COURSE OUTLINE

ENGS 102 – PROGRAMMING FOR ENGINEERS

Prepared By: Michael J. Newtown, P.E.
Revised By: David Wells
Revised By: Cullen Haskins (Fall 2015)
A. **TITLE:** Programming for Engineers

B. **COURSE NUMBER:** ENGS 102

C. **CREDIT HOURS:** 2

D. **WRITING INTENSIVE COURSE:** N/A

E. **COURSE LENGTH:** 15 weeks

F. **SEMESTER(S) OFFERED:** Fall/Spring

G. **HOURS OF LECTURE, LABORATORY, RECITATION, TUTORIAL, ACTIVITY:**
   2 hours recitation in computer lab.

H. **CATALOG DESCRIPTION:**
   This course provides the software skills necessary to create predictive models and solve basic engineering problems. Student will learn to make statistical inferences about the data while creating graphical presentation of the results using engineering related software. The skills taught in this course will assist in the analysis of engineering problems in more advanced course work.

I. **PRE-REQUISITES/CO-REQUISITES:** None

J. **GOALS (STUDENT LEARNING OUTCOMES):**
   By the end of this course, the student will be able to:

<table>
<thead>
<tr>
<th>Course Objective</th>
<th>Institutional SLO</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Develop simple predictive models using equation modeling software</td>
<td>2. Crit. Thinking</td>
</tr>
<tr>
<td></td>
<td>3. Prof. Competence</td>
</tr>
<tr>
<td>b. Duplicate complex code to solve differential equations</td>
<td>2. Crit. Thinking</td>
</tr>
<tr>
<td></td>
<td>3. Prof. Competence</td>
</tr>
<tr>
<td>c. Illustrate data through 2D and 3D plots</td>
<td>1. Communication</td>
</tr>
<tr>
<td></td>
<td>3. Prof. Competence</td>
</tr>
<tr>
<td>d. Demonstrate the use of control structures in programming</td>
<td>2. Crit. Thinking</td>
</tr>
<tr>
<td></td>
<td>3. Prof. Competence</td>
</tr>
<tr>
<td>e. Recognize the correct use of string vs. numerical data</td>
<td>2. Crit. Thinking</td>
</tr>
<tr>
<td>f. Apply programming to engineering problems</td>
<td>2. Crit. Thinking</td>
</tr>
<tr>
<td></td>
<td>3. Prof. Competence</td>
</tr>
</tbody>
</table>

K. **TEXTS:** No text required

L. **REFERENCES:**

M. **EQUIPMENT:** Computer Lab

N. **GRADING METHOD:** A-F
O. **MEASUREMENT CRITERIA/METHODS:**

- Homework
- Quizzes
- Projects
- Exams
- Final Project

P. **DETAILED COURSE OUTLINE:** (must use the outline format listed below)

1. Introduction to MATLAB environment
   a. Use of the matlab command screen and mathematical operators
   b. Management of variable and syntax
   c. Annotation requirements for program code

2. Built in functions
   a. Math functions
   b. Trigonometric functions
   c. Data analysis functions
   d. Random / complex numbers
   e. Computational limitations

3. Built in functions
   a. Math functions
   b. Trigonometric functions
   c. Data analysis functions
   d. Random / complex numbers
   e. Computational limitations

4. Matrices / Plotting
   a. Manipulating matrices
   b. Introduction to plotting
   c. Subplots
   d. 2-D plotting
   e. 3-D plotting

5. User defined functions
   a. Creating function m-files
   b. Global variables
   c. Sub-functions

6. Input / Output and reading and writing files
   a. User define input
   b. Output options
   c. Graphical input
   d. Reading and writing data files

7. Repetition structures
   a. For loops
   b. If and else
   c. While Loops
d. Break and continue
e. Switch and case

8. Symbolic mathematics
   a. Symbolic Algebra
   b. Calculus
   c. Differential Equations

9. Numerical Techniques
   a. Interpolation
   b. Curve fitting
   c. Numerical differentiation
   d. Numerical integration

10. Advanced graphics
    a. Scatter plots
    b. Contour plots
    c. Animation

Q. **LABORATORY OUTLINE:** N/A