects our current admission practices.

(2) Currently, very few students provide a breadth statement as part of their applications. Therefore, it is proposed to drop this requirement. This proposed change has minimal resource implications (see item 11).

D.

(1) In the last decade, not a single student in our program had to take additional courses to satisfy the breadth and depth requirements. This is not a surprise since students admitted to our program have completed a Master’s degree in Computer Engineering, Computer Science, Electrical Engineering, Software Engineering or a closely related field. Therefore, we propose to simplify our degree requirements by dropping the breadth and depth requirements. Note that the depth and breadth of the students’ background will be evaluated during their qualifying examination. Thus the proposed change in degree requirements does not represent a change in the program standards with respect to a students’ background. This proposed change has minimal resource implications (see item 11).

(2) By requiring students to attend departmental seminars, they are exposed to research outside their area and, hence, increase their breadth. This requirement will also contribute towards research intensification in the department. This proposed change has minimal resource implications (see item 11), but is not expected to have any impact on the time to

completion.

(3) Developing professional training opportunities for graduate students is one of the action items of Lassonde's Integrated Resource Plan. To contribute towards this action item, it is proposed that students are required to attend professional development workshops. Relevant workshops are provided by the Faculty of Graduate Studies, Mitacs, Lassonde's Co-op program, York's Teaching Commons and the graduate program. Short online professional development courses, as provided by coursera and edX can also be used to satisfy this degree requirement. This proposed change has resource implications (see item 11), but is not expected to have any impact on the time to completion.

7. Comment on the alignment between the program changes with Faculty and/or University academic plans.

In light of the University Academic Plan, York's Plan for the Intensification and Enhancement of Research and the Faculty of Graduate Studies Integrated Resource Plan, this proposal directly speaks to research intensification and an expansion of engineering. Both themes are also key ingredients of the Provostial White Paper. According to York's Strategic Mandate Agreement, engineering is one of the proposed areas of growth. Engineering, as well as Computer Science also feature prominently in York's Strategic Research Plan.

In Lassonde's Strategic Research Plan, research in Electrical Engineering, as well as in Computer Engineering, Computer Science and Software Engineering, features prominently. This proposal addresses the intensification of research in those fields by establishing studies in Computer Engineering, Electrical Engineering and Software Engineering at the doctoral level. The integration of Electrical Engineering into Lassonde's graduate programs, which this proposal addresses, is one of the objectives of Lassonde's Integrated Resource Planning. Also in the Departmental Five-Year Plan, the integration of Electrical Engineering at the graduate level is one of the key objectives.

8. Provide a detailed outline of the changes to the program and the associated learning outcomes, including how the proposed requirements will support the achievement of program learning objectives (i.e., the mapping of the requirements to the program learning outcomes).

As part of this proposal, program learning outcomes and a mapping of the requirements to these program learning outcomes have been developed. These can be found in Appendix B.

9. Summarize the consultation undertaken with relevant academic units, including commentary on the impact of the proposed changes on other programs. Provide individual statements from the relevant program(s) confirming consultation and their support.

The proposed changes do *not* impact other programs.

Students and faculty members of the Graduate Program in Electrical Engineering & Computer Science were consulted on several occasions. On September 23, 2016 the faculty members of the program voted unanimously in favor of this proposal.

10. Are changes to the program's admission requirements being proposed coincident with the program change(s)? If so, outline the admission changes, and comment on the appropriateness of the revised requirements to the achievement of the program learning outcomes.

The changes to the admission requirements can be found in part C of item 5. These are consistent with the changes proposed in part A and B of item 5. Since the program learning outcomes have been developed as part of this proposal (see Appendix B), no revision of them has taken place.

11. Describe any resource implications and how they are being addressed (e.g., through a reallocation of existing resources). If new/additional resources are required, provide a statement from the relevant Dean(s)/Principal confirming resources will be in place to implement the changes.

A minimal increased need for resources related to the introduction of the new degree requirements (the attendance of departmental seminars and professional development workshops needs to be monitored by the graduate program assistant, two professional development workshops need to be given by the graduate program director) is offset by a minimal decrease in need for resources by dropping the admission requirement of a breadth statement (which will lessen the burden on the admission committee) and the degree requirement of breadth and depth related to the graduate courses (which does not need to be monitored any more by the graduate program assistant).

12. Is the mode of delivery of the program changing? If so, comment on the appropriateness of the revised mode(s) of delivery to the achievement of the program learning outcomes.

Apart from the introduction of the seminars and professional development workshops, the mode of delivery of the program does *not* change. Seminars are usually one hour in duration. On average, there is a departmental seminar every other week. Information about the Faculty of Graduate Studies workshops can be found at the URL gradstudies.yorku.ca/current-students/enhancing-your-experience/graduate-professional-skills. Information about the Mitacs workshops can be found at the URL www.mitacs.ca/en/programs/step/step-workshops. Information about the workshops of Lassonde's Co-op program can be found at the URL lassondecoop.com/student/resources.htm. Information about the workshops of the Teaching Commons can be found at the URL http://teachingcommons.yorku.ca/graduate-students/courses-workshops-and-events/workshops. Information about short online professional development courses can be found at the URLs http://www.coursera.org and http://www.edx.org. At the URL http://www.pre.ethics.gc.ca/eng/education/tutorial-didacticiel/ a short online course on research ethics can be found.

13. Is the assessment of teaching and learning within the program changing? If so, comment on the appropriateness of the revised forms of assessment to the achievement of the program learning outcomes.

Two new degree requirements are proposed: attendance of departmental seminars and completion of professional development workshops. Students have to complete a progress report each term. Each year, one of those three reports is the annual progress report which is evaluated by the supervisory committee. The other two are evaluated by the supervisor. All reports are reviewed by the graduate program director. Students have to document both the attendance of departmental seminars and the completion of professional development workshops in their progress reports by providing the relevant details and a short summary for each seminar and workshop. The former is associated with the program learning outcome "analyze and evaluate ideas presented by other researchers." The latter is associated with the program learning outcomes "evaluate how ethical, social, environmental, legal and regulatory influences may affect one's research" and "evaluate how non-compliance with relevant laws, regulations and intellectual property guidelines create risks in managing one's research." Both degree requirements are assessed (pass or fail) by the supervisor or supervisory committee when they meet with the student to discuss the progress report. If a student does not attend at least 50% of the departmental seminars (requirement D(2)) for a given term, the student receives a warning and needs to attend additional seminars next term to reach at least 50% of the departmental seminars of that given term and the next term. If a student does not meet requirement D(3) for a given term, the student receives a warning and needs to attend one extra workshop in the next term. If the student does not meet this amended requirement, then the student will be withdrawn from the program.

For the remainder, the assessment of teaching and learning within the program does *not* change.

14. Provide a summary of how students currently enrolled in the program will be accommodated.

Current students may follow the existing regulations. However, they have the opportunity to select a field in the fall of 2017. Furthermore, they are encouraged to satisfy the new degree requirements, described in part D(2) and D(3) of item 5, as those contribute to their breadth and professional development (see program learning outcomes).

15. Provide as an appendix a side-by-side comparison of the existing and proposed program requirements as they will appear in the Undergraduate or Graduate Calendar.

This comparison can be found in Appendix A.

Existing Program/Graduate Diploma Information (change from)	Proposed Program/Graduate Diploma Information (change to)
Electrical Engineering & Computer Science	**Electrical Engineering & Computer Science**
The Graduate Program in Electrical Engineering & Computer Science offers ~~courses and opportunities for advanced studies and research leading to~~ the degrees of Master of Science (MSc), Master of Applied Science (MASc) and Doctor of Philosophy (PhD). The MSc program covers ~~a wide variety of disciplines in~~ Computer Science. The MASc program concentrates on ~~Computer Systems Engineering, Electrical Engineering and Interactive Systems Engineering~~. The PhD program ~~concentrates on Theoretical Computer Science (especially, algorithms, complexity, computability, logic, parallel, concurrent and distributed computing), Intelligent and Interactive Systems (especially, artificial intelligence, computer vision, human-computer interaction, information retrieval, robotics, speech, virtual reality), and Systems Software and Hardware (especially, communications, data mining, databases, networks, signal processing and software engineering).~~	The Graduate Program in Electrical Engineering & Computer Science offers the degrees of Master of Science (MSc), Master of Applied Science (MASc) and Doctor of Philosophy (PhD). The MSc program covers Computer Science. The MASc program concentrates on Computer Engineering and Electrical Engineering. The PhD program covers Computer Engineering, Computer Science, Electrical Engineering and Software Engineering.
Master of Science Program	**Master of Science Program**
Admission Requirements	**Admission Requirements**
Graduates with an honours degree in Computer Science or equivalent, with at least a B+ average in the last two years of study, may be admitted as candidates for the Masters of Science program in computer science. In addition, those admitted must have completed the equivalent of a senior-level course in the area of theoretical computer science. The following are the minimum English Language test scores (if required): ~~Test of English as a Foreign Language~~ 577 or York English Language Test 4. The Graduate Record Examination general test and computer science subject test are strongly recommended, especially for applicants who did their work outside of Canada and/or the United States.	Graduates with an honors degree in Computer Science or equivalent, with at least a B+ average in the last two years of study, may be admitted as candidates for the Masters of Science program in computer science. In addition, those admitted must have completed the equivalent of a senior-level course in the area of theoretical computer science. The following are the minimum English language test scores (if required): TOEFL 577 (paper-based) or 90-91 (Internet-based), IELTS 7, or York English Language Test 4. The Graduate Record Examination general test and computer science subject test are strongly recommended, especially for applicants who did their work outside of Canada and/or the United States.
Degree Requirements	**Degree Requirements**
Students are expected to choose between the degree by thesis or by project before the end of their second term. There is a breadth requirement on the selected graduate courses. At least one course must be from each of the following three areas: - Theory of Computing & Scientific Computing - Artificial Intelligence & Interactive Systems - Systems: Hardware & Software No more than one-third of the course requirements can be integrated with undergraduate courses.	Students are expected to choose between the degree by thesis or by project before the end of their second term. There is a breadth requirement on the selected graduate courses. At least one course must be from each of the following three areas: - Theory of Computing & Scientific Computing - Artificial Intelligence & Interactive Systems - Systems: Hardware & Software No more than one-third of the course requirements can be integrated with undergraduate courses.
MSc Degree by Thesis	**MSc Degree by Thesis**
Candidates for the MSc degree must complete five graduate three-credit courses and successfully defend a master's thesis. Candidates must conduct a piece of approved research under the general direction of a supervisor. The	Candidates for the MSc degree must complete five graduate three-credit courses and successfully defend a master's thesis. Candidates must conduct a piece of approved research under the general direction of a supervisor. The

resulting thesis should demonstrate the Candidates’ research ability in the research subject. **MSc Degree by Project** Candidates for the MSc degree must complete seven graduate three-credit courses and conduct a research project. The research project will have a more limited scope and/or degree of originality than a thesis. The project is under the general direction of a supervisor. A paper describing the project must be submitted and graded by the supervisory committee. **Time Requirements** Students are expected to complete all of their master’s degree requirements in no more than five terms (twenty months). For more details refer to the program’s supplemental calendar. **Master of Applied Science Program** **Admission Requirements** Graduates with an honours undergraduate degree or equivalent (typically a four-year program with full-time enrolment) from an accredited university in electrical or computer engineering, with at least a B+ average in the last two years of study, may be admitted as candidates for the Master of Applied Science program in electrical and computer engineering. In addition, those admitted must have completed the equivalent of a senior-level project course or thesis in electrical or computer engineering. Significant industrial or research experience in electrical or computer engineering coupled with an honours undergraduate degree program or equivalent from an accredited university will be considered equivalent to an undergraduate electrical or computer engineering thesis. The following are the minimum English Language test scores (if required): ~~Test of English as a Foreign Language~~ 577 or York English Language Test 4. The Graduate Record Examination general test is strongly recommended, especially for applicants who did their work outside of Canada and/or the United States. **Degree Requirements** Candidates for the MASc degree in electrical and computer engineering must complete three graduate three-credit courses, a full-year, six-credit research project course (**Electrical Engineering & Computer Science 6400 6.0**) and write and successfully defend a master’s thesis. The **Electrical Engineering & Computer Science 6400 6.0** project must be distinct from course assignments and the MASc thesis. There is a breadth requirement for selected graduate courses. At least one course must be from two of the three fields of specialization: computer systems engineering, electrical engineering and interactive systems engineering. No more	resulting thesis should demonstrate the Candidates’ research ability in the research subject. **MSc Degree by Project** Candidates for the MSc degree must complete seven graduate three-credit courses and conduct a research project. The research project will have a more limited scope and/or degree of originality than a thesis. The project is under the general direction of a supervisor. A paper describing the project must be submitted and graded by the supervisory committee. **Time Requirements** Students are expected to complete all of their master’s degree requirements in no more than five terms (twenty months). For more details refer to the program’s supplemental calendar. **Master of Applied Science Program** **Admission Requirements** Graduates with an honors undergraduate degree or equivalent (typically a four-year program with full-time enrolment) from an accredited university in electrical or computer engineering, with at least a B+ average in the last two years of study, may be admitted as candidates for the Master of Applied Science program in electrical and computer engineering. In addition, those admitted must have completed the equivalent of a senior-level project course or thesis in electrical or computer engineering. Significant industrial or research experience in electrical or computer engineering coupled with an honors undergraduate degree program or equivalent from an accredited university will be considered equivalent to an undergraduate electrical or computer engineering thesis. The following are the minimum English language test scores (if required): TOEFL 577 (paper-based) or 90-91 (Internet-based), IELTS 7, or York English Language Test 4. The Graduate Record Examination general test is strongly recommended, especially for applicants who did their work outside of Canada and/or the United States. **Degree Requirements** Candidates for the MASc degree in electrical and computer engineering must complete three graduate three-credit courses, a full-year, six-credit research project course (**Electrical Engineering & Computer Science 6400 6.0**) and write and successfully defend a master’s thesis. The **Electrical Engineering & Computer Science 6400 6.0** project must be distinct from course assignments and the MASc thesis. There is a breadth requirement for selected graduate courses. At least one course must be from two of the three fields of specialization: computer systems engineering, electrical engineering and interactive systems engineering. No more

than one course integrated with an undergraduate course can be used to satisfy degree requirements.

