

EECS 1011: Computational Thinking Through Procedural Programming and Mechatronics

Department of Electrical Engineering and Computer Science

Lassonde School of Engineering

York University

Instructor: Dr. James Andrew Smith, P.Eng.

Email: drsmith@yorku.ca

Class email: drsmith.yorku@outlook.com

Brief Course Description

The objectives of 1011 are threefold: providing a first exposure to procedural programming, teaching students a set of computing skills (such as reasoning about algorithms, debugging and hardware interfacing and interaction), and demonstrating how computers are used in a variety of engineering disciplines. It uses problem-based pedagogy to expose the underlying concepts and experiential laboratory activities and projects to implement them. An integrated computing environment (such as MATLAB) is used so that students can learn key programming concepts (such as variables and control flow) without being exposed to complex or abstract constructs. The problems are chosen with consultation with the various engineering disciplines in the Faculty with a view of exposing how computing is used in these disciplines. In support of program-related objectives, WHMIS I and Active Bystander training will take place in this course. Course credit exclusions: LE/EECS 1541 3.00 and LE/ENG 1112 3.00.

Expanded Course Description

The objectives of 1011 are threefold: providing a first exposure to procedural programming, teaching students a set of computing skills (such as reasoning about algorithms, debugging and hardware interfacing and interaction), and demonstrating how computers are used in a variety of engineering disciplines. It uses problem-based pedagogy to expose the underlying concepts and experiential laboratory activities and projects to implement

them. An integrated computing environment (such as MATLAB) is used so that students can learn key programming concepts (such as variables and control flow) without being exposed to complex or abstract constructs. The problems are chosen with consultation with the various engineering disciplines in the Faculty with a view of exposing how computing is used in these disciplines. The course is supplemented with a three-hour weekly lab.

Main Topics

1. The Computing Environment: workspace, built-in commands, the debugger, plots, etc.
2. Variables and Expressions: types, operators, precedence, roundoff errors
3. Control Structures: selection and iteration; combinations in simple state machines
4. Encapsulation: scripts and functions
5. Computational Thinking: reasoning about algorithms, problem solving, unit tests and debugging
6. Hardware interfacing related to sensing and actuation

Applications

1. General Science and Math
2. Engineering Applications derived from the various engineering programs in the Faculty.

Learning outcomes for this course include:

1. Apply a soft computing skill such as reasoning about algorithms, problem solving, unit tests or debugging
2. Explain and apply the fundamental constructs in procedural programming, including variables and expressions, control structures (conditionals/loops), and documentation
3. Write simple programs using a contemporary integrated development environment
4. Use the computing environment to implement/simulate selected applications from science, math, and engineering
5. Programmatically plot data from a computer-connected sensor
6. Demonstrate WHMIS I competency
7. Interface sensor and/or actuation hardware, with any required modifications, to a computer
8. Express an ethical dilemma or conflict in the context of active bystanding

Lab Kit

The EECS 1011 / 1021 lab kit is available in the [York University Bookstore](#). It includes all technical parts needed for the labs and project, except for a multimeter. The multimeter can be obtained from the Bookstore or other sources. The lab kit includes the "crib sheets" for both Matlab and Java. The Bookstore is in charge of pricing for the kit. Refer to the Bookstore signage for costs. Kits will be available in two batches, first on Sept. 9th and then on Sept. 16th. You can purchase them either after the 9th or after the 16th. We will start using the kits after the end of September. That said, each kit contains crib sheets for the course, so they're worth getting as early as you can manage. Also, make sure to get the Fall 2025 version of the kit. Other versions contain different hardware.

Textbook

Textbook is available (for free) in the Library: [MATLAB : a practical introduction to programming and problem solving](#) by [Stormy Attaway](#).

Dr. Smith's Blog

Important material about lab kits, Matlab etc. are posted on my blog: <https://www.yorku.ca/professor/drsmith/> You can look for specific topics like "[MATLAB](#)" or "[Arduino](#)" there.

Minimum Technology Requirements

Contemporary Engineering practice and education both require the use of modern computers. As such, students are required to have their own computer hardware for this course. This requirement was approved by the Lassonde Faculty Council. This means that you need to have a contemporary laptop or tablet capable of running Matlab and interfacing with the lab kit and you must have it available for classroom, laboratory and project activities. This is not optional. Here are [computer suggestions](#).

Matlab Software

You must [create a Matlab User Account](#) to obtain the software, which you must install on your personal computer. The cost of the licence for Matlab is covered by the Lassonde School, so its cost is indirectly paid for by you. You can [download it for free](#) from The Mathworks. You can [read through the download instructions and system requirements](#) if you wish.

