Department of Ocean and Mechanical Engineering
Florida Atlantic University
Course Syllabus

1. Course title/number, number of credit hours	
EOC 4133 – Introduction to Programming for Ocean and Mechanical Engineers	3 credit hours
2. Course prerequisites, corequisites, and where the course fits in the program of study	
Prerequisite: Calculus II (MAC 2312)	
3. Course logistics	
4. Instructor contact information	
Instructor's name *Office address* *Office Hours* *Contact telephone number* *Email address*	
5. TA contact information	
TA's name *Office address* *Office Hours* *Contact telephone number* *Email address*	
6. Course description	
This course will introduce the fundamentals of programming required for Ocean and Mechanical Engineering applications and data analysis using MATLAB and C/C++. The material will cover basic concepts such as arrays and vectors, strings, 'for' and 'while' loops, conditional 'if-else' statements, Boolean logic, and writing modular user-defined functions. The emphasis will be on Ocean and Mechanical Engineering applications, which will involve solving systems of equations, analyzing time-series data, manipulating images, and advanced visualization and plotting of vibration data, tide charts, and time-varying flow fields.	
7. Course objectives/student learning outcomes/program outcomes	
Course objectives	The objective of the course is to provide the students with a basic introduction to programming for engineering applications:

	1. The emphasis will be on learning how to 'translate' engineering problem statements into workable programs, using a combination of flowcharts and pseudocode. 2. The students will become proficient in constructing complex programs using basic 'for' and 'while' loops, 'if-else' conditionals, and user-defined functions. 3. The students will learn how to use debuggers to find low-level programming mistakes effectively. 4. The students will learn advanced 2D and 3D plotting techniques for visualizing and analyzing complex engineering datasets, such as vibration time-series data, tide charts, time-varying flow fields, etc. 5. The students will learn the differences between high-level interpreted languages such as Matlab, and low-level compiled languages such as C/C++.
Student learning outcomes & relationship to ABET 1-7 objectives	1. Students should be able to write simple engineering programs in Matlab and C/C++ using 'for' and 'while' loops, 'if-else' statements, matrices and arrays, and user-defined functions. (1,2,6) 2. Students should be able to document their programs using pseudocode and in-program comments, and they should be able to write an interactive program. (1,2,6) 3. Students should be able to create 2D and 3D plots in Matlab, to visualize a variety of different engineering datasets. (1,2,3,6) 4. Students should be able to present results from their programs in a neat and clear manner, including the use of graphs or tables wherever appropriate. (3)

8. Course evaluation method

Homework	40%
Project	20%
Midterm (Coding-based)	20%
Final (Coding-based)	30%

9. Course grading scale

A	**> 95%**	**C+**	**70 – 74.9**
A-	**90 – 94.9**	**C**	**65 – 69.9**
B+	**85 – 89.9**	**C-**	**60 – 64.9**
B	**80 – 84.9**	**D**	**50 – 59.9**
B-	**75 – 79.9**	**F**	**< 50**

In case the final class average is lower than a 'B-', all grades will be adjusted upward.

10. Policy on makeup tests, late work, and incompletes

Late work will not be accepted unless there is solid evidence of a medical or otherwise serious emergency that prevented the student from completing the assignments on time. Incomplete grades are against the policy of the department. Unless there is solid evidence of medical or otherwise serious emergency situation, incomplete grades will not be given.

11. Special course requirements

This course contains no special requirements.

12. Classroom etiquette policy

University policy requires that in order to enhance and maintain a productive atmosphere for education, personal communication devices, such as cellular phones and laptops, are to be disabled in class sessions.

13. Disability policy statement

In compliance with the Americans with Disabilities Act Amendments Act (ADAAA), students who require reasonable accommodations due to a disability to properly execute coursework must register with Student Accessibility Services (SAS)—in Boca Raton, SU 133 (561-297-3880); in Davie, LA 203 (954-236-1222); or in Jupiter, SR 110 (561-799-8585) —and follow all SAS procedures.

14. Honor code policy

Students at Florida Atlantic University are expected to maintain the highest ethical standards. Academic dishonesty is considered a serious breach of these ethical standards, because it interferes with the university mission to provide a high quality education in which no student enjoys unfair advantage over any other. Academic dishonesty is also destructive of the university community, which is grounded in a system of mutual trust and place high value on personal integrity and individual responsibility. Harsh penalties are associated with academic dishonesty. See University Regulation 4.001 at www.fau.edu/regulations/chapter4/4.001_Code_of_Academic_Integrity.pdf

15. Counseling and Psychological Services Center

Life as a university student can be challenging physically, mentally and emotionally. Students who find stress negatively affecting their ability to achieve academic or personal goals may wish to consider utilizing FAU's Counseling and Psychological Services (CAPS) Center. CAPS provides FAU students a range of services – individual counseling, support meetings, and psychiatric services, to name a few – offered to help improve and maintain emotional well-being. For more information, go to http://www.fau.edu/counseling/

16. Required texts/reading

No required textbook.

17. Supplementary/recommended readings

1.) Mathworks documentation and tutorials:
https://www.mathworks.com/learn/tutorials/matlab-onramp.html?s_cid=learn_ONRAMP_BAN
https://www.mathworks.com/help/matlab/index.html

2.) Matlab by example: programming basics, by Munther Gdeisat, Francis Lilley
(Available as free e-book from the FAU Library)

3.) Matlab: An introduction with applications, by Rao V. Dukkipati
(Available as free e-book from the FAU Library)

4.) Matlab for Newbies: The bare essentials, by Siddhartha Verma
(**Available as free e-book from the FAU Library)**

5.) Introduction to C++ Programming and Graphics, by Constantine Pozrikidis
(**Available as free e-book from the FAU Library)**

18. Course topical outline, including dates for exams, papers, completion of reading

Topics:

1. Translating engineering problems into programs using flowcharts and pseudocode
2. 'Vectors' or 'arrays' in Matlab – 1D, 2D, … 'n-D' arrays
3. Mathematical operations - sum, product, matrix multiplication, vectorized operations vs. element-wise operations, Boolean operators
4. 'for' and 'while' loops, 'if-else' and 'switch' conditionals
5. Using the built-in debugger in Matlab
6. Built-in commands – 'ones', 'zeros', 'linspace', 'meshgrid', 'min/max', 'find', 'sort', etc.
7. Writing modular programs using user-defined functions
8. Importing/Exporting numerical data (with/without headers) and images from files on disk
9. Data visualization - line plots, contour plots, mesh/surface plots, quiver plots, volume viz., etc.
10. Engineering applications – numerical quadrature, solving systems of equations, loading and analyzing vibration time-series data, plotting tide charts, visualizing time-varying flow fields, etc.
11. C/C++ - Data types (int, float, double, char, string), precision, variable declaration, initialization, compilation
12. C/C++ - Loops, conditionals, and Boolean operators
13. C/C++ - Debugging with 'gdb' or a similar program
14. C/C++ - Writing user-defined functions, modular programming with 'header files', 'std' libraries
15. C/C++ - Pointers, passing 'by value' vs passing 'by reference', const parameters
16. C/C++ - Single- and multi-dimensional arrays and vectors, importing/exporting data (printf, scanf, fopen, fprintf, fscanf, fclose), parsing and manipulating strings and characters
17. C/C++ - Binary digits, bit manipulation and bitwise operations
18. C/C++ - Solving systems of equations by linking to scientific libraries (GSL)