A candidate must conduct approved thesis research that demonstrates their ability in the selected field of specialization under the general direction of a supervisor. Typically, the thesis includes a practical demonstration or implementation of the research work undertaken. For more details refer to the program's supplemental calendar.

Time Requirements

Students are expected to complete all of their master's degree requirements in no more than five terms (twenty months). For more details refer to the program's supplemental calendar.

Doctor of Philosophy Program

Admission Requirements

Applicants must have an MSc degree equivalent to the MSc Computer Science degree (thesis option) at York University. ~~The York MSc Computer Science degree is based upon course work and a defended thesis~~. A minimum average grade of B+ on all course work is required. Applications must include official copies of all academic transcripts, ~~a breadth statement,~~ an extended abstract/copy of the MSc thesis, three letters of reference and a one-page statement of purpose and previous experience. The statement of purpose should indicate the applicant's area(s) of interest ~~in computer science.~~ The following are the minimum English Language test scores (if required): ~~Test of English as a Foreign Language~~ 577 or York English Language Test 4. The Graduate Record Examination general test ~~and computer science subject test are~~ strongly recommended, especially for applicants who did their work outside of Canada and~~/or~~ the United States.

Degree Requirements

Candidates for the PhD degree must complete at least three three-credit graduate courses ~~to satisfy both breadth and depth requirements~~. No more than one-third of the course requirements can be integrated with undergraduate courses. Candidates must successfully complete a qualifying examination consisting of a written report on the candidate's field of interest and have an oral defense of the report. Candidates must present a dissertation proposal outlining the anticipated results of their dissertation. Candidates ~~are required to enrol in~~ either an industrial internship or a teaching practicum. ~~Finally,~~ candidates must conduct a significant body of original research under the supervision of a supervisory committee and successfully defend the resulting dissertation.

than one course integrated with an undergraduate course can be used to satisfy degree requirements.

A candidate must conduct approved thesis research that demonstrates their ability in the selected field of specialization under the general direction of a supervisor. Typically, the thesis includes a practical demonstration or implementation of the research work undertaken. For more details refer to the program's supplemental calendar.

Time Requirements

Students are expected to complete all of their master's degree requirements in no more than five terms (twenty months). For more details refer to the program's supplemental calendar.

Doctor of Philosophy Program

Admission Requirements

Applicants must have a Master's degree in Computer Engineering, Computer Science, Electrical Engineering, Software Engineering, or closely related field, which is equivalent to the MSc degree in Computer Science (thesis option) or the MASc degree in Electrical and Computer Engineering at York University. A minimum average grade of B+ on all course work is required. Applications must include official copies of all academic transcripts, an extended abstract/copy of the MSc or MASc thesis, three letters of reference and a one-page statement of purpose and previous experience. The statement of purpose should indicate the applicant's area(s) of interest. The following are the minimum English language test scores (if required): TOEFL 577 (paper-based) or 90-91 (Internet-based), IELTS 7, or York English Language Test 4. For applicants to the fields of Computer Engineering, Computer Science and Software Engineering, the Graduate Record Examination general test is strongly recommended, especially for applicants who did their work outside of Canada and the United States.

Degree Requirements

Candidates for the PhD degree must complete at least three three-credit graduate courses. No more than one-third of the course requirements can be integrated with undergraduate courses. Candidates must successfully complete a qualifying examination consisting of a written report on the candidate's field of interest and have an oral defence of the report. Candidates must present a dissertation proposal outlining the anticipated results of their dissertation. Each term candidates must attend departmental seminars. Each fall and winter term, candidates must attend one professional development seminar. Candidates must complete either an industrial internship or a teaching practicum. Candidates must conduct a significant body of original research under the supervision of a supervisory committee and successfully defend the resulting dissertation in their field of interest.

Time Requirements Students are expected to complete their requirements in no more than four years. ~~More detailed information is available in the program's supplemental calendar.~~	**Time Requirements** Students are expected to complete their requirements in no more than four years. Courses must be completed within three terms. The qualifying examination must be completed within five terms. The dissertation proposal must be completed within eight terms.

The PhD degree in Electrical Engineering and Computer Science extends the skills associated with the Master's degree and is awarded to students who have demonstrated the degree level expectations described in the following table. This table contains

- the degree level expectations as specified by the Ontario Universities Council on Quality Assurance,
- the description for each degree level expectation provided by the Ontario Universities Council on Quality Assurance,
- the program learning outcomes for each degree level expectation, and
- the degree requirements associated with those program learning outcomes.

1. Depth and breadth of knowledge
A thorough understanding of a substantial body of knowledge that is at the forefront of their academic discipline or area of professional practice including, where appropriate, relevant knowledge outside the field and/or discipline.
A. Review, analyze, assimilate and interpret a body of scientific literature in a number of areas, some of which are outside but pertinent to the research being undertaken. B. Identify gaps in the literature and opportunities for new research. C. Apply the techniques (mathematical, scientific, engineering, experimental) pertinent to the research being undertaken.
A. Courses, qualifying exam, industrial internship, teaching practicum, and dissertation. B. Dissertation proposal. C. Courses, qualifying exam, and dissertation.
2. Research and scholarship
a) The ability to conceptualize, design, and implement research for the generation of new knowledge, applications, or understanding at the forefront of the discipline, and to adjust the research design or methodology in the light of unforeseen problems; *b) The ability to make informed judgments on complex issues in specialist fields, sometimes requiring new methods; and* *c) The ability to produce original research, or other advanced scholarship, of a quality to satisfy peer review, and to merit publication.*
A. Identify novel and significant open research problems. (a) B. Design a research plan to tackle such a research problem. (a) C. Define and defend a research method that addresses such a research problem. (a) D. Discuss how applications of the research findings might impact the field. (a) E. Strategize how to address unforeseen outcomes of research by developing new methods within the field. (b) F. Formulate possible approaches to solving such a research problem and decide upon an appropriate approach by comparing them in relation to the issues relevant to the problem. (b) G. Analyze and evaluate ideas presented by other researchers. (c) H. Select appropriate methods for validating research results. (c)
A. Dissertation proposal and dissertation. B. Dissertation proposal and dissertation. C. Dissertation. D. Dissertation. E. Dissertation. F. Dissertation. G. Seminars and dissertation. H. Dissertation proposal and dissertation.
3. Level of application of knowledge
The capacity to: *a) undertake pure and/or applied research at an advanced level; and* *b) contribute to the development of academic or professional skills, techniques, tools, practices, ideas, theories, approaches, and/or materials.*
A. Conduct independent research appreciating the limitations of one's knowledge and seeking support and advice when warranted. (a) B. Identify and formulate research problems. (a) C. Solve research problems using established methods or new variations of those methods. (a) D. Develop academic or professional skills. (b)
A. Dissertation. B. Dissertation. C. Dissertation. D. Industrial internship and teaching practicum.

4. Professional capacity / autonomy
a) The qualities and transferable skills necessary for employment requiring the exercise of personal responsibility and largely autonomous initiative in complex situations; *b) The intellectual independence to be academically and professionally engaged and current;* *c) The ethical behaviour consistent with academic integrity and the use of appropriate guidelines and procedures for responsible conduct of research; and* *d) The ability to evaluate the broader implications of applying knowledge to particular contexts.*
A. Accept responsibility for one's research. (a) B. Evaluate individual progress towards meeting degree requirements and timelines. (a) C. Before engaging in academic debate, evaluate the relevant literature to remain up-to-date on findings in the field. (b) D. Evaluate how ethical, social, environmental, legal and regulatory influences may affect one's research. (c) E. Evaluate how non-compliance with relevant laws, regulations and intellectual property guidelines create risks in managing one's research. (c) F. Evaluate the implications of applying knowledge in an industrial or academic setting. (d)
A. Dissertation. B. Yearly progress report. C. Qualifying exam. D. Professional development workshops. E. Professional development workshops. F. Industrial internship and teaching practicum.
5. Level of communications skills
The ability to communicate complex and/or ambiguous ideas, issues and conclusions clearly and effectively.
A. Present material in a coherent and organized way, using an appropriate combination of media, to a variety of audiences. B. Listen carefully and gather feedback and opinions. C. Debate one's research position.
A. Courses, professional development workshops, qualifying exam, industrial internship, teaching practicum, dissertation proposal, and dissertation. B. Courses, professional development workshops, qualifying exam, dissertation proposal, teaching practicum, dissertation proposal, and dissertation. C. Dissertation.
6. Awareness of limits of knowledge
An appreciation of the limitations of one's own work and discipline, of the complexity of knowledge, and of the potential contributions of other interpretations, methods, and disciplines.
A. Revise the research methodology to account for limitations of the original approach. B. Recognize the importance of consultation with experts in the field. C. Develop a realistic appreciation of one's own strengths and weaknesses in research.
A. Courses, qualifying exam, dissertation proposal, and dissertation. B. Courses, qualifying exam, dissertation proposal, and dissertation. C. Courses, qualifying exam, dissertation proposal, and dissertation.

Note that no specific courses are associated with the program learning outcomes. The supervisor plays a crucial role in the selection of the courses. Students have to submit a course selection form at the start of their studies. Each term, students have to complete a progress report. Both are used to monitor that students achieve the program learning outcomes.

Graduate Fields
Definition and Proposal Template

Definition

In graduate programs, field refers to an area of specialization or concentration (in multi/interdisciplinary programs a clustered area of specialization) that is related to the demonstrable and collective strengths of the program's faculty. Institutions are not required to declare fields at either the master's or doctoral level. Institutions may wish, through an expedited approval process, to seek the endorsement of the Quality Council.

Graduate Field Proposal Guidelines

1. Indicate the name of the field being proposed and identify the parent program.

Field: Computer Engineering
Parent program: PhD in Electrical Engineering and Computer Science

2. Provide a description of the field (its intellectual focus, etc.) including the appropriateness and consistency of the field name with current usage in the discipline or area of study.

Computers control power generation systems, aircrafts, automobiles, homes and many other systems. Computer engineering is a discipline that focuses on the design, construction, implementation, and maintenance of software and hardware components of computers. Within the computer engineering field of the graduate program, students will significantly expand and deepen their understanding of the theory and practice of computer engineering in at least one of the following areas: computer systems engineering and interactive systems engineering. The former area concentrates on the architecture, design and evaluation of large-scale hardware and software systems. Examples include real-time systems, distributed computing and networking, and computer architectures. The latter area focuses on systems that interact intelligently with the user or the environment. Examples include mobile robotics, computer vision, natural-language speech recognition and synthesis, and advanced interface design.

3. Comment on the relationship of the admission requirements for the field to those of the parent program. If the same, describe the program admission requirements. If different, describe the field admission requirements, indicate how they are different from those of the parent program, and provide a rationale for the difference in relation to the focus and learning outcomes of the field.

The admission requirements for the field are the same as those of the parent program. These admission requirements are the following.

Applicants must have a Master's degree in Computer Engineering, Computer Science, Electrical Engineering, Software Engineering, or closely related field, which is equivalent to the MSc degree in Computer Science (thesis option) or the MASc degree in Electrical and Computer Engineering at York University. A minimum average grade of B+ on all course work is required. Applications must include official copies of all academic transcripts, an extended abstract/copy of the MSc or MASc thesis, three letters of reference and a one-page statement of purpose and previous experience. The statement of purpose should indicate the applicant's area(s) of interest. The following are the minimum English language test scores (if required): TOEFL 577 (paper-based) or 90-91 (Internet-based), IELTS 7, or

York English Language Test 4. The Graduate Record Examination general test and subject test are strongly recommended, especially for applicants who did their work outside of Canada and the United States.

4. Comment on the relationship of the curricular requirements for the field to those of the parent program. If the same, describe the program requirements. If different, describe the field requirements, indicate how they are different from those of the parent program, and provide a rationale for the difference in relation to the focus and learning outcomes of the field.

The curricular requirements for the field are the same as those of the parent program. These curricular requirements are the following.

Candidates for the PhD degree must complete at least three three-credit graduate courses. No more than one-third of the course requirements can be integrated with undergraduate courses. Candidates must successfully complete a qualifying examination consisting of a written report on the candidate's field of interest and have an oral defence of the report. Candidates must present a dissertation proposal outlining the anticipated results of their dissertation. Each term candidates must attend departmental seminars. Each fall and winter term, candidates must attend one professional development seminar. Candidates must complete either an industrial internship or a teaching practicum. Candidates must conduct a significant body of original research under the supervision of a supervisory committee and successfully defend the resulting dissertation.

