1. Course number and name
   ENGR 213: Introduction to C Programming for Engineers

2. Credits and contact hours
   3 credit hours; two 50-minute lecture session/week

3. Instructor’s or course coordinator’s name
   Instructor: Mohammad Hajiaboli, Lecturer of Electrical and Computer Engineering
   Course coordinator: Thomas Holton, Professor of Electrical and Computer Engineering

4. Text book, title, author, and year
   a. other supplemental materials
      Lecture notes, practice problems, projects on the iLearn website

5. Specific course information
   a. brief description of the content of the course (catalog description)
      Introduction to C programming; defining and analyzing problems; design of algorithms;
      implementation, testing, debugging, maintenance and documentation of programs; coverage of basic
      algorithms, programming concepts and data types; C programming of microcontrollers. Classwork, 1
      unit; laboratory, 1 unit.

   b. prerequisites or co-requisites
      MATH 226 (Calculus I) with a grade of C- or better.

   c. indicate whether a required, elective, or selected elective course in the program
      Required for Computer and Electrical Engineering.

6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current
      research about a particular topic.
      • The student will be able to use an IDE to compile, link and run programs.
      • The student will be familiar with C pragmatics, source organization, header files, and local and external
         code.
      • The student will be familiar with program structure and components.
      • The student will understand data types, identifiers, variables and constants.
      • The student will understand input and output operators.
      • The student will be able to write programs using conditional and repeated execution: if-else, loop, block
         constructs.
      • The student will be able to write programs using functions, arguments, recursion.
      • The student will be able to write programs that use pointers, pointer arithmetic and dereferencing.
      • The student will be able to write programs that use strings and text processing.
      • The student will understand dynamic memory allocation.
b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.
   Course addresses ABET Student Outcome(s): j, k.

7. Brief list of topics to be covered
   - C overview. Application area, efficiency, standardization
   - Your first program. Compiling, linking, running. IDEs
   - Program structure and components
   - Data types, identifiers, variables, constants
   - Input and output
   - Operators, expressions
   - Conditional and repeated execution: if-else, loop, block constructs
   - Basic types and conversions
   - Arrays
   - Functions, arguments, recursion
   - C pragmatics. Source organization, header files, local and external code, linking
   - Pointers, pointer arithmetic, dereferencing
   - Strings and text processing
   - C preprocessor
   - Structures and unions
   - Dynamic memory allocation. Heap