College of San Mateo
Official Course Outline

1. **COURSE ID:** ENGR 215 **TITLE:** Computational Methods for Engineers and Scientists
   Units: 3.0 units **Hours/Semester:** 32.0-36.0 Lecture hours; 48.0-54.0 Lab hours; and 64.0-72.0 Homework hours
   Method of Grading: Letter Grade Only
   Prerequisite: MATH 251

2. **COURSE DESIGNATION:**
   Degree Credit
   Transfer credit: CSU; UC

3. **COURSE DESCRIPTIONS:**
   Catalog Description:
   Introduction to problem solving, programming, and computational methods using the MATLAB programming environment. Procedural programming, recursion, sorting, object-oriented representations, and data structures. Plotting and data visualization, introduction to statistical analysis of data, systems of linear equations, numerical methods. Applications in engineering, mathematics, and the sciences. Assignments may require the use of MATLAB software outside of class hours.

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**
   Upon successful completion of this course, a student will meet the following outcomes:
   1. Use MATLAB as a computational tool to solve problems in engineering, mathematics, and the sciences.
   2. Use MATLAB as a plotting and visualization tool.
   3. Design, implement, and test procedural computer programs to achieve stated objectives.
   4. Design, implement, and test elementary object oriented programs to achieve stated objectives.
   5. Begin to use standard data structures.
   6. Working from a verbal or equation-based problem description, set up systems of simultaneous equations and use MATLAB to determine existence and uniqueness of solution and solve.
   7. Working from a verbal or equation-based problem description, use appropriate numerical techniques for differentiation, integration, and solving differential equations.
   8. Apply numeric techniques and computer simulations to analyze and solve problems in engineering, mathematics, and the sciences.

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**
   Upon successful completion of this course, a student will be able to:
   1. Use MATLAB as a computational tool to solve problems in engineering, mathematics, and the sciences.
   2. Use MATLAB as a plotting and visualization tool.
   3. Design, implement, and test procedural computer programs to achieve stated objectives.
   4. Design, implement, and test elementary object oriented programs to achieve stated objectives.
   5. Begin to use standard data structures.
   6. Working from a verbal or equation-based problem description, set up systems of simultaneous equations and use MATLAB to determine existence and uniqueness of solution and solve.
   7. Working from a verbal or equation-based problem description, use appropriate numerical techniques for differentiation, integration, and solving differential equations.
   8. Apply numerical techniques and computer simulations to analyze and solve problems in engineering, mathematics, and the sciences.

6. **COURSE CONTENT:**
   Lecture Content:
   1. Problem solving in engineering and science
   2. Introduction to computers, programming, and the MATLAB environment
   3. Array and matrix basics
   4. Variables, expressions, and order of operation
   5. Documentation: pseudocode, flow charts, etc.
   6. Formatted input and output
   7. Execution control: selection and repetition
   8. Functions (built-in and user-defined)

9. Recursion
10. Sorting and searching
11. Plotting and Visualization
12. Applications of MATLAB functions and programming techniques to
    A. Systems of linear algebraic equations
    B. Numerical methods (curve fitting, root-finding, integration, differentiation, solution of differential equations)
    C. Optimization (optional)
    D. Stochastic simulation (optional)
13. Introduction to data structures
14. Classes and object oriented programming
15. Optional additional topics: graphical user interfaces, image processing, audio processing,

Lab Content:

Laboratory activities require students to use MATLAB as a computational tool and programming environment to explore and apply concepts covered in the lecture portion of the course. Activities are based on applications in engineering, mathematics, and the sciences.

7. **REPRESENTATIVE METHODS OF INSTRUCTION:**

Typical methods of instruction may include:

A. Lecture
B. Lab
C. Other (Specify): Lectures to introduce new material and topics. Instructor demonstration of software. Textbook reading assignments to expand knowledge. Individual lab and take-home computational and problems to develop skills. Group lab assignments (optional) Individual or group term programming project to integrate knowledge.

8. **REPRESENTATIVE ASSIGNMENTS**

Representative assignments in this course may include, but are not limited to the following:

Writing Assignments:

Term project report (optional).

Reading Assignments:

Textbook reading assignments to expand knowledge.

Other Outside Assignments:

Computational and programming problems relevant to course topics and implemented in MATLAB.

9. **REPRESENTATIVE METHODS OF EVALUATION**

Representative methods of evaluation may include:

A. Exams/Tests
B. Homework
C. Lab Activities
D. Individual problem-solving assignments. Individual programming assignments. Group assignments (optional). Individual or group project. Individual exams.

10. **REPRESENTATIVE TEXT(S):**

Possible textbooks include:

A. Smith, D. M.. *Engineering Computation with MATLAB* , 3rd ed. Pearson Addison Wesley, 2012
B. Siauw, T. , A. M. Bayen. *An Introduction to MATLAB Programming and Numerical Methods for Engineers*, 1st ed. Boston: Academic Press, 2015
C. Palm, W.J.. *A Concise Introduction to MATLAB*, 1st ed. Boston: McGraw-Hill, 2008

Origination Date: October 2016
Curriculum Committee Approval Date: December 2016
Effective Term: Fall 2017
Course Originator: Laura Demsetz