The qualifying examination, the dissertation proposal and the dissertation must be on topics within the field of computer science. The fields of the student's supervisor must include computer engineering (see item 6).

5. Provide a list of courses that will be offered in support of the field. The list of courses must indicate the unit responsible for offering the course (including cross-lists and integrations, as appropriate), the course number, the credit value, the short course description, and whether or not it is an existing or new course. For existing courses, the frequency of offering should be noted. For new courses, full course proposals are required and should be included in the proposal as an appendix. (The list of courses may be organized to reflect the manner in which the courses count towards the program/field requirements, as appropriate; e.g. required versus optional; required from a list of specified courses; specific to certain concentrations, streams or fields within the program, etc.)

All the courses listed below are optional. They are all existing courses. The supervisor plays an important role in the course selection and will normally encourage students to take these courses.

Course: EECS 5101 3.0
Title: Advanced Data Structures
Short course description: The course discusses advanced data structures: heaps, balanced binary search trees, hashing tables, red-black trees, B-trees and their variants, structures for disjoint sets, binomial heaps, Fibonacci heaps, finger trees, persistent data structures, etc. When feasible, a mathematical analysis of these structures will be presented, with an emphasis on average case analysis and amortized analysis. If time permits, some lower bound techniques may be discussed, as well as NP-completeness proof techniques and approximation algorithms.
Number of offerings (in last five years): 4

Course: EECS 5311 3.0
Title: Logic Programming
Short course description: This course discusses core concepts and recent advances in the area of logic programming. Topics include logical foundations of logic programming systems, PROLOG as a logic programming system, constraints and dependencies, the closed-world assumption, and the problem of sound negation. Other topics include sequential versus parallel implementations, the problem of non-logical control primitives, optimizing backtracking, and applications to knowledge based programming.
Number of offerings (in last five years): 1

Course: EECS 5323 3.0
Title: Computer Vision
Short course description: This course introduces the basic concepts in computer vision. Primarily a survey of current computational methods, we begin by examining methods for measuring visual data (image based operators, edge detection, feature extraction), and low-level processes for feature aggregation (optic flow, segmentation, correspondence). Finally, we consider some issues in "high-level" vision by examining current high-level vision systems.
Number of offerings (in last five years): 5

Course: EECS 5324 3.0
Title: Introduction to Robotics
Short course description: This course introduces concepts in robotics. The course begins with a study of the mechanics of manipulators and robot platforms. Trajectory and course planning, environmental layout and sensing are discussed. Finally, high-level concerns are introduced. The need for real-time response and dynamic-scene analysis are covered, and recent development in robotics systems from an artificial intelligence viewpoint are discussed.
Number of offerings (in last five years): 5

Course: EECS 5326 3.0
Title: Artificial Intelligence
Short course description: This course will be an in-depth treatment of one or more specific topics within the field of artificial intelligence.
Number of offerings (in last five years): 1

Course: EECS 5327 3.0
Title: Introduction to Machine Learning and Pattern Recognition
Short course description: Machine learning is the study of algorithms that learn how to perform a task from prior experience. This course introduces the student to machine learning concepts and techniques applied to a pattern recognition problem in a diversity of application areas.
Number of offerings (in last five years): 3

Course: EECS 5351 3.0
Title: Human-Computer Interaction
Short course description: This course introduces the concepts and technology necessary to design, manage and implement interactive software. Students work in small groups and learn how to design user interfaces, how to realize them and how to evaluate the end result. Both design and evaluation are emphasized.
Number of offerings (in last five years): 3

Course: EECS 5421 3.0
Title: Operating System Design
Short course description: A modern operating system has four major components: process management, input/output, memory management, and the file system. This project-oriented course puts operating system principles into action and presents a practical approach to studying implementation aspects of operating systems. A series of projects are included for students to acquire direct experience in the design and construction of operating system components and have each interact correctly with the existing software. The programming environment is C/C++ under UNIX.
Number of offerings (in last five years): 5

Course: EECS 5441 3.0
Title: Real-Time Systems Theory
Short course description: Specification and verification techniques for real-time systems with many interacting components. Formal design of real-time systems using (a) programming languages with unambiguous semantics of time-related behavior and (b) scheduling algorithms.
Number of offerings (in last five years): 2

Course: EECS 5442 3.0
Title: Real-Time Systems Practice
Short course description: The course will focus on the technologies related to the design and implementation of real-time systems. Topics may include: typical real-time applications, process models of real-time systems, scheduling technologies in real-time systems, design and implementation of real-time systems software, real-time systems hardware, real-time operating systems, real-time programming languages, and inspection and verification methods for real-time systems
Number of offerings (in last five years): 4

Course: EECS 5443 3.0
Title: Mobile User Interfaces
Short course description: This course teaches the design and implementation of user interfaces for touchscreen phones and tablet computers. Students develop user interfaces that include touch, multi-touch, vibration, device motion, position, and orientation, environment sensing, and video and audio capture. Lab exercises emphasise these topics in a practical manner.
Number of offerings (in last five years): 3

Course: EECS 5501 3.0
Title: Computer Architecture
Short course description: This course presents the core concepts of computer architecture and design ideas embodied in many machines and emphasizes a quantitative approach to cost/performance tradeoffs. This course concentrates on uniprocessor systems. A few machines are studies to illustrate how these concepts are implemented; how various tradeoffs that exit among design choices are treated; and how good designs make efficient use of technology. Future trends in computer architecture are also discussed.
Number of offerings (in last five years): 5

Course: EECS 6111 3.0
Title: Advanced Algorithm Design and Analysis
Short course description: This is an advanced theoretical computer science course directed at non-theory students with the standard undergraduate background. The goal is to survey the key theory topics that every computer science graduate student should know. In about two weeks for each selected topic, we

will gain insights into the basics and study one or two example in depth. These might include: a deepening of student's knowledge of key algorithmic techniques, randomized algorithms, NP-completeness, approximation algorithms, linear programming, distributed systems, computability, concurrency theory, cryptography, structural complexity, data structures, and quantum algorithms.
Number of offerings (in last five years): 3

Course: EECS 6117 3.0
Title: Distributed Computing
Short course description: Can a given problem be solved in a distributed system? If so, how efficiently? This course investigates how the answers to these questions depend on aspects of the underlying distributed system including synchrony, fault-tolerance and the means of communication between processes. Topics include models of distributed systems, mutual exclusion, agreement problems, lower bounds and consensus hierarchy.
Number of offerings (in last five years): 3

Course: EECS 6222 3.0
Title: Coding and Information Theory
Short course description: This course introduces students to the fundamentals of information theory, as well as methods for achieving information-theoretic results using source codes and channel codes. Students will learn Shannon's source coding and channel coding theorems, as well as the mathematical machinery required to prove these and other information theoretic results. Students will also be exposed to source coding techniques, as well as channel coding techniques for state-of-the-art systems. Advanced topics such as multiterminal (Slepian-Wolf) source coding and rateless codes will also be covered, time permitting.
Number of offerings (in last five years): 1

Course: EECS 6323 3.0
Title: Advanced Topics in Computer Vision
Short course description: An advanced topics course in computer vision which covers selected topics in greater depth. Topics covered will vary from year to year depending on the interests of the class and instructor. Possible topics include: stereo vision, visual motion, computer audition, fast image processing algorithms, vision based mobile robots and active vision sensors, and object recognition.
Number of offerings (in last five years): 3

Course: EECS 6324 3.0
Title: From Control to Actuators
Short course description: A "robot building course", this course will follow the issues involved in building a robot or robotic system from control to actuators. This includes microcomputer control, actuator design, high-level software models, and sensor inputs.
Number of offerings (in last five years): 1

Course: EECS 6327 3.0
Title: Probabilistic Models & Machine Learning
Short course description: Intelligent systems must make effective judgements in the face of uncertainty. This requires probabilistic models to represent complex relationships between random variables (learning) as well as algorithms that produce good estimates and decisions based on these models (inference). This course explores both probabilistic learning and inference, in a range of application areas.
Number of offerings (in last five years): 0

Course: EECS 6328 3.0
Title: Speech and Language Processing
Short course description: Introducing the latest technologies in speech and language processing, including speech and recognition and understanding, key-word spotting, spoken language processing, speaker identification and verification, statistical machine translation, information retrieval, and other interesting topics.
Number of offerings (in last five years): 1

Course: EECS 6329 3.0
Title: Advanced Human-Computer Interaction
Short course description: This course examines advanced concepts and technologies for human-computer interaction. Students will learn about advanced input and output devices (e.g. for mobile computing and/or virtual reality), about advanced design methods, how to implement effective interfaces, and how to perform rapid, effective iterative user tests.
Number of offerings (in last five years): 4

Course: EECS 6331 3.0
Title: Advanced Image Synthesis
Short course description: This course concentrates on raster algorithms and image synthesis. Some of the topics may include visible surface algorithms, modeling, shading, global illumination, anti-aliasing, texture mapping and animation.
Number of offerings (in last five years): 1

Course: EECS 6332 3.0
Title: Statistical Visual Motion Analysis
Short course description: A seminar course that examines statistical approaches to visual motion analysis, including 3-D structure and motion estimation, optical flow, segmentation and tracking using tools like maximum likelihood estimation, maximum a posteriori, least squares and expectation maximization.
Number of offerings (in last five years): 1

Course: EECS 6335 3.0
Title: Topics in Virtual Reality
Short course description: This course considers how to present to a user a compelling illusion of being in an alternate (virtual) reality. It considers how humans perceive visual, audio, haptic and other perceptual inputs, and how technology can be used to stimulate these senses appropriately to simulate some virtual environment.
Number of offerings (in last five years): 1

Course: EECS 6337 3.0
Title: 3D User Interfaces
Short course description: The course introduces the ways to interact with computers in a three dimensional (3D) environment, where the environment is either fully virtual or represents a mixture of real and virtual. It covers topics ranging from the hardware necessary to interface with virtual worlds, over techniques for interacting with 3D environments, to design and evaluation of 3D user interfaces.
Number of offerings (in last five years): 1

Course: EECS 6338 3.0
Title: Computer-Aided Interventions
Short course description: This course introduces students to the fundamentals of computer-aided intervention (CAI) in medicine. The use of computing and computing technology before, during, and after intervention are examined. Clinical applications are also discussed.
Number of offerings (in last five years): 1

Course: EECS 6339 3.0
Title: Introduction to Computational Linguistics
Short course description: Introduction to Computational Linguistics explores computational techniques for understanding, translating and producing natural language, and investigates the structure and meaning of sentences and connected discourse. Some applications are discussed, e.g., question answering, machine translation, text classification, information extraction and so on.
Number of offerings (in last five years): 3

Course: EECS 6340 3.0
Title: Embodied Intelligence
Short course description: This course is intended as a follow-on from a first course on artificial intelligence. Whereas such first courses focus on the important foundations of AI, such a knowledge representation or reasoning, this course will examine how these separate foundational elements can be integrated into real systems. This will be accomplished by detailing some general overall concepts that form the basis of intelligent systems in the real world, and then presenting a number of in-depth cases studies of a variety of systems from several applications domains. The embodiment of intelligence may be in a physical system (such as a robot) or a software system (such as in game-playing) but in both cases, the goal is to interact with, and solve a problem in, the real world.
Number of offerings (in last five years): 3

Course: EECS 6390A 3.0
Title: Special Topics: Knowledge Representation
Short course description: This course examines some of the techniques used to represent knowledge in artificial intelligence, and the associated methods of automated reasoning. The emphasis will be on the compromises involved in providing a useful but tractable representation and reasoning service to a knowledge-based system. The topics may include: formal models of knowledge and belief, systems of limited reasoning, languages of limited expressive power, defaults and exceptions, meta-level representation and reasoning, reasoning about action, and theories of rational agency.
Number of offerings (in last five years): 2

Course: EECS 6390B 3.0
Title: Scheduling in Hard Real-Time Systems
Short course description: This course discusses concepts and methods for satisfying timing constraints in large, complex hard-real-time systems. Topics include: characteristics of hard-real-time systems, timing constraints, periodic and asynchronous processes, run-time and pre-run-time scheduling, cyclic executives, priority scheduling, preemptive and non-preemptive scheduling, synchronization, schedulability analysis, resource management, and real-time programming language constructs.
Number of offerings (in last five years): 1

Course: EECS 6390D 3.0
Title: Computational Models of Visual Perception
Short course description: This course examines the problem of developing rigorous computational models for visual processing. Computational strategies may draw upon techniques in statistical inference, signal processing, optimization theory, graph theory and distributed computation.
Number of offerings (in last five years): 1

Course: EECS 6390E 3.0
Title: Special Topics in AI and Interactive Systems II: Reasoning about Interaction between Rational Agents
Short course description: Not available.
Number of offerings (in last five years): 0

Course: EECS 6411 3.0
Title: Programming Logic for Complex Systems
Short course description: This course covers program verification methods for a class of programs, commonly referred to as reactive programs. Reactive programs typically never terminate and are run in order to maintain some interaction with the environment. An adequate description of reactive systems must refer not only to initial and final states, but also to the ongoing behavior as a (possibly infinite) sequence of states and events. The purpose of this course is to investigate the use of logical calculi for the specification, design and verification of reactive systems. Topics include: modeling of discrete event systems, semantics of real-time languages, logical and discrete calculi (e.g. temporal logic) for specifying and verifying safety, liveness, deadlock, priority and fairness properties of reactive programs, and prolog tools for automating verification.
Number of offerings (in last five years): 0

