ENGINEERING SOFTWARE OPTION

for

Civil, Environmental (Civil) & Geological Engineering Students

Ever increasing demands are being placed on all engineering graduates to have the information technology skills needed to apply effectively the principles of their individual disciplines in professional practice. More and more graduates of all disciplines are becoming involved in the development of computer-based productivity tools for the engineering profession. Indeed, engineers are vitally needed in software development because their background and training are invaluable to understanding, and getting right, the engineering principles underlying the software application. This is particularly true for civil, environmental and geological engineers as information technology becomes more and more pervasive across the broad range of structural, transportation, geological, environmental and water resources engineering.

In response to the needs noted in the foregoing, the Faculty of Engineering has worked over the past two years to develop a Software Option for Engineering Students. This option includes material ranging from software skills to software development and design techniques that will enhance any engineer's ability to perform engineering tasks using computer and information technologies. The option is comprised of four core technical courses, a fourth year design project with a significant software design component, one complementary studies impact course that includes material on 'software, the professional engineer and society', and two elective courses that focus on engineering applications and the software design process. The option is primarily intended for undergraduate students in Chemical, Civil, Environmental (Chemical/Civil), Geological, Mechanical and Systems Design Engineering.

The first group of students eligible to take the Software Option for Engineering Students will be those entering the 2A term in Winter 2001. Civil engineering students in their 2A term in Fall 2000 who would like to enroll in the option may register to take Gen. E. 240: *Algorithms & Data Structures* in their 2B term in Spring 2001.

A concise outline of the Software Option for Engineering Students is presented in the following. If you have an additional need for information, please contact Professor D. E. Grierson in CPH-2369C, ext. 2412, email: grierson@uwaterloo.ca

Software Option for Engineering Students

Technical Core Courses

GenE 121: Digital Computation (or equivalent)
GenE 240: Algorithms and Data Structures
GenE 241: Introduction to Computer Structures and Real-time Systems
GenE 342: Principles of Software Engineering
Core Design Project

One of: ChE 046/047, CivE 400/401, EnvE 430/431/483, GeoE 400/401, ME 481/482, SyDe 461/462.

The Associate Chair (or designate) in each Department is responsible for ensuring that the project has a software design component that utilizes significant material from the technical core courses for the Option.

Complementary Studies Impact Course (CSE-List A)

MSci 442: Impact of Information Systems on Organizations and Society (or equivalent).

Technical Elective Courses

The student is required to take two Technical Elective courses, one selected from List A and the other from List B below. List A courses are intended to provide a discipline-specific application in which students can apply the software skills learned in the Option. List B courses predominantly focus on software design and include such topics as requirements, specification, implementation, testing, maintenance and management of the software design process.

List A: Application-Oriented Courses (Select one)

ChE 524: Process Control laboratory
CivE 422: Finite Element Analysis
E&CE 413: Digital Signal Processing
E&CE 485: Computer Controlled Applications
ME 447: Advanced Manufacturing Technologies
ME 555: Computer Aided Design
ME 559: Finite Element Methods
ME 566: Fluid Mechanics 3
SyDE 454: Computer Simulation of Systems
SyDE 558: Fuzzy Logic and Neural Networks
SyDE 575: Image Processing

List B: Software-Focused Courses (Select one)

E&CE 428: Computer Networks and Security
E&CE 454: Distributed and Network Computing
E&CE 456: Database Systems
GenE 344: Programming Languages and Translators (cross-listed with E&CE 251)
SyDE 422: Machine Intelligence (or E&CE 457: Applied Artificial Intelligence)

Term Schedule

Software Option for Civil, Environmental (Civil) & Geological Engineering Students

Please note from the table below that the technical core courses in the Software Option may be taken:

1) by shifting a complementary studies elective (CSE) course;
2) as an extra (E) course;
3) possibly as a technical elective (TE).

The course sequence shown in boldface has a minimum impact on the program.

	2A F	2B S	3A W	3B F	4A S
CivE		**GenE 240(CSE)**	GenE 240(CSE) GenE 241(CSE) **MSci 442(CSE-A)**	**GenE 241(CSE)** GenE 342(CSE/TE)	**GenE 342(TE)**
	2A W	**2B F**	**3A S**	**3B W**	**4A F**
EnvE (Civil)	**GenE 240(E)**	**GenE 241(CSE)**	GenE 240(E) GenE 342(E)	GenE 241(E) **MSci 442(CSE-A)**	**GenE 342(TE)**
GeoE	**GenE 240(CSE)**	**GenE 241(E)**	GenE 240(E) GenE 342(E)	GenE 241(E) **MSci 442(CSE-A)**	**GenE 342(CSE)**

CSE: *The option does not involve an extra course as the CSE course in the term could be shifted elsewhere (perhaps by taking it as a Distant Education Course or at another university in a work term)*

E: *The option requires the student to take an Extra course above the standard load for the term*

TE: *At the discretion of the Civil Department, the option course may be taken as a Technical Elective course (i.e., no increase above the standard course load for the term)*

Technical Core Courses Outlines

GenE 240 Algorithms and Data Structures

3C, 1T, 1.5L 0.5 W, S

Calendar Description:

Algorithms and Data Structures emphasizes the following topics: structured software design data structures, abstract data types, recursive algorithms, algorithm analysis and design, sorting and searching, hashing, and problem-solving strategies.

