Lecture Contact Hours: 32-36; Outside-of-Class Hours: 64-72;
Laboratory Contact Hours: 48-54; Outside-of-Class Hours: 0;
Total Student Learning Hours: 144-162

CUYAMACA COLLEGE
COURSE OUTLINE OF RECORD

Engineering 120 – Engineering Computer Applications

2 hours lecture, 2 units
3 hours laboratory, 1 unit
Total units: 3

Catalog Description
Use of computerized mathematical analysis, computer programming, and computer graphics as tools for solving engineering problems.

Prerequisite
"C" grade or higher or "Pass" in MATH C2210 (formerly MATH 180) or equivalent or concurrent enrollment

Entrance Skills
Without the following skills, competencies and/or knowledge, students entering this course will be highly unlikely to succeed:
1) Essential concepts in differentiation and integration.
2) Evaluating derivatives of trigonometric functions.
3) Modeling and applications related to rates, relative extrema, area between curves, surface area and volumes.

Course Content
The following topics will be taught using the MATLAB programming language:
1) Computer program design, top-down design techniques, use of pseudocode for program development
2) Constants, variables and arrays
3) Additional data types: double-precision, complex data, character strings
4) Relational and logical operators
5) *If* structures for branching
6) Loops: *While* Loops, *Do* Loops and *For* Loops. Nested loop structures and recursive operations
7) Input/output functions
8) Reading and writing data files
9) Multi-dimensional array processing
10) Callable functions, user-defined functions, passing variables
11) Basic computer graphics: simple xy plots, multiple plots, line control
12) Advanced computer graphics: multi-dimensional data and advanced graphical control
13) Development of graphical user interfaces
14) Numerical techniques: systems of linear equations, numerical integration and differentiation, interpolation and curve fitting
15) Modeling real world phenomena using computers. Solutions of various engineering problems
16) Use of computers for data exploration and presentation

Course Objectives
Students will be able to use MATLAB to:

1) Design and write efficient computer programs using top-down design techniques and pseudocode for program development.
2) Apply various data types including single, double, integer, complex, and logical in conjunction with constants, variables and multi-dimensional arrays in the computer analysis of engineering problems.
3) Apply relational and logical operators in conjunction with branching structures.
4) Apply appropriate loop structures including nested loop structures and recursive operations.
5) Apply input and output functions, formatted I/O, and communication with data files.
6) Apply general intrinsic functions, and design and write callable functions.
7) Apply basic computer graphics techniques to produce simple xy plots, multiple plots, and simple enhanced control of plotted lines.
8) Apply advanced computer graphics to the display of multi-dimensional data and images and use of advanced graphical controls.
9) Design and write computer programs to solve engineering problems using numerical techniques and/or intrinsic functions including: polynomial operations, statistical operations, matrix operations, symbolic operations, curve fitting, linear interpolation, and integration and differentiation.

Method of Evaluation

A grading system will be established by the instructor and implemented uniformly. Grades will be based on demonstrated proficiency in subject matter determined by multiple measurements for evaluation, one of which must be essay exams, skills demonstration or, where appropriate, the symbol system.

1) Programming assignments that demonstrate students' ability to apply knowledge gained in class to using MATLAB to solve engineering problems. An example would be to apply knowledge of matrix computations to develop a program which reads the data of a system of linear equations in 1200 variables from a data file, solves the system using two methods, and prints the results in tabulated form.
2) Homework that provides students with an opportunity to interact with the course material and evaluate their ability to extend the classroom and reading experience to novel situations. Questions are primarily word problems. An example would be to write a MATLAB program that studies the distance traveled by a ball thrown with a specified initial velocity to determine the angle that maximizes the range of the ball, then represent the result both graphically and numerically.
3) Periodic (e.g., midterm and final) projects that demonstrate students' ability to integrate knowledge gained in class and apply it resourcefully. An example would be to define the input/output, design the algorithm, write the pseudocode, and generate an interactive MATLAB program that simulates the position of the sun as it moves across the sky during one day for user-inputs latitude and month. The program must validate the input data and generate the solar altitude and azimuth angles versus time in tabulated form and graphically, between sunrise and sunset.
4) Periodic quizzes, midterm exams, and final examination to evaluate student learning and retention of the material on the time scale of weeks to months. Questions are mostly word problems but with some short-answer conceptual questions. Students may be asked to write short MATLAB programs to provide either graphical or numerical support for their theoretical answers. An example would be a problem in which students find the first derivative of a given equation of heat loss from a round insulated pipe with respect to the insulation thickness, and use the result to determine the theoretical "critical insulation thickness." This theoretical result must then be verified by writing a short program.

Special Materials Required of Student

Graphing calculator

Minimum Instructional Facilities

Computer lab

Method of Instruction
Lecture and laboratory assignments

Out-of-Class Assignments
1) **Reading**: Lecture notes; assigned sections from the textbook; supplemental problem explanations; online MATLAB documentation (e.g. tool guides, help guides); instructor-provided conceptual summaries and study guides. Students may also read articles or excerpts demonstrating real-world applications of the material.
2) **Writing**: Homework that provides students with an opportunity to interact with the course material and evaluate their ability to extend the classroom to novel situations. An example would be to write a MATLAB program that studies the distance traveled by a ball thrown with a specified initial velocity to determine the angle that maximizes the range of the ball, then represent the result both graphically and numerically.
3) **Other**: Programming assignments that demonstrate students' ability to apply knowledge gained in class to using MATLAB to solve engineering problems. An example would be to apply knowledge of matrix computations to develop a program which reads the data of a system of linear equations in 1200 variables from a data file, solves the system using two methods, and prints the results in tabulated form.

Texts and References
1) Required (representative examples):
   a. *MATLAB Student Version*. Mathworks, 2023.
   b. Chapman, Stephen J. *MATLAB Programming for Engineers*. 7th edition, 2024.
   c. Siauw, Timmy, and Bayen, Alexandre. *An Introduction to MATLAB Programming and Numerical Methods for Engineers*, 2015.
2) Supplemental: None

Student Learning Outcomes
Upon successful completion of this course, students will be able to:
1) Design and write efficient computer programs in a high-level engineering programming language such as MATLAB using top-down design techniques and pseudocode for program development.
2) Write code that makes use of all appropriate programming structures such as functions, branching statements, loops, recursion, formatted I/O (including file operations) for a given problem.
3) Solve engineering problems using numerical methods including all the following: Linear algebra, least squares regression, linear and cubic interpolation, Taylor Series, root finding, numerical differentiation, and numerical integration.