Course: EECS 6412 3.0
Title: Data Mining
Short course description: This course introduces fundamental concepts of data mining. It presents various data mining technologies, algorithms and applications. Topics include association rule mining, classification models, sequential pattern mining and clustering.
Number of offerings (in last five years): 3

Course: EECS 6421 3.0
Title: Advanced Data Systems
Short course description: This course provides and introduction to and an in-depth study on several new developments in database systems and intelligent information systems. Topics include: Internet databases, data warehousing and OLAP, object-relational, object-oriented, and deductive databases.
Number of offerings (in last five years): 3

Course: EECS 6431 3.0
Title: Software Re-Engineering
Short course description: Industrial software systems are usually large and complex, while knowledge of their structure is either lost or inadequately documented. This course presents techniques that aid the comprehension and design recovery of large software systems.
Number of offerings (in last five years): 1

Course: EECS 6432 3.0
Title: Adaptive Software Systems
Short course description: Adaptive software systems are software systems that change their behaviour and structure to cope with changes in environment conditions or in user requirements. Adaptation includes self-optimization, self-protection, self-configuration and self-healing. This course covers basic and advanced concepts in engineering adaptive systems and has a special focus on self-optimization. It introduces the students to the mathematical foundations of adaptive systems including performance models, estimators for performance models, feedback loop architectures and strategies, and optimization.
Number of offerings (in last five years): 0

Course: EECS 6441
Title: Methods for Large-Scale Software
Short course description: This course studies the application of mathematical methods to the construction of large-scale software systems. It considers issues relevant to large-scale design and the application of mathematics. It involves a large-scale software project in which industrial-strength tools are applied.
Number of offerings (in last five years): 1

Course: EECS 6444 3.0
Title: Mining Software Engineering Data to Support the Development, Testing and Maintenance of Large Scale Software Systems
Short course description: Software engineering data (such as source code repositories, execution logs, performance counters, developer mailing lists and bug databases) contains a wealth of information about a project's status and history. Applying data mining techniques on such data, researchers can gain empirically based understanding of software development practices, and practitioners can better manage, maintain and evolve complex software projects.
Number of offerings (in last five years): 2

Course: EECS 6490A
Title: Concurrent Object Oriented Languages
Short course description: In this course, we focus on concurrent programming in the object oriented language Java. The course consists of three main parts. In the first part, we discuss concurrent programming in general. In the second part, we concentrate on writing concurrent programs in Java. In the third and final part, we look at techniques and tools to verify concurrent Java programs.
Number of offerings (in last five years): 3

Course: EECS 6490B 3.0
Title: Issues in Information Integration
Short course description: This course explores the challenges and research issues that arise in scaling current information systems technology to a widely-distributed heterogeneous database environment. The focus of the course is on using semantic information to integrate information sources, optimize query processing and provide cooperative response to a user in such systems. Topics to be covered in this course: heterogeneous database systems, management of uncertain (disjunctive) information, integrating relational and object-oriented database models, dynamic query processing, semantic query caching, semantic query optimization, and cooperative answering systems.
Number of offerings (in last five years): 2

Course: EECS 6503 3.0
Title: Cyber-Physical Systems: Modelling, Specification, Synthesis and Verification
Short course description: This course presents concepts and techniques for cyber-physical systems design. It covers models of computation for embedded systems, system level design, specification, synthesis and verification of embedded and cyber-physical systems. The instructor may choose to cover other topics such as: architecture for embedded systems, the software hardware co-design, design space exploration, performance evaluation of embedded systems and/or security. Design methodologies and tools are used for case studies and design projects.
Number of offerings (in last five years): 0

Course: EECS 6505 3.0
Title: Physical and Systems Design Issues in Application Specific Integrated Circuits
Short course description: Designers of modern very large scale integrated (VLSI) systems face the conflicting pressure of realizing application-specific integrated circuits (ASICs) with increasingly complex and varied functionality while subject to more demanding physical electronic constraints. This design-centric course addresses critical issues in both of these aspects by giving students a hands-on opportunity to architect VLSI systems using modern CAD tools spanning both physical and systems design. Topics include: high-speed/low-power circuit analysis and design strategies, interconnect, clock and power distribution, timing strategies, floor-planning and layout, synthesis and verification.
Number of offerings (in last five years): 0

Course: EECS 6590A 3.0
Title: Special Topics: High-Performance Computer Networks
Short course description: This course focuses on high performance computer networks. It presents a comprehensive study of modern high speed communication networks that is capable of providing data, voice, and video services. It also covers mobile and wireless communication networks.
Number of offerings (in last five years): 5

6. Comment on the expertise of the faculty who will actively support/participate the field and provide a Table of Faculty by field, as follows:

All faculty members mentioned in the table below conduct research in the field and supervise graduate students in the field.

Faculty Member & Rank	Home Unit	Primary Field	Category
Mokhtar Aboelaze, Associate professor	EECS	Computer Engineering	Associate member
Robert Allison, Professor	EECS	Computer Engineering	Full member
Amir Asif, Professor	EECS	Computer Engineering	Full member
James Elder, Professor	EECS	Computer Engineering	Full member
Hui Jiang, Professor	EECS	Computer Engineering	Full member
Xiaodan Zhu	EECS	Computer Engineering	Adjunct member

Note: Up-to-date CVs of faculty who will actively participate in delivering the graduate program must be included as an appendix.

7. Comment on the projected in-take into the field, including the anticipated implementation date (i.e. year and term of initial in-take), and indicate if the projected in-take is within or in addition to the existing enrolment targets for the parent program.

The in-take into the field is projected to be at least three students per year. This projected in-take is within the existing enrolment targets for the parent program. The anticipated implementation date is Fall 2017.

The field highlights Computer Engineering within the broad spectrum covered by the parent program (from Electrical Engineering to Computer Science) and, therefore, will be helpful for recruitment of new students. Within the highly regulated discipline of engineering, recognition of the type of engineering is very valuable for graduates when seeking employment.

8. Comment on the impact of the field on the parent program, focusing on the extent of diversion of faculty from existing graduate courses and/or supervision, as well as the capacity of the program to absorb any anticipated additional enrolment.

The introduction of this field will *not* impact the parent program. The course requirements are not changed. There is sufficient capacity for supervision.

9. Support statements

- **from the relevant Dean(s)/Principal, with respect to the adequacy of existing resources necessary to support the new field, as well as the commitment to any plans for new/additional resources necessary to implement and/or sustain the new field**
- **from the relevant Faculties/units/programs confirming consultation on/support for the new program, as appropriate**
- **from professional associations, government agencies or policy bodies with respect to the need/demand for the proposed program, as appropriate**

Graduate Fields
Definition and Proposal Template

Definition

In graduate programs, field refers to an area of specialization or concentration (in multi/interdisciplinary programs a clustered area of specialization) that is related to the demonstrable and collective strengths of the program's faculty. Institutions are not required to declare fields at either the master's or doctoral level. Institutions may wish, through an expedited approval process, to seek the endorsement of the Quality Council.

Graduate Field Proposal Guidelines

1. Indicate the name of the field being proposed and identify the parent program.

Field: Computer Science
Parent program: PhD in Electrical Engineering and Computer Science

2. Provide a description of the field (its intellectual focus, etc.) including the appropriateness and consistency of the field name with current usage in the discipline or area of study.

Computers play a pivotal role in daily life. Computer science studies the scientific and practical approach to computation and its applications. Within the computer science field of the graduate program, students will significantly expand and deepen their understanding of the theory and practice of computer science in at least one of the following areas: theoretical computer science, intelligent and interactive systems, and systems software and hardware. The area of theoretical computer science concentrates on algorithms, complexity, computability, logic, parallel, concurrent and distributed computing. The focus of the area of intelligent and interactive systems includes artificial intelligence, computer vision, human-computer interaction, information retrieval, robotics, speech and virtual reality. The area of systems software and hardware encompasses data mining, databases, networks, and signal processing.

3. Comment on the relationship of the admission requirements for the field to those of the parent program. If the same, describe the program admission requirements. If different, describe the field admission requirements, indicate how they are different from those of the parent program, and provide a rationale for the difference in relation to the focus and learning outcomes of the field.

The admission requirements for the field are the same as those of the parent program. These admission requirements are the following.

Applicants must have a Master's degree in Computer Engineering, Computer Science, Electrical Engineering, Software Engineering, or closely related field, which is equivalent to the MSc degree in Computer Science (thesis option) or the MASc degree in Electrical and Computer Engineering at York University. A minimum average grade of B+ on all course work is required. Applications must include official copies of all academic transcripts, an extended abstract/copy of the MSc or MASc thesis, three letters of reference and a one-page statement of purpose and previous experience. The statement of purpose should indicate the applicant's area(s) of interest. The following are the minimum English language test scores (if required): TOEFL 577 (paper-based) or 90-91 (Internet-based), IELTS 7, or

York English Language Test 4. The Graduate Record Examination general test and subject test are strongly recommended, especially for applicants who did their work outside of Canada and the United States.

4. Comment on the relationship of the curricular requirements for the field to those of the parent program. If the same, describe the program requirements. If different, describe the field requirements, indicate how they are different from those of the parent program, and provide a rationale for the difference in relation to the focus and learning outcomes of the field.

The curricular requirements for the field are the same as those of the parent program. These curricular requirements are the following.

Candidates for the PhD degree must complete at least three three-credit graduate courses. No more than one-third of the course requirements can be integrated with undergraduate courses. Candidates must successfully complete a qualifying examination consisting of a written report on the candidate's field of interest and have an oral defence of the report. Candidates must present a dissertation proposal outlining the anticipated results of their dissertation. Each term candidates must attend departmental seminars. Each fall and winter term, candidates must attend one professional development seminar. Candidates must complete either an industrial internship or a teaching practicum. Candidates must conduct a significant body of original research under the supervision of a supervisory committee and successfully defend the resulting dissertation.

The qualifying examination, the dissertation proposal and the dissertation must be on topics within the field of computer science. The fields of the student's supervisor must include computer science (see item 6).

5. Provide a list of courses that will be offered in support of the field. The list of courses must indicate the unit responsible for offering the course (including cross-lists and integrations, as appropriate), the course number, the credit value, the short course description, and whether or not it is an existing or new course. For existing courses, the frequency of offering should be noted. For new courses, full course proposals are required and should be included in the proposal as an appendix. (The list of courses may be organized to reflect the manner in which the courses count towards the program/field requirements, as appropriate; e.g. required versus optional; required from a list of specified courses; specific to certain concentrations, streams or fields within the program, etc.)

All the courses listed below are optional. They are all existing courses. The supervisor plays an important role in the course selection and will normally encourage students to take these courses.

Course: EECS 5101 3.0
Title: Advanced Data Structures
Short course description: The course discusses advanced data structures: heaps, balanced binary search trees, hashing tables, red-black trees, B-trees and their variants, structures for disjoint sets, binomial heaps, Fibonacci heaps, finger trees, persistent data structures, etc. When feasible, a mathematical analysis of these structures will be presented, with an emphasis on average case analysis and amortized analysis. If time permits, some lower bound techniques may be discussed, as well as NP-completeness proof techniques and approximation algorithms.
Number of offerings (in last five years): 4

Course: EECS 5111 3.0
Title: Automata, Computability and Complexity
Short course description: This course is intended to give students a detailed understanding of the basic concepts of abstract machine structure, information flow, computability, and complexity. The emphasis will be on appreciating the significance of these ideas and the formal techniques used to establish their properties. Topics chosen for study include: models of finite and infinite automata, the limits to computation, and the measurement of the intrinsic difficulty of computational problems.
Number of offerings (in last five years): 5

Course: EECS 5290 3.0
Title: Scientific Computing: Bioinformatics
Short course description: Bioinformatics deals with the computation of biological information. This course presents an introduction to the basic concepts of molecular genetics; concepts and algorithms for sequence comparison; examples of algorithms for protein structure prediction; and biological data mining.
Number of offerings (in last five years): 1

Course: EECS 5311 3.0
Title: Logic Programming
Short course description: This course discusses core concepts and recent advances in the area of logic programming. Topics include logical foundations of logic programming systems, PROLOG as a logic programming system, constraints and dependencies, the closed-world assumption, and the problem of sound negation. Other topics include sequential versus parallel implementations, the problem of non-logical control primitives, optimizing backtracking, and applications to knowledge based programming.
Number of offerings (in last five years): 1

Course: EECS 5323 3.0
Title: Computer Vision
Short course description: This course introduces the basic concepts in computer vision. Primarily a survey of current computational methods, we begin by examining methods for measuring visual data (image based operators, edge detection, feature extraction), and low-level processes for feature aggregation (optic flow, segmentation, correspondence). Finally, we consider some issues in "high-level" vision by examining current high-level vision systems.
Number of offerings (in last five years): 5

