**SYLLABUS - ENEE 148A** **Programming Elements for Electrical Engineers**

**Credits & Contact Hours:** 3 credits (2 hours lecture, 3 hour laboratory)

**Instructor:** W. Lawson

**Office / Phone:** AVW 2325 / x54972

**Office Hours:** by appointment

**E-mail:**  lawson@umd.edu

**Course Description:** A problem-based introduction to C programming in the context of microcontroller applications. This course provides an introduction to C programming in the context of simple electrical engineering applications and also teaches elements of sensor and actuator operation, communications, control, and circuit theory.

**Pre-Requisites:** none

### Textbooks: [C Programming Language (2nd Edition)](http://www.amazon.com/Programming-Language-2nd-Brian-Kernighan/dp/0131103628/ref=sr_1_1?s=books&ie=UTF8&qid=1382114942&sr=1-1&keywords=introduction+to+c+programming) by [Brian W. Kernighan](http://www.amazon.com/Brian-W.-Kernighan/e/B000AQ1TNQ/ref=sr_ntt_srch_lnk_1?qid=1382114942&sr=1-1) and Dennis M. Ritchie (Apr 1, 1988)

**Other Required Material:** course notes and online videos, data sheets and documents

**Student Learning Outcomes:**

All students who pass this proposed course will have an:

1. appreciation for the enabling role of programmable devices in technological systems and applications.
2. operational familiarity with elementary programming concepts: program flow, data types, arrays and memory, logic and arithmetic operations, and functions.
3. ability to utilize good programming practices to write efficient, clear, and maintainable code.
4. ability to use an IDE to write, debug, load and run code to solve engineering problems and to perform basic calculations, input and output.
5. understanding of the operation of basic electronic components, sensors and actuators.
6. ability to work effectively in teams.
7. ability to communicate effectively in written and oral formats.

**Topics Covered:**

1. C programming:
   1. basics
   2. program flow
   3. data types
   4. logic and arithmetic operations
   5. functions
   6. arrays
   7. input / output
   8. memory
2. IDE interfaces, makefiles, and gcc
3. Sensors, e.g. Photosensors, temperature sensors, proximity sensors, accelerometers
4. Motors and motor controllers, relays
5. Basic elements of circuit theory
6. Basic engineering design elements

**Grading Policy:**

Homework 15%

Quizzes 15% A: 90-100%

Projects/Labs 35% B: 80-89.99%

Midterm 10% C: 70-79.99%

Final Exam 25% D: 60-69.99%

Lab Schedule:

1 Assembly of Raspberry Pi kits / running the IDE

2 Raspberry PI multiple blinks – Morse code *group* lab

3Raspberry PI multiple blinks – Morse code *group* lab

4Light sensors lab

5A/D converter / IR distance sensors *group* lab

6AnalogAccelerometers *group* lab

7DigitalMagnetic sensors lab

8AcousticDistance Sensors *group* Lab

9Servomotors *group* lab

10Mux/demux circuits *optional* lab

11 Group Project – Tank Maze

12 Group Project – Tank Maze

13 Group Project – Tank Maze

14 Group Project – Tank Maze competition