Anti-Requisite: E&CE 250, SyDe 324, CS 241

This course will be offered for the first time in Winter 2001

Detailed Course Content:

1. Review of specific concepts in the language to be used in the projects (2 hours)
2. Data representations; data types and data structures; hardware versus software; linear versus linked storage models; basic array and linked structures (3 hours)
3. Abstract Data Types. Object-oriented programming and inheritance, introduction to all of the types to be discussed during the term: arrays, stacks, queues, priority queues, lists, trees, graphs and tables. (2 hours)
4. Algorithm Analysis. Review Exp, logs, Series, Proof, Recurrence; time-space estimates (3 hours)
5. Data Structure Implementation and performance. The following data structures are as discussed and their performance examined:
   - Arrays (including vectors and matrices) An examination of array characteristics, multidimensional arrays, and sparse matrices. (3 hours).
   - Lists Sequential and linked list implementation; singly and doubly linked lists and generalized lists. (2 hours)
   - Trees Operations on trees, binary trees (2 hours)
   - Priority Queues: Use of heaps to implement priority queues .(2 hours)
   - Graphs Graph implementations, shortest path algorithms, minimum spanning trees and topological sort. (3 hours)
6. Hash Tables: Hashing functions, open and chained collision resolution. (1 hour)
7. Search algorithms and structures Array and tree based search techniques including both internal (binary search trees as well as AVL trees); performance. (3 lectures)
8. Sorting Techniques; basic internal methods (bubble sort, insertion sort and selection sort); enhanced versions (quicksort, heap sort and Shell's sort) (2 hours)
9. Algorithm design techniques Greedy Algorithms, Divide and Conquer, Dynamic programming, Back tracking algorithms are examined. .(3 hours)

The outline has been trimmed to 31 hours. There is some additional room (3 hours) for material deemed important by the committee (i.e. some numerical techniques).

Gen E 241 Computer Structures and Real-Time Systems

3C, 1T, 1.5L 0.5 F, W

Calendar Description:

Introduction to computer organization, basic real-time concepts, process management, interprocess communication and synchronization, memory management, resource management, interrupt handling, concurrent programming, file systems.

Prerequisite: Gen E 240

Anti-Requisite: E&CE 354

This course will be offered for the first time in Fall 2001

Detailed Course Content:

1. Week Digital systems including topics such as: Number Systems, Boolean algebra and functions, Flip-flops, registers, Arithmetic Units (3 hours)
2. Basic computer organization, data path, functional units (e.g., ALU, memory, registers), instruction cycle (4 hours)
3. Stack and stack frame management, interrupts and polling I/O (4 hours)
4. Multiple process environment, critical sections, process synchronization and inter-process communication mechanisms. (6 hours)
5. Pre-emptive and non-pre-emptive processor scheduling, comparison of scheduling strategies. (4 hours)
6. Memory Management: use of relocation registers, comparison of replacement strategies, shared code, protection. (4 hours)
7. Resource Management: sharing and protection of resources, multiple resource scheduling, deadlock detection and prevention strategies. (5 hours)
8. File Systems: data structures of various file systems, naming and protection mechanisms, device-independence, program language access mechanisms. (4 hours)

Total: 34 hours.

Gen E 342 Principles of Software Engineering

3C, 1T, 1.5L 0.5 F, S

Calendar Description:

Software development process; software requirement specification, software design; software testing and quality assurance; software maintenance; software management; case studies.

Prerequisite: Gen E 240 and Gen E 241

Anti-Requisite: E&CE 355, Sy De 221

This course will be offered for the first time in the Summer 2002

Detailed Course Content:

1. Introduction; software lifecycle models; CASE tools and their role; human factors in software development. (4 hours)
2. Software requirement specification. Requirement analysis, specification represenations (formal/informal); function, control, and object-oriented represenations; specification of functional and non-functional requirements; requirements validation. (6 hours)
3. Software design. Models of design process; design representations; function, control and object-oriented design; design criteria. (6 hours)
4. Software testing and quality assurance. Testing strategies; defect testing; residual defect/reliability estimation; statistical testing; quality assurance processes; software metrics. (6 hours)
5. Software maintenance. Maintenance categories and activities; maintainability indices; software reuse; reverse engineering and re-engineering. (4 hours)
6. Software management. Software project planning, cost estimation, staffing, monitoring; software product configuration management. (8 hours)

Total: 34 hours

Course project: specify, implement, demonstrate operation of a small embedded software system.