Course: EECS 5324 3.0
Title: Introduction to Robotics
Short course description: This course introduces concepts in robotics. The course begins with a study of the mechanics of manipulators and robot platforms. Trajectory and course planning, environmental layout and sensing are discussed. Finally, high-level concerns are introduced. The need for real-time response and dynamic-scene analysis are covered, and recent development in robotics systems from an artificial intelligence viewpoint are discussed.
Number of offerings (in last five years): 5

Course: EECS 5326 3.0
Title: Artificial Intelligence
Short course description: This course will be an in-depth treatment of one or more specific topics within the field of artificial intelligence.
Number of offerings (in last five years): 1

Course: EECS 5327 3.0
Title: Introduction to Machine Learning and Pattern Recognition
Short course description: Machine learning is the study of algorithms that learn how to perform a task from prior experience. This course introduces the student to machine learning concepts and techniques applied to a pattern recognition problem in a diversity of application areas.
Number of offerings (in last five years): 3

Course: EECS 5331 3.0
Title: Advanced Topics in 3D Computer Graphics
Short course description: This course introduces advanced 3D computer graphics algorithms. Topics may include direct programming of graphics hardware via pixel and vertex shaders, real-time rendering, global illumination algorithms, advanced texture mapping and anti-aliasing, data visualization.
Number of offerings (in last five years): 5

Course: EECS 5351 3.0
Title: Human-Computer Interaction
Short course description: This course introduces the concepts and technology necessary to design, manage and implement interactive software. Students work in small groups and learn how to design user interfaces, how to realize them and how to evaluate the end result. Both design and evaluation are emphasized.
Number of offerings (in last five years): 3

Course: EECS 5421 3.0
Title: Operating System Design
Short course description: A modern operating system has four major components: process management, input/output, memory management, and the file system. This project-oriented course puts operating system principles into action and presents a practical approach to studying implementation aspects of operating systems. A series of projects are included for students to acquire direct experience in the design and construction of operating system components and have each interact correctly with the existing software. The programming environment is C/C++ under UNIX.
Number of offerings (in last five years): 5

Course: EECS 5431 3.0
Title: Mobile Communications
Short course description: This course provides an overview of the latest technology, developments and trends in wireless mobile communications, and addresses the impact of wireless transmission and user mobility on the design and management of wireless mobile systems.
Number of offerings (in last five years): 3

Course: EECS 5441 3.0
Title: Real-Time Systems Theory
Short course description: Specification and verification techniques for real-time systems with many interacting components. Formal design of real-time systems using (a) programming languages with unambiguous semantics of time-related behavior and (b) scheduling algorithms.
Number of offerings (in last five years): 2

Course: EECS 5442 3.0
Title: Real-Time Systems Practice
Short course description: The course will focus on the technologies related to the design and implementation of real-time systems. Topics may include: typical real-time applications, process models of real-time systems, scheduling technologies in real-time systems, design and implementation of real-time systems software, real-time systems hardware, real-time operating systems, real-time programming languages, and inspection and verification methods for real-time systems
Number of offerings (in last five years): 4

Course: EECS 5443 3.0
Title: Mobile User Interfaces
Short course description: This course teaches the design and implementation of user interfaces for touchscreen phones and tablet computers. Students develop user interfaces that include touch, multi-touch, vibration, device motion, position, and orientation, environment sensing, and video and audio capture. Lab exercises emphasise these topics in a practical manner.
Number of offerings (in last five years): 3

Course: EECS 6111 3.0
Title: Advanced Algorithm Design and Analysis
Short course description: This is an advanced theoretical computer science course directed at non-theory students with the standard undergraduate background. The goal is to survey the key theory topics that every computer science graduate student should know. In about two weeks for each selected topic, we will gain insights into the basics and study one or two example in depth. These might include: a deepening of student's knowledge of key algorithmic techniques, randomized algorithms, NP-completeness, approximation algorithms, linear programming, distributed systems, computability, concurrency theory, cryptography, structural complexity, data structures, and quantum algorithms.
Number of offerings (in last five years): 3

Course: EECS 6114 3.0
Title: Computational Geometry
Short course description: The purpose of this course is to give a state of the art introduction to computational geometry so as to be beneficial both for students interested in theory and for students interested in its applied fields such as computer-aided design, computer graphics, and robotics, etc. The course will also use some program animation packages. Several techniques important to computational geometry will be emphasized: divide-&-conquer, amortization, multi-dimensional search, space sweep, duality and randomization. Topics include: convex hulls, Voronoi and Delaunay diagrams, arrangements, hidden surface removal, polygon triangulation, art gallery theorems, shortest paths, and lower-bounds.
Number of offerings (in last five years): 0

Course: EECS 6115 3.0
Title: Computational Complexity
Short course description: This course is an introduction to computational complexity, focusing on the computational resource requirements (such as time and space) which are required for important computational tasks. Topics include: the general theory of complexity classes, and specific complexity classes of interest such as problems which can be solved in polynomial time and the class NP; model-theoretic (Turing machine and circuit) and logical (expressibility) characterizations of complexity classes; relations between complexity classes, such as the cost of simulating nondeterminism by determinism; complexity hierarchies, reductions and NP-completeness; the polynomial space hierarchy,

intractability. There will also be included a selection of other topics from the areas of cryptography and protocols, axiomatic complexity theory, randomized complexity, the approximability of optimization problems, circuit complexity, parallel complexity and the complexity of logical theories, and other current research topics in computational complexity theory.
Number of offerings (in last five years): 4

Course: EECS 6116 3.0
Title: Advanced Computational Complexity
Short course description: This course is an advanced course on computational complexity. Topics covered will include complexity classes, models of computation, lower bounds, parallel complexity, randomized algorithms, cryptography, along with techniques from combinatorics, probability theory, and logic.
Number of offerings (in last five years): 0

Course: EECS 6117 3.0
Title: Distributed Computing
Short course description: Can a given problem be solved in a distributed system? If so, how efficiently? This course investigates how the answers to these questions depend on aspects of the underlying distributed system including synchrony, fault-tolerance and the means of communication between processes. Topics include models of distributed systems, mutual exclusion, agreement problems, lower bounds and consensus hierarchy.
Number of offerings (in last five years): 3

Course: EECS 6118 3.0
Title: Optimization
Short course description: This course investigates the algorithmic and computational complexity aspects of combinatorial optimization problems. Optimization problem areas include linear, non-linear, convex, integer, and semidefinite programming, as well as their application to specific areas such as network flow, matching and various graph optimization problems.
Number of offerings (in last five years): 3

Course: EECS 6323 3.0
Title: Advanced Topics in Computer Vision
Short course description: An advanced topics course in computer vision which covers selected topics in greater depth. Topics covered will vary from year to year depending on the interests of the class and instructor. Possible topics include: stereo vision, visual motion, computer audition, fast image processing algorithms, vision based mobile robots and active vision sensors, and object recognition.
Number of offerings (in last five years): 3

Course: EECS 6324 3.0
Title: From Control to Actuators
Short course description: A "robot building course", this course will follow the issues involved in building a robot or robotic system from control to actuators. This includes microcomputer control, actuator design, high-level software models, and sensor inputs.
Number of offerings (in last five years): 1

Course: EECS 6326 3.0
Title: Principles of Human Perception and Performance in Human-Computer Interactions
Short course description: This course considers the role of human perception in human-computer interaction particularly computer generated graphics/sound and immersive virtual reality. Fundamental findings from sensory physiology and perceptual psychophysics are presented in the context of interface and display design.
Number of offerings (in last five years): 2

Course: EECS 6327 3.0
Title: Probabilistic Models & Machine Learning
Short course description: Intelligent systems must make effective judgements in the face of uncertainty. This requires probabilistic models to represent complex relationships between random variables (learning) as well as algorithms that produce good estimates and decisions based on these models (inference). This course explores both probabilistic learning and inference, in a range of application areas.
Number of offerings (in last five years): 0

Course: EECS 6328 3.0
Title: Speech and Language Processing
Short course description: Introducing the latest technologies in speech and language processing, including speech and recognition and understanding, key-word spotting, spoken language processing, speaker identification and verification, statistical machine translation, information retrieval, and other interesting topics.
Number of offerings (in last five years): 1

Course: EECS 6329 3.0
Title: Advanced Human-Computer Interaction
Short course description: This course examines advanced concepts and technologies for human-computer interaction. Students will learn about advanced input and output devices (e.g. for mobile computing and/or virtual reality), about advanced design methods, how to implement effective interfaces, and how to perform rapid, effective iterative user tests.
Number of offerings (in last five years): 4

Course: EECS 6330 3.0
Title: Critical Technical Practise: Computer Accessibility and Assistive Technology
Short course description: Many interactive systems strive to afford the same mechanisms to human users that are used in face-to-face conversation. This course examines the formal models and computational techniques that concern the pragmatics of language use that such systems employ.
Number of offerings (in last five years): 1

Course: EECS 6331 3.0
Title: Advanced Image Synthesis
Short course description: This course concentrates on raster algorithms and image synthesis. Some of the topics may include visible surface algorithms, modeling, shading, global illumination, anti-aliasing, texture mapping and animation.
Number of offerings (in last five years): 1

Course: EECS 6332 3.0
Title: Statistical Visual Motion Analysis
Short course description: A seminar course that examines statistical approaches to visual motion analysis, including 3-D structure and motion estimation, optical flow, segmentation and tracking using tools like maximum likelihood estimation, maximum a posteriori, least squares and expectation maximization.
Number of offerings (in last five years): 1

Course: EECS 6333 3.0
Title: Multiple View Image Understanding
Short course description: This course considers how multiple images of a scene, as captured by multiple stationary cameras, single moving cameras or their combination, can be used to recover information about the viewed scene (e.g., three-dimensional layout, camera and/or scene movement). Theoretical and practical issues of calibration, correspondence/matching and interpretation will be considered.
Number of offerings (in last five years): 1

Course: EECS 6335 3.0
Title: Topics in Virtual Reality
Short course description: This course considers how to present to a user a compelling illusion of being in an alternate (virtual) reality. It considers how humans perceive visual, audio, haptic and other perceptual inputs, and how technology can be used to stimulate these senses appropriately to simulate some virtual environment.
Number of offerings (in last five years): 1

Course: EECS 6337 3.0
Title: 3D User Interfaces
Short course description: The course introduces the ways to interact with computers in a three dimensional (3D) environment, where the environment is either fully virtual or represents a mixture of real and virtual. It covers topics ranging from the hardware necessary to interface with virtual worlds, over techniques for interacting with 3D environments, to design and evaluation of 3D user interfaces.
Number of offerings (in last five years): 1

Course: EECS 6339 3.0
Title: Introduction to Computational Linguistics
Short course description: Introduction to Computational Linguistics explores computational techniques for understanding, translating and producing natural language, and investigates the structure and meaning of sentences and connected discourse. Some applications are discussed, e.g., question answering, machine translation, text classification, information extraction and so on.
Number of offerings (in last five years): 3

Course: EECS 6340 3.0
Title: Embodied Intelligence
Short course description: This course is intended as a follow-on from a first course on artificial intelligence. Whereas such first courses focus on the important foundations of AI, such a knowledge representation or reasoning, this course will examine how these separate foundational elements can be integrated into real systems. This will be accomplished by detailing some general overall concepts that form the basis of intelligent systems in the real world, and then presenting a number of in-depth cases studies of a variety of systems from several applications domains. The embodiment of intelligence may

be in a physical system (such as a robot) or a software system (such as in game-playing) but in both cases, the goal is to interact with, and solve a problem in, the real world.
Number of offerings (in last five years): 3

Course: EECS 6390A 3.0
Title: Special Topics: Knowledge Representation
Short course description: This course examines some of the techniques used to represent knowledge in artificial intelligence, and the associated methods of automated reasoning. The emphasis will be on the compromises involved in providing a useful but tractable representation and reasoning service to a knowledge-based system. The topics may include: formal models of knowledge and belief, systems of limited reasoning, languages of limited expressive power, defaults and exceptions, meta-level representation and reasoning, reasoning about action, and theories of rational agency.
Number of offerings (in last five years): 2

Course: EECS 6390B 3.0
Title: Scheduling in Hard Real-Time Systems
Short course description: This course discusses concepts and methods for satisfying timing constraints in large, complex hard-real-time systems. Topics include: characteristics of hard-real-time systems, timing constraints, periodic and asynchronous processes, run-time and pre-run-time scheduling, cyclic executives, priority scheduling, preemptive and non-preemptive scheduling, synchronization, schedulability analysis, resource management, and real-time programming language constructs.
Number of offerings (in last five years): 1

Course: EECS 6390D 3.0
Title: Computational Models of Visual Perception
Short course description: This course examines the problem of developing rigorous computational models for visual processing. Computational strategies may draw upon techniques in statistical inference, signal processing, optimization theory, graph theory and distributed computation.
Number of offerings (in last five years): 1

Course: EECS 6390E 3.0
Title: Special Topics in AI and Interactive Systems II: Reasoning about Interaction between Rational Agents
Short course description: Not available.
Number of offerings (in last five years): 0

Course: EECS 6411 3.0
Title: Programming Logic for Complex Systems
Short course description: This course covers program verification methods for a class of programs, commonly referred to as reactive programs. Reactive programs typically never terminate and are run in order to maintain some interaction with the environment. An adequate description of reactive systems must refer not only to initial and final states, but also to the ongoing behavior as a (possibly infinite) sequence of states and events. The purpose of this course is to investigate the use of logical calculi for the specification, design and verification of reactive systems. Topics include: modeling of discrete event systems, semantics of real-time languages, logical and discrete calculi (e.g. temporal logic) for specifying and verifying safety, liveness, deadlock, priority and fairness properties of reactive programs, and prolog tools for automating verification.
Number of offerings (in last five years): 0