Grading Scheme

Grade	Grade Point	Percent Range
A+	9	[90, 100]
A	8	[80, 90)
B+	7	[75, 80)
B	6	[70, 75)
C+	5	[65, 70)
C	4	[60, 65)
D+	3	[55, 60)
D	2	[50, 55)
E	1	[49, 50)
F	0	[0, 49)

How do you read the brackets? Square bracket means inclusive. Parenthesis means not inclusive. So "50)" means "up to 50, but not including 50", whereas "[50" means "50 and above".

Key dates and deliverables

Important dates are listed [here](#). Class starts on Sept 3 and last day is December 2, 2025. Reading week is Oct 11-17, 2025.

- Midterm exam 1 (10%): date to be determined
- Midterm exam 2 (10%): date to be determined
- Final exam(s) (45%): during December exam period (may be composed of up to two separate tests)
- Final Project (15%)
- Labs (15%): demos live in lab. Any reports due by Sunday night of that week at 11:59pm.
- In-Class homework (5%). Flipped class style.

Grades are not "rounded up" or "curved". The previously used "proficiency grading" method is no longer applied.

The tentative schedule for classes, labs, assessments and project is shown here:

	Section E			Section H			Lab or Project	Report Due		Note 1		Note 2			
	1st class	2nd class		1st class	2nd class			(only if there's a report)							
Week 1	03-Sep-25	08-Sep-25		09-Sep-25	09-Sep-25		A	14-Sep-25							
Week 2	10-Sep-25	15-Sep-25		16-Sep-25	16-Sep-25		B	21-Sep-25							
Week 3	17-Sep-25	22-Sep-25		23-Sep-25	23-Sep-25		C	28-Sep-25							
Week 4	24-Sep-25	29-Sep-25		30-Sep-25	30-Sep-25		D	05-Oct-25							
Week 5	01-Oct-25	06-Oct-25		07-Oct-25	07-Oct-25		E	12-Oct-25		Midterm 1 (6-8 Oct)		Half of class takes it on 1st class, half on 2nd			
Week 6	08-Oct-25	20-Oct-25		21-Oct-25	21-Oct-25		F	26-Oct-25				Dates straddle Reading Week			
Week 7	22-Oct-25	27-Oct-25		28-Oct-25	28-Oct-25		G	02-Nov-25							
Week 8	29-Oct-25	03-Nov-25		04-Nov-25	04-Nov-25		H	09-Nov-25							
Week 9	05-Nov-25	10-Nov-25		11-Nov-25	11-Nov-25		I	16-Nov-25		Midterm 2 (10-12 Nov)		Half of class takes it on 1st class, half on 2nd			
Week 10	12-Nov-25	17-Nov-25		18-Nov-25	18-Nov-25		J	23-Nov-25							
Week 11	19-Nov-25	24-Nov-25		25-Nov-25	25-Nov-25		K	30-Nov-25							
Week 12	26-Nov-25	01-Dec-25		02-Dec-25	02-Dec-25		Project	02-Dec-25				No Lab			
Exam Period	-----	-----		-----	-----		-----			Final exam during exam period					

Please review and immediately bring up any important scheduling conflicts with Professor Smith. This includes any religious holidays that may overlap with the timetable shown. Such discussions must happen early in the semester. There is no obligation to consider scheduling conflicts unless they are brought up in a timely manner.

Academic Integrity

Except where explicitly told by the instructor (for example, during a group component of a two-stage exam, if applicable), during tests, quizzes or exams, students are expected to do their own work. In those contexts, looking at someone else's work, talking, using unauthorized aids (such as, but not limited to, a phone, chatbot or contract service), and impersonation are all examples of academically dishonest behaviour. As this is an important aspect of the university experience, students are expected to read and understand the York University Code of Student Rights and Responsibilities and its Senate Policy on Academic Conduct:

- <https://www.yorku.ca/secretariat/policies/policies/academic-conduct-policy-and-procedures/>
- <https://students.yorku.ca/oscr/the-code>

In short: do the work yourself, whether it is in person or online. While it is expected that you will be discussing elements of the course activities with others, in the class or outside the class, simply copying someone else's answer is not acceptable. **A good strategy** is to exchange ideas in generalized terms or related, parallel examples and then to then apply the shared knowledge to the specific example in the assessment.

ACADEMIC INTEGRITY LINKS

- Academic Integrity - <http://lassonde.yorku.ca/academic-integrity>

STUDENT LINKS

- Student Rights and Responsibilities - <https://students.yorku.ca/oscr/the-code>
- Religious Observance - <https://w2prod.sis.yorku.ca/Apps/WebObjects/cdm.woa/wa/regobs>
- Academic Accommodation for Students with Disabilities - <https://students.yorku.ca/accessibility>
- York University Centre for Human Rights, Equity & Inclusion: <https://www.yorku.ca/services/rights/>
- York University's Centre for Sexual Violence Response, Support & Education: <https://thecentre.yorku.ca>