1. **Course number and name**

EECE231 Introduction to Programming Using C++ and MATLAB

1. **Credits and contact hours**

3 Credits, 3 hours/week lectures + 2 hours/week lab

1. **Instructor’s or course coordinator’s name**

Course Coordinator: Fadi Zaraket

1. **Text book: title, author, and year**

Alex. Bielajew, Introduction to Computers and Programming using C++ and MATLAB, University of Michigan, 2004, <http://www-personal.umich.edu/~paulko/book.pdf>

* 1. **Other supplemental materials**

Handouts, lab assignments and problem sets are posted on Moodle

1. **Specific course information**
	1. **Catalog description**

An introductory course for non- ECE students on the principles of programming using C++ and MATLAB. Basic data types, control structures, and arrays will be covered in C++. Algorithms, functions, and arrays will be covered in MATLAB. In addition, the course will expose students to the MATLAB environment and toolboxes with applications in Engineering. Weekly laboratory assignments are an integral part of this course.

* 1. **Prerequisites or co-requisites**

Prerequisite: none.

* 1. **Required, elective, or selected elective course in the program**

Required for Civil engineers, is not considered equivalent to EECE 230.

1. **Specific goals for the course**
	1. **Course learning outcomes**

At the end of the course, students will be able to**[[1]](#footnote-1)**:

* Can edit, compile, and run a program using C++ and MATLAB
* Know the various built-in data types in C++ and MATLAB
* Know how to input/output data
* Know how to use control structures (for, while, if/else…).
* Know how to modularize a program through the use of functions.
* Know how to write functions with call by value and call by reference.
* Know how to write recursive functions.
* Can use arrays in programs and functions.
* Have learned and used the string manipulation functions.
* Become Familiar with different MATLAB computational and plotting Libraries and toolboxes
* Have gained hands-on experience in C++ and MATLAB programming through weekly lab assignments
* Know how to apply the principles of computer programming to the solution of engineering problems.
	1. **Mapping to KPIs**

|  |  |  |  |
| --- | --- | --- | --- |
| **KPI** | **Competence introduced or reinforced in this course** | **Students demonstrate this competence in the course through homework, projects, exams, etc.** | **Students are given formal feedback (oral or written) on their performance on this competence** |
| **a.1** | X | X | X |
| **a.2** | X | X | X |
| **b.1** |  |  |  |
| **b.2** | X | X | X |
| **b.3** | X | X | X |
| **c.1** |  |  |  |
| **c.2** |  |  |  |
| **c.3** | X | X | X |
| **d.1** |  |  |  |
| **d.2** |  |  |  |
| **e.1** |  | X |  |
| **e.2** | X | X | X |
| **e.3** | X | X | X |
| **f.1** |  | X | X |
| **f.2** |  | X | X |
| **g.1** |  | X |  |
| **g.2** |  | X | X |
| **h.1** | X |  |  |
| **h.2** | X |  |  |
| **h.3** |  |  |  |
| **h.4** |  |  |  |
| **i.1** | X |  |  |
| **i.2** | X | X | X |
| **j.1** | X | X | X |
| **k.1** | X | X | X |
| **k.2** | X | X | X |
| **k.3** | X | X | X |

1. **Brief list of topics covered**
2. Basics: introduction, basic elements of C++, Primitive data types
3. Control structures: Boolean expression, selection (if-else, switch/case) and repetition (while/for)
4. Arrays and programming examples
5. Matlab: introduction, style, multi-dimensional dynamic arrays
6. Functions: recursion, libraries, file read and write, simple parsing (MATLAB)
7. Graphics: 2D plots, 3D plots, contours, meshes
8. Practical Applications: statistical parameters, area under a curve, interpolating within a set of data, fitting a line or a power equation to a set of data, solution of a first order differential equation using 1st and 2nd order schemes
9. Symbolic Computations: introduction, plotting
1. The learning outcomes are partially quoted from [1] ``Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering'', A Report in the Computing Curricula Series, Joint Task Force on Computer Engineering Curricula: IEEE Computer Society, Association for Computing Machinery, December 12, 2004 [↑](#footnote-ref-1)