Course: EECS 6412 3.0
Title: Data Mining
Short course description: This course introduces fundamental concepts of data mining. It presents various data mining technologies, algorithms and applications. Topics include association rule mining, classification models, sequential pattern mining and clustering.
Number of offerings (in last five years): 3

Course: EECS 6421 3.0
Title: Advanced Data Systems
Short course description: This course provides and introduction to and an in-depth study on several new developments in database systems and intelligent information systems. Topics include: Internet databases, data warehousing and OLAP, object-relational, object-oriented, and deductive databases.
Number of offerings (in last five years): 3

Course: EECS 6431 3.0
Title: Software Re-Engineering
Short course description: Industrial software systems are usually large and complex, while knowledge of their structure is either lost or inadequately documented. This course presents techniques that aid the comprehension and design recovery of large software systems.
Number of offerings (in last five years): 1

Course: EECS 6432 3.0
Title: Adaptive Software Systems
Short course description: Adaptive software systems are software systems that change their behaviour and structure to cope with changes in environment conditions or in user requirements. Adaptation includes self-optimization, self-protection, self-configuration and self-healing. This course covers basic and advanced concepts in engineering adaptive systems and has a special focus on self-optimization. It introduces the students to the mathematical foundations of adaptive systems including performance models, estimators for performance models, feedback loop architectures and strategies, and optimization.
Number of offerings (in last five years): 0

Course: EECS 6441
Title: Methods for Large-Scale Software
Short course description: This course studies the application of mathematical methods to the construction of large-scale software systems. It considers issues relevant to large-scale design and the application of mathematics. It involves a large-scale software project in which industrial-strength tools are applied.
Number of offerings (in last five years): 1

Course: EECS 6444 3.0
Title: Mining Software Engineering Data to Support the Development, Testing and Maintenance of Large Scale Software Systems
Short course description: Software engineering data (such as source code repositories, execution logs, performance counters, developer mailing lists and bug databases) contains a wealth of information about a project's status and history. Applying data mining techniques on such data, researchers can gain empirically based understanding of software development practices, and practitioners can better manage, maintain and evolve complex software projects.
Number of offerings (in last five years): 2

Course: EECS 6490A
Title: Concurrent Object Oriented Languages
Short course description: In this course, we focus on concurrent programming in the object oriented language Java. The course consists of three main parts. In the first part, we discuss concurrent programming in general. In the second part, we concentrate on writing concurrent programs in Java. In the third and final part, we look at techniques and tools to verify concurrent Java programs.
Number of offerings (in last five years): 3

Course: EECS 6490B 3.0
Title: Issues in Information Integration
Short course description: This course explores the challenges and research issues that arise in scaling current information systems technology to a widely-distributed heterogeneous database environment. The focus of the course is on using semantic information to integrate information sources, optimize query processing and provide cooperative response to a user in such systems. Topics to be covered in this course: heterogeneous database systems, management of uncertain (disjunctive) information, integrating relational and object-oriented database models, dynamic query processing, semantic query caching, semantic query optimization, and cooperative answering systems.
Number of offerings (in last five years): 2

Course: EECS 6590A 3.0
Title: Special Topics: High-Performance Computer Networks
Short course description: This course focuses on high performance computer networks. It presents a comprehensive study of modern high speed communication networks that is capable of providing data, voice, and video services. It also covers mobile and wireless communication networks.
Number of offerings (in last five years): 5

6. Comment on the expertise of the faculty who will actively support/participate the field and provide a Table of Faculty by field, as follows:

All faculty members mentioned in the table below conduct research in the field and supervise graduate students in the field.

Faculty Member & Rank	Home Unit	Primary Field	Category
Aijun An, Professor	EECS	Computer Science	Full member
Ebrahim Bagheri	EECS	Computer Science	Adjunct member
Melanie Baljko, Associate professor	EECS	Computer Science	Full member
Michael Brown, Professor	EECS	Computer Science	Full member
Marcus Brubaker, Assistant professor	EECS	Computer Science	Full member
Suprakash Datta, Assistant professor	EECS	Computer Science	Full member
Kosta Derpanis	EECS	Computer Science	Adjunct member
Patrick Dymond, Professor	EECS	Computer Science	Full member
Jeff Edmonds, Professor	EECS	Computer Science	Full member
Hasham ElSawy	EECS	Computer Science	Adjunct member
Petros Faloutsos, Professor	EECS	Computer Science	Full member
Parke Godfrey, Associate professor	EECS	Computer Science	Full member
Jarek Gryz, Professor	EECS	Computer Science	Full member
Rainer Herpers	EECS	Computer Science	Adjunct member

Jimmy Huang, Professor	ITEC	Computer Science	Full member
Michael Jenkin, Professor	EECS	Computer Science	Full member
Igor Jurisca	EECS	Computer Science	Adjunct member
Yves Lesperance, Associate professor	EECS	Computer Science	Full member
Burton Ma, Associate professor	EECS	Computer Science	Full member
Scott MacKenzie, Associate professor	EECS	Computer Science	Full member
UT Nguyen, Associate professor	EECS	Computer Science	Full member
Manos Papangelis, Assistant professor	EECS	Computer Science	Full member
Eric Ruppert, Associate professor	EECS	Computer Science	Full member
Mikhail Soutchanski	EECS	Computer Science	Adjunct member
Zbigniew Stachniak, Associate professor	EECS	Computer Science	Full member
George Tourlakis, Professor	EECS	Computer Science	Full member
John Tsotsos, Professor	EECS	Computer Science	Full member
Natalija Vlajic, Associate professor	EECS	Computer Science	Full member
Richard Wildes, Associate professor	EECS	Computer Science	Full member
Thilo Womelsdorf, Associate professor	BIOL	Computer Science	Associate member
Jia Xu, Associate professor	EECS	Computer Science	Associate member
Xiaohui Yu, Associate professor	ITEC	Computer Science	Full member

Note: Up-to-date CVs of faculty who will actively participate in delivering the graduate program must be included as an appendix.

7. Comment on the projected in-take into the field, including the anticipated implementation date (i.e. year and term of initial in-take), and indicate if the projected in-take is within or in addition to the existing enrolment targets for the parent program.

The in-take into the field is projected to be at least five students per year. This projected in-take is within the existing enrolment targets for the parent program. The anticipated implementation date is Fall 2017.

8. Comment on the impact of the field on the parent program, focusing on the extent of diversion of faculty from existing graduate courses and/or supervision, as well as the capacity of the program to absorb any anticipated additional enrolment.

The introduction of this field will *not* impact the parent program. The course requirements are not changed. There is sufficient capacity for supervision.

9. Support statements

- **from the relevant Dean(s)/Principal, with respect to the adequacy of existing resources necessary to support the new field, as well as the commitment to any plans for new/additional resources necessary to implement and/or sustain the new field**
- **from the relevant Faculties/units/programs confirming consultation on/support for the new program, as appropriate**
- **from professional associations, government agencies or policy bodies with respect to the need/demand for the proposed program, as appropriate**

Graduate Fields
Definition and Proposal Template

Definition

In graduate programs, field refers to an area of specialization or concentration (in multi/interdisciplinary programs a clustered area of specialization) that is related to the demonstrable and collective strengths of the program's faculty. Institutions are not required to declare fields at either the master's or doctoral level. Institutions may wish, through an expedited approval process, to seek the endorsement of the Quality Council.

Graduate Field Proposal Guidelines

1. Indicate the name of the field being proposed and identify the parent program.

Field: Electrical Engineering
Parent program: PhD in Electrical Engineering and Computer Science

2. Provide a description of the field (its intellectual focus, etc.) including the appropriateness and consistency of the field name with current usage in the discipline or area of study.

Electrical engineering refers to the study and application of electric power, electronic circuits, electro-magnetics and electrical signals. The study of electrical engineering is paramount in our technological based world; it facilitates the development of all our electronic devices and enables reliable and efficient power delivery to our homes and offices. Within the electrical engineering field of the graduate program, students will significantly expand and deepen their understanding of the theory and practice of electrical engineering in at least one of the following areas: power engineering and energy systems, electronics, and bio-medical devices. Research focuses on the development of advanced high power conversion power architectures, intelligent smart-grid control and protection techniques, energy efficient electronic devices, renewable energy, and innovative biomedical sensors for clinical applications, and advanced analog and digital integrated circuits (ICs) for high-speed communications and low-power computation.

3. Comment on the relationship of the admission requirements for the field to those of the parent program. If the same, describe the program admission requirements. If different, describe the field admission requirements, indicate how they are different from those of the parent program, and provide a rationale for the difference in relation to the focus and learning outcomes of the field.

The admission requirements for the field are the same as those of the parent program. These admission requirements are the following.

Applicants must have a Master's degree in Computer Engineering, Computer Science, Electrical Engineering, Software Engineering, or closely related field, which is equivalent to the MSc degree in Computer Science (thesis option) or the MASc degree in Electrical and Computer Engineering at York University. A minimum average grade of B+ on all course work is required. Applications must include official copies of all academic transcripts, an extended abstract/copy of the MSc or MASc thesis, three letters of reference and a one-page statement of purpose and previous experience. The statement of purpose should indicate the applicant's area(s) of interest. The following are the minimum English language test scores (if required): TOEFL 577 (paper-based) or 90-91 (Internet-based), IELTS 7, or York English Language Test 4. The Graduate Record Examination general test and subject test are strongly recommended, especially for applicants who did their work outside of Canada and the United States.

4. Comment on the relationship of the curricular requirements for the field to those of the parent program. If the same, describe the program requirements. If different, describe the field requirements, indicate how they are different from those of the parent program, and provide a rationale for the difference in relation to the focus and learning outcomes of the field.

The curricular requirements for the field are the same as those of the parent program. These curricular requirements are the following.

Candidates for the PhD degree must complete at least three three-credit graduate courses. No more than one-third of the course requirements can be integrated with undergraduate courses. Candidates must successfully complete a qualifying examination consisting of a written report on the candidate's field of interest and have an oral defence of the report. Candidates must present a dissertation proposal outlining the anticipated results of their dissertation. Each term candidates must attend departmental seminars. Each fall and winter term, candidates must attend one professional development seminar. Candidates must complete either an industrial internship or a teaching practicum. Candidates must conduct a significant body of original research under the supervision of a supervisory committee and successfully defend the resulting dissertation.

The qualifying examination, the dissertation proposal and the dissertation must be on topics within the field of electrical engineering. The fields of the student's supervisor must include electrical engineering (see item 6).

5. Provide a list of courses that will be offered in support of the field. The list of courses must indicate the unit responsible for offering the course (including cross-lists and integrations, as appropriate), the course number, the credit value, the short course description, and whether or not it is an existing or new course. For existing courses, the frequency of offering should be noted. For new courses, full course proposals are required and should be included in the proposal as an appendix. (The list of courses may be organized to reflect the manner in which the courses count towards the program/field requirements, as appropriate; e.g. required versus optional; required from a list of specified courses; specific to certain concentrations, streams or fields within the program, etc.)

All the courses listed below are optional. They are all existing courses. The supervisor plays an important role in the course selection and will normally encourage students to take these courses.

Course: EECS 5441 3.0
Title: Real-Time Systems Theory
Short course description: Specification and verification techniques for real-time systems with many interacting components. Formal design of real-time systems using (a) programming languages with unambiguous semantics of time-related behavior and (b) scheduling algorithms.
Number of offerings (in last five years): 2

Course: EECS 5612 3.0
Title: Digital Very Large Scale Integration

Short course description: A course on modern aspects of VLSI CMOS chips. Key elements of complex digital system design are presented including design automation, nanoscale MOS fundamentals, CMOS combinational and sequential logic design, datapath and control system design, memories, testing, packaging, I/O, scalability, reliability, and IC design economics.
Number of offerings (in last five years): 0

Course: EECS 6221 3.0
Title: Statistical Signal Processing Theory
Short course description: This course introduces theory and algorithms of stochastic signals and their applications to the real world. Discrete random variables, random vectors, and stochastic processes are reviewed followed by signal processing methods used for detection, estimation, and optimal filtering.
Number of offerings (in last five years): 0

Course: EECS 6324 3.0
Title: From Control to Actuators
Short course description: A "robot building course", this course will follow the issues involved in building a robot or robotic system from control to actuators. This includes microcomputer control, actuator design, high-level software models, and sensor inputs.
Number of offerings (in last five years): 1

Course: EECS 6601 3.0
Title: Nanoelectronics
Short course description: The sustained demand for increased memory and computational power has driven the physical size of electronic components to nanoscale dimensions. The need to investigate size effects and to find viable ways to manufacture at the nanoscale has also led to the discovery of new phenomena and functionality. This course covers electronic transport and other properties in nanoscale systems, devices, characterization and fabrication techniques. Topics to be covered include quantum confinement, quantum dots, nanowires, 2D electron gases, single electron transistors, spintronic devices, electronic transport and optical properties, nanoscale materials, top-down and bottom-up fabrication approaches.
Number of offerings (in last five years): 1

Course: EECS 6701 3.0
Title: High Frequency Power Electronic Converters
Short course description: This course discusses the fundamentals of loss-less switching techniques in high frequency power converters: zero-voltage switching and zero-current switching. The course then focuses on various resonant converter topologies and soft-switching converters with auxiliary storage elements. The course then discusses various control techniques used in high frequency power converters. Special emphasis is placed on the design techniques using practical examples.
Number of offerings (in last five years): 0

Course: EECS 6704 3.0
Title: Smart Distribution Grids
Short course description: The following topics are covered: introduction to electric power distribution system structure and components; concept of distributed and renewable energy resources (DG); distribution system load/DG characteristics and modelling; integration of DG in power flow analysis; voltage and reactive power planning and control with consideration of DG; self-healing mechanisms; microgrids concept, planning, operation, and energy management.
Number of offerings (in last five years): 1

Course: EECS 6801 3.0
Title: Advanced Microelectronic Biochips
Short course description: This course offers an introduction to the Biochips. This course takes a multi-path approach: micro-fabrication techniques, microelectronic design and implementation of bio interfaces offering a vital contemporary view of a wide range of integrated circuits and system for electrical, magnetic, optical and mechanical sensing and actuating devices and much more; classical knowledge of biology, biochemistry as well as micro-fluidics. The coverage is both practical and in depth integrating experimental, theoretical and simulation examples.
Number of offerings (in last five years): 0

Course: EECS 6803 3.0
Title: Micro-fluidics for Cellular and Molecular Biology
Short course description: This course offers an introduction to the micro-fluidics for life science applications. This course offers a unique opportunity to all science, health and engineering students to learn the fundamental of micro-fluidic technologies for a variety of cellular and molecular applications. The coverage is both practical and in depth integrating experimental, theoretical and simulation examples.
Number of offerings (in last five years): 0

The undergraduate program in Electrical Engineering started less than four years ago. As a result, fourth year courses are offered for the first time this year. The number of graduate course in Electrical Engineering and their frequency of offering are expected to grow over the next five years.

6. Comment on the expertise of the faculty who will actively support/participate the field and provide a Table of Faculty by field, as follows:

All faculty members mentioned in the table below conduct research in the field and supervise graduate students in the field.

Faculty Member & Rank	Home Unit	Primary Field	Category
Andrew Eckford, Associate professor	EECS	Electrical Engineering	Full member
Hany Farag, Assistant professor	EECS	Electrical Engineering	Full member
Ebrahim Ghafar-Zadeh, Assistant professor	EECS	Electrical Engineering	Full member
Gerd Grau, Assistant professor	EECS	Electrical Engineering	Full member
Ali Hooshyar, Assistant Professor	EECS	Electrical Engineering	Full member
Richard Hornsey, Professor	EECS	Electrical Engineering	Full member
Ahmed Hussein	EECS	Electrical Engineering	Adjunct member
Hossein Kassiri, Assistant professor	EECS	Electrical Engineering	Full member
Matthew Kyan, Associate professor	EECS	Electrical Engineering	Full member
John Lam, Assistant Professor	EECS	Electrical Engineering	Full member
Peter Lian, Professor	EECS	Electrical Engineering	Full member
Sebastian Magierowski, Associate professor	EECS	Electrical Engineering	Full member
Simone Pisana, Associate professor	EECS	Electrical Engineering	Full member
Amir Sodagar, Associate professor	EECS	Electrical Engineering	Full member

Note: Up-to-date CVs of faculty who will actively participate in delivering the graduate program must be included as an appendix.

7. Comment on the projected in-take into the field, including the anticipated implementation date (i.e. year and term of initial in-take), and indicate if the projected in-take is within or in addition to the existing enrolment targets for the parent program.

The in-take into the field is projected to be at least two students per year. Given that Electrical Engineering is relatively new to York University, this number is expected to increase in the near future. This projected in-take is within the existing enrolment targets for the parent program. The anticipated implementation date is Fall 2017.

The field highlights Electrical Engineering within the broad spectrum covered by the parent program (from Electrical Engineering to Computer Science) and, therefore, will be helpful for recruitment of new students. Within the highly regulated discipline of engineering, recognition of the type of engineering is very valuable for graduates when seeking employment.

8. Comment on the impact of the field on the parent program, focusing on the extent of diversion of faculty from existing graduate courses and/or supervision, as well as the capacity of the program to absorb any anticipated additional enrolment.

The introduction of this field will *not* impact the parent program. The course requirements are not changed. There is sufficient capacity for supervision.

9. Support statements

- **from the relevant Dean(s)/Principal, with respect to the adequacy of existing resources necessary to support the new field, as well as the commitment to any plans for new/additional resources necessary to implement and/or sustain the new field**
- **from the relevant Faculties/units/programs confirming consultation on/support for the new program, as appropriate**
- **from professional associations, government agencies or policy bodies with respect to the need/demand for the proposed program, as appropriate**

Graduate Fields
Definition and Proposal Template

Definition

In graduate programs, field refers to an area of specialization or concentration (in multi/interdisciplinary programs a clustered area of specialization) that is related to the demonstrable and collective strengths of the program's faculty. Institutions are not required to declare fields at either the master's or doctoral level. Institutions may wish, through an expedited approval process, to seek the endorsement of the Quality Council.

Graduate Field Proposal Guidelines

1. Indicate the name of the field being proposed and identify the parent program.

Field: Software Engineering
Parent program: PhD in Electrical Engineering and Computer Science

2. Provide a description of the field (its intellectual focus, etc.) including the appropriateness and consistency of the field name with current usage in the discipline or area of study.

Software enables technological advances that lead to new, high-performance products and systems in every commercial sector, including medical devices, automobiles, aircrafts, power generation systems, mobile phones, and entertainment systems. As a product and a system's functionality grow, so does the need to efficiently and correctly implement the complex software that enables growth. Software engineering is the application of a systematic, disciplined, quantifiable approach to the design, development, testing, operation, and maintenance of software. Software engineering principles and practices are essential for the development of large, complex, or trustworthy systems. Within the software engineering field of the graduate program, students will significantly expand and deepen their understanding of the theory and practice of software engineering in at least one of the following areas: software requirements, architecture and design, testing, verification, and maintenance. Students will gain the necessary skills to evaluate, adapt and develop software engineering processes, metrics, and tools.

3. Comment on the relationship of the admission requirements for the field to those of the parent program. If the same, describe the program admission requirements. If different, describe the field admission requirements, indicate how they are different from those of the parent program, and provide a rationale for the difference in relation to the focus and learning outcomes of the field.

The admission requirements for the field are the same as those of the parent program. These admission requirements are the following.

Applicants must have a Master's degree in Computer Engineering, Computer Science, Electrical Engineering, Software Engineering, or closely related field, which is equivalent to the MSc degree in Computer Science (thesis option) or the MASc degree in Electrical and Computer Engineering at York University. A minimum average grade of B+ on all course work is required. Applications must include official copies of all academic transcripts, an extended abstract/copy of the MSc or MASc thesis, three letters of reference and a one-page statement of purpose and previous experience. The statement of purpose should indicate the applicant's area(s) of interest. The following are the minimum English language test scores (if required): TOEFL 577 (paper-based) or 90-91 (Internet-based), IELTS 7, or York English Language Test 4. The Graduate Record Examination general test and subject test are strongly recommended, especially for applicants who did their work outside of Canada and the United States.

4. Comment on the relationship of the curricular requirements for the field to those of the parent program. If the same, describe the program requirements. If different, describe the field requirements, indicate how they are different from those of the parent program, and provide a rationale for the difference in relation to the focus and learning outcomes of the field.

The curricular requirements for the field are the same as those of the parent program. These curricular requirements are the following.

Candidates for the PhD degree must complete at least three three-credit graduate courses. No more than one-third of the course requirements can be integrated with undergraduate courses. Candidates must successfully complete a qualifying examination consisting of a written report on the candidate's field of interest and have an oral defence of the report. Candidates must present a dissertation proposal outlining the anticipated results of their dissertation. Each term candidates must attend departmental seminars. Each fall and winter term, candidates must attend one professional development seminar. Candidates must complete either an industrial internship or a teaching practicum. Candidates must conduct a significant body of original research under the supervision of a supervisory committee and successfully defend the resulting dissertation.

The qualifying examination, the dissertation proposal and the dissertation must be on topics within the field of software engineering. The fields of the student's supervisor must include software engineering (see item 6).

5. Provide a list of courses that will be offered in support of the field. The list of courses must indicate the unit responsible for offering the course (including cross-lists and integrations, as appropriate), the course number, the credit value, the short course description, and whether or not it is an existing or new course. For existing courses, the frequency of offering should be noted. For new courses, full course proposals are required and should be included in the proposal as an appendix. (The list of courses may be organized to reflect the manner in which the courses count towards the program/field requirements, as appropriate; e.g. required versus optional; required from a list of specified courses; specific to certain concentrations, streams or fields within the program, etc.)

All the courses listed below are optional. They are all existing courses. The supervisor plays an important role in the course selection and will normally encourage students to take these courses.

Course: EECS 5421 3.0
Title: Operating System Design
Short course description: A modern operating system has four major components: process management, input/output, memory management, and the file system. This project-oriented course puts operating system principles into action and presents a practical approach to studying implementation aspects of operating systems. A series of projects are included for students to acquire direct experience in the design and construction of operating system components and have each interact correctly with the existing software. The programming environment is C/C++ under UNIX.

Number of offerings (in last five years): 5

Course: EECS 5441 3.0
Title: Real-Time Systems Theory
Short course description: Specification and verification techniques for real-time systems with many interacting components. Formal design of real-time systems using (a) programming languages with unambiguous semantics of time-related behavior and (b) scheduling algorithms.
Number of offerings (in last five years): 2

Course: EECS 5442 3.0
Title: Real-Time Systems Practice
Short course description: The course will focus on the technologies related to the design and implementation of real-time systems. Topics may include: typical real-time applications, process models of real-time systems, scheduling technologies in real-time systems, design and implementation of real-time systems software, real-time systems hardware, real-time operating systems, real-time programming languages, and inspection and verification methods for real-time systems
Number of offerings (in last five years): 4

Course: EECS 6390B 3.0
Title: Scheduling in Hard Real-Time Systems
Short course description: This course discusses concepts and methods for satisfying timing constraints in large, complex hard-real-time systems. Topics include: characteristics of hard-real-time systems, timing constraints, periodic and asynchronous processes, run-time and pre-run-time scheduling, cyclic executives, priority scheduling, preemptive and non-preemptive scheduling, synchronization, schedulability analysis, resource management, and real-time programming language constructs.
Number of offerings (in last five years): 1

Course: EECS 6411 3.0
Title: Programming Logic for Complex Systems
Short course description: This course covers program verification methods for a class of programs, commonly referred to as reactive programs. Reactive programs typically never terminate and are run in order to maintain some interaction with the environment. An adequate description of reactive systems must refer not only to initial and final states, but also to the ongoing behavior as a (possibly infinite) sequence of states and events. The purpose of this course is to investigate the use of logical calculi for the specification, design and verification of reactive systems. Topics include: modeling of discrete event systems, semantics of real-time languages, logical and discrete calculi (e.g. temporal logic) for specifying and verifying safety, liveness, deadlock, priority and fairness properties of reactive programs, and prolog tools for automating verification.
Number of offerings (in last five years): 0

Course: EECS 6412 3.0
Title: Data Mining
Short course description: This course introduces fundamental concepts of data mining. It presents various data mining technologies, algorithms and applications. Topics include association rule mining, classification models, sequential pattern mining and clustering.
Number of offerings (in last five years): 3

Course: EECS 6431 3.0
Title: Software Re-Engineering
Short course description: Industrial software systems are usually large and complex, while knowledge of their structure is either lost or inadequately documented. This course presents techniques that aid the comprehension and design recovery of large software systems.
Number of offerings (in last five years): 1

Course: EECS 6432 3.0
Title: Adaptive Software Systems
Short course description: Adaptive software systems are software systems that change their behaviour and structure to cope with changes in environment conditions or in user requirements. Adaptation includes self-optimization, self-protection, self-configuration and self-healing. This course covers basic and advanced concepts in engineering adaptive systems and has a special focus on self-optimization. It introduces the students to the mathematical foundations of adaptive systems including performance models, estimators for performance models, feedback loop architectures and strategies, and optimization.
Number of offerings (in last five years): 0

Course: EECS 6441
Title: Methods for Large-Scale Software
Short course description: This course studies the application of mathematical methods to the construction of large-scale software systems. It considers issues relevant to large-scale design and the application of mathematics. It involves a large-scale software project in which industrial-strength tools are applied.
Number of offerings (in last five years): 1

Course: EECS 6444 3.0
Title: Mining Software Engineering Data to Support the Development, Testing and Maintenance of Large Scale Software Systems
Short course description: Software engineering data (such as source code repositories, execution logs, performance counters, developer mailing lists and bug databases) contains a wealth of information about a project's status and history. Applying data mining techniques on such data, researchers can gain empirically based understanding of software development practices, and practitioners can better manage, maintain and evolve complex software projects.
Number of offerings (in last five years): 2

Course: EECS 6490A
Title: Concurrent Object Oriented Languages
Short course description: In this course, we focus on concurrent programming in the object oriented language Java. The course consists of three main parts. In the first part, we discuss concurrent programming in general. In the second part, we concentrate on writing concurrent programs in Java. In the third and final part, we look at techniques and tools to verify concurrent Java programs.
Number of offerings (in last five years): 3

Note that two courses have not been taught in the last five year. One of them, EECS 6432, is offered in fall 2016. The other course has not been offered since faculty members qualified to teach it were assigned to teach other courses.

6. Comment on the expertise of the faculty who will actively support/participate the field and provide a Table of Faculty by field, as follows:

All faculty members mentioned in the table below conduct research in the field and supervise graduate students in the field.

Faculty Member & Rank	Home Unit	Primary Field	Category
Jack Jiang, Assistant professor	EECS	Software Engineering	Full member
Sotirios Liaskos, Associate professor	ITEC	Software Engineering	Full member
Marin Litoiu, Associate professor	ITEC	Software Engineering	Full member
Jonathan Ostroff, Professor	EECS	Software Engineering	Full member
Vassilios Tzerpos, Associate professor	EECS	Software Engineering	Full member
Franck van Breugel, Professor	EECS	Software Engineering	Full member

Note: Up-to-date CVs of faculty who will actively participate in delivering the graduate program must be included as an appendix.

7. Comment on the projected in-take into the field, including the anticipated implementation date (i.e. year and term of initial in-take), and indicate if the projected in-take is within or in addition to the existing enrolment targets for the parent program.

The in-take into the field is projected to be at least two students per year. This projected in-take is within the existing enrolment targets for the parent program. The anticipated implementation date is Fall 2017.

The field highlights Software Engineering within the broad spectrum covered by the parent program (from Electrical Engineering to Computer Science) and, therefore, will be helpful for recruitment of new students. Within the highly regulated discipline of engineering, recognition of the type of engineering is very valuable for graduates when seeking employment.

8. Comment on the impact of the field on the parent program, focusing on the extent of diversion of faculty from existing graduate courses and/or supervision, as well as the capacity of the program to absorb any anticipated additional enrolment.

The introduction of this field will *not* impact the parent program. The course requirements are not changed. There is sufficient capacity for supervision.

9. Support statements

- **from the relevant Dean(s)/Principal, with respect to the adequacy of existing resources necessary to support the new field, as well as the commitment to any plans for new/additional resources necessary to implement and/or sustain the new field**
- **from the relevant Faculties/units/programs confirming consultation on/support for the new program, as appropriate**
- **from professional associations, government agencies or policy bodies with respect to the need/demand for the proposed program, as appropriate**

MEMO

FROM THE OFFICE OF THE DEAN AT THE
LASSONDE SCHOOL OF ENGINEERING

LASSONDE SCHOOL OF ENGINEERING
YORK UNIVERSITÉ UNIVERSITY

150 ATKINSON BUILDING - 4700 KEELE STREET
TORONTO, ONTARIO, CANADA M3J 1P3

TO Franck van Breugel, Graduate Program Director

FROM Janusz Kozinski, Dean, Lassonde School of Engineering

SUBJECT Graduate Program in Electrical Engineering and Computer Science

DATE September 27, 2016

It gives me great pleasure to offer my enthusiastic support for these proposed changes to the PhD program in Electrical Engineering and Computer Science. The introduction of fields in Computer Engineering, Computer Science, Electrical Engineering, and Software Engineering are essential to the future of the department and the Lassonde School of Engineering. These changes to the program play a pivotal role in the development of Engineering at York and are complementary to the recent introduction of undergraduate programs in Software Engineering and Electrical Engineering and the significant increase in faculty complement in these areas. These changes in the graduate program truly represents a natural next step in the expansion of York's Department of Electrical Engineering and Computer Science.

The program development was informed by a careful planning phase, careful consultation and benchmarking against leading programs in Canada and the US. The initiative is fully aligned with the strategic directions of the Lassonde School of Engineering and the University. Our strategic planning envisions a multi-phase development for Engineering at York, in which Electrical Engineering features prominently as a cornerstone. The proposal is also aligned with the principal goals of the most recent University Academic Plan and the Provostial White Paper, which call for expansion of the scope of the University's teaching and research activities in the areas of engineering and applied science.

The resources for the changes to the program are incremental and the most significant impact is the need to supervise students in the new field. Faculty members have been hired in these areas and new colleagues will join in the next few years thus supervisory capacity is more than adequate (indeed most of these faculty have been supervising students in the existing program although it is difficult for some faculty to attract students due to the current mismatch in fields and faculty research foci). Faculty complement planning has been developed in the context of the larger planning exercise for the expansion of Engineering at York. The academic financial resources and planning processes will be subject to a very stringent planning and accountability framework, as would be expected with any project of the magnitude and size as envisioned for the Lassonde School of Engineering.

Plans for faculty complement and enrollment growth have been developed to strike the essential balance between professional and academic standards, with the average student-to-faculty ratios aligning with comparable programs of similar size. Resources for the appropriate administrative, technical and student support staff have already been built into the plans for expansion and will be allocated as the new program comes online.

In conclusion, I am pleased to offer my strong support for the changes to the doctoral program in Electrical Engineering and Computer Science in the Lassonde School of Engineering.

Cc: S. Pagiatakis
R. Allison

FUGRO ROADWARE

2505 Meadowvale Blvd
Mississauga, ON
L5N 5S2
Canada

Tel: +1 905 567 2870
Fax: +1 905 567 2871
Toll-Free NA: 800-828-ARAN (2726)
Web: www.fugroroadware.com

Professor Franck van Breugel
Director, Graduate Program in Electrical Engineering & Computer Science
York University
4700 Keele Street
Toronto, ON
M3J 1P3

Dear Professor van Breugel:

As President of Fugro Roadware, I am writing to support the modification of the PhD Program in Computer Science at York University to become the PhD Program in Electrical Engineering & Computer Science (EECS), including a Field of Study in Computer Engineering.

Founded in 1969, Fugro Roadware is a Mississauga based multinational company with over 40 years of experience in Infrastructure Asset Management Technology and Data Collection service. Fugro Roadware is widely known and respected for its innovative and customer driven Automatic Road Analyzer (ARAN) technology. First developed in the mid-1980s, the modular ARAN is a multifunctional vehicle used for road surface data collection. Our ARAN vehicles are equipped with laser rangefinders for measuring the pavement surface for roughness, texture, rutting and distresses. Calibrated right-of-way (ROW) cameras facilitate the collection of imagery for customer viewing and road asset inventory. Pavement cameras and 3D scanners provide an orthographic range view of the pavement for measurement, visualization and evaluation. Telemetry and geo referencing are conducted via a state-of-the-art navigation (POS) system. ARAN is currently employed by numerous transportation and infrastructure agencies around the world and its subsystems and software form the core of an integrated pavement management solution.

Computer Engineering is a well-established and recognized credential. Fugro Roadware depends on key employees with advanced degrees in Computer Engineering and related fields for continuing innovation of its product and service lines, and in fact is currently hosting a York EECS PhD student as an intern. We fully support the proposed modification of the York graduate program to incorporate the field of Computer Engineering.

Sincerely,

David Lowe
President, Fugro Roadware

Voice: (613) 270-4718
Fax: (613) 738-0002
Email: marcellus_mindel@ca.ibm.com

3755 Riverside Drive
Ottawa, Ontario K1G 4K9

September 22, 2016

Mamdouh Shoukri
President and Vice-Chancellor
York University
1050 Kaneff Tower
4700 Keele Street
Toronto, ON M3J 1P3

Re: York University PhD program in Software Engineering

Dear President Shoukri,

I am writing on behalf of the IBM Canada Lab Centre for Advanced Studies to express our strongest support for the creation of a PhD Program in Electrical Engineering and Computer Science at York University. Specifically, we are eager to support the field of *Software Engineering* as a specialization area within this program.

The IBM Canada Lab Centre for Advanced Studies is part of a world-wide group of centres that facilitates joint IBM-academic applied research, innovation, education and entrepreneurship related to IBM's offerings and technology platforms.

IBM invests more than a quarter of a billion dollars annually in R&D activities in Canada, and contributes over a billion dollars of annual exports to Canada's balance of trade. Our Canadian development labs in Toronto, Ottawa, Fredericton, Victoria and London develop software solutions in big data and analytics, security, cloud, mobile and social business for the global market.

Software development using sound software engineering principles is fundamental to IBM's business in general and to the success of the IBM Canada Lab in particular. It requires highly qualified researchers and developers to create, maintain and operate the new generations of software products and services that touch every aspect of our lives.

York University has played an important part in the 26 year history of the Centre for Advanced Studies. We are justifiably proud of our continuing track record of successful and impactful research projects with York faculty and graduate students, and the close ties that we have forged.

As well, both York and IBM continue to make important contributions to the Canadian Consortium for Software Engineering Research (CSER). In the past, we worked together through CSER to establish Masters programs in Software Engineering at York and other Canadian universities. These efforts helped build a base of software engineering expertise in Canada that has been instrumental in securing new and expanding missions for the IBM Canada Lab, and as well, advanced Canadian institutions as globally recognized leaders in software engineering research. We continue as important members of CSER today, and host the CSER annual general meeting as part of our IBM CASCON conference. (http://cascon.ca)

We anticipate that the new PhD program in Software Engineering at York University will create new generations of researchers with the skills and knowledge to address the challenges of ever increasing software importance, complexity and pervasiveness. We look forward to future exceptional joint research involving York PhD candidates, as well as the potential unlocked by their transition to IBM fulltime employment after successfully defending their theses.

Sincerely,

Marcellus

Marcellus Mindel
Head, Centre for Advanced Studies
IBM Canada Laboratory

Phone: (613) 270-4718
Fax: (613) 738-0002
Email: marcellus_mindel@ca.ibm.com

RE: **Support for the modifications proposed by York's Department of Electrical Engineering and Computer Science (EECS) to its PhD program**

To whom it may concern,

Please accept this letter as a formal declaration of the Mircom Group of Companies (MGC™) support of the modifications proposed by York's Department of Electrical Engineering and Computer Science (EECS) to its PhD program.

We understand that the major changes proposed to the program include:

- A change of the degree program name to "PhD in Electrical Engineering and Computer Science" from "PhD in Computer Science".
- Introduction of the fields: Computer Engineering, Computer Science, Electrical Engineering, and Software Engineering.
- Change of the admission requirements.
- Change of the degree requirements.

As representatives of MGC we strongly agree with all of the above initiatives. The following details the reasons for our support.

MGC is headquartered in Vaughan, Ontario, less than 2-km from York University's Keele campus. MGC is North America's largest independent designer, manufacturer and distributor of life safety equipment. The company's activities include the engineering and construction of sensors, computing and communications hardware, software and networking along with the test and verification of systems integrating such components. Examples of MGC end products include customized fire detection, mass notification, and telephone entry systems. Besides evolving the technology already serving their existing market base, MGC's research and development team is involved in the application of the company's expertise towards emerging opportunities in sensors and networking for life safety. Technically, the systems solutions developed by MGC span the gamut of Computer, Electrical, and Software Engineering as well as Computer Science. MGC's industry leadership is benefits greatly from the expertise of graduates in such areas.

We strongly endorse the proposed change in degree name. Having interacted and established collaborations with faculty in York's EECS department we feel that such a name fully encompasses the nature of the expertise, research, and graduate-level education offered by this developing Department. In addition to a more accurate reflection of the Department's activities, we believe that such a name change should help attract more graduate students whose specific area of interest lies in the electrical, computer, and software engineering domains. To our knowledge, the amalgam of engineering and computer science is unique among Canadian PhD programs and should further boost the interest of engineering and computer scientists alike. Given our broad expertise in these areas at MGC we are confident of the energetic synergies that exist between these areas and their associated sub-specialties. To not adopt the EECS label given the Department's historical foundation in computer science and its emerging expertise in electrical engineering would fail to leverage a substantial advantage already in the Department's possession.

Understanding the benefit of the EECS brand to the Department's PhD program we also recognize the importance of conveying more directly the focus of a graduate student's research to potential employers. For this reason we support the EECS Department's plan to introduce the aforementioned four fields as a means of designating the field of specialization on one's transcripts. As practitioners we can vouch for the validity and utility of the named specializations. Given its relatively brief history at York, we wish to particularly extend our support for the Electrical Engineering (EE) specialization which is an extremely well-established and recognized credential and of particular interest to the activities of MGC as well as countless other local, national, and international engineering entities. We wish to further highlight the strong demand for PhD graduates in the field of electrical engineering. Again this is not only the case for MGC which values EEs from such specialties as sensors, circuits, communications, control systems, and networks, but for a multitude of other industries in optics, nanotechnology, signal processing, power and energy, semiconductors, robotics, etc.

In conclusion we wish to re-iterate our support for the EECS Department's proposed changes to its PhD program. We feel that adopting the PhD in Electrical Engineering and Computer Science is not only an accurate statement of the Department's expertise but itself promotes a research environment unique to the country and will help attract more graduate interest. Adoption of the four fields of specialization is also a change we support, especially with regards to the Department's newly established electrical engineering pursuits, which should be further bolstered by providing applicants with interest in this area to further distinguish their PhD activities.

Sincerely,

J. FALBO

Jason Falbo, P.Eng., M.B.A,

Vice President, Engineering

THE MIRCOM GROUP OF COMPANIES.

☎ 905.660.4655 Ext 7025 905.695.3536 jfalbo@mircom.com