Instructor: Assoc. Prof. Dr. Doğu Arifler
Office Hours: TBA. Please refer to the course Web site.
Course Web Site: http://cmpe.emu.edu.tr/courses/cmpe112

Course Description: An overview of C programming language. Sequential structures, data types and classes of data, arithmetic operators and expressions, assignment statements, type conversions, simple I/O functions (printf, scanf, fprintf, fscanf, gets, puts, fgets, fputs). Selective structures, relational operators, logical operators, conditional expression operator, conditional statements (if, switch). Repetitive structures, while, do-while, for loops, loop interruptions (goto, break, continue). Functions, function definitions and function calls, recursion. Arrays, array declaration, array initialization, arrays as function arguments. Pointers, basics of pointers, functions and pointers arrays and pointers, strings and pointers. Library functions for processing strings, pointer arrays. Structures, basics of structures, structures and functions, arrays of structures.

Pre-requisite: CMPE 101. Credit Hours: (4, 2) 4. Designation: Area Core.


Grading Policy: Midterm 40%, Final 50%, Programming Assignments 10%.

Lab Policy: There are no exemptions from labs. Consult the course Web site for details of lab/programming assignments and other lab policies.

Attendance and Participation: Attendance to every lecture is mandatory. Up to 5 points may be awarded at instructor’s discretion based on your class attendance and participation if your cumulative grade at the end of the semester is ≥ 50%.

Make-Up Policy: Only one comprehensive make-up examination will be given to those who miss any of the exams. There are no make-ups for missed labs. The make-up exam will be given to only those who provide a valid excuse in writing within the next three working days following the missed exam. This rule is a University by-law, and I will enforce it. Students who miss an exam due to a serious medical condition are required to provide official documentation (doctor’s report approved by the Student Health Center). However, eligibility to take the make-up exam will still be subject to my approval.

Academic Dishonesty: Any conduct that attempts to gain unfair academic advantage is considered academic dishonesty. Copying labs and assignments, cheating during exams, substituting for another person are some examples of academic dishonesty. Cases of academic dishonesty will not be tolerated and will be punished according to EMU’s disciplinary policies.
Tentative Outline: Below is a tentative outline for this course. I reserve the right to adjust the pace and topics of the class as the semester progresses.

<table>
<thead>
<tr>
<th>Week 1</th>
<th>Introduction (Ch. 1, 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 2</td>
<td>Formatted I/O; Expressions (Ch. 3, 4)</td>
</tr>
<tr>
<td>Week 3</td>
<td>Selections (Ch. 5)</td>
</tr>
<tr>
<td>Week 4</td>
<td>Loops; Basic types (Ch. 6, 7)</td>
</tr>
<tr>
<td>Week 5</td>
<td>Arrays (Ch. 8)</td>
</tr>
<tr>
<td>Week 6</td>
<td>Functions (Ch. 9)</td>
</tr>
<tr>
<td>Week 7</td>
<td>Program organization (Ch. 10)</td>
</tr>
<tr>
<td>Weeks 8-9</td>
<td>Midterm Week</td>
</tr>
<tr>
<td>Week 10</td>
<td>Pointers (Ch. 11)</td>
</tr>
<tr>
<td>Week 11</td>
<td>More on pointers (Ch. 11)</td>
</tr>
<tr>
<td>Week 12</td>
<td>Pointers and arrays (Ch. 12)</td>
</tr>
<tr>
<td>Week 13</td>
<td>Strings (Ch. 13)</td>
</tr>
<tr>
<td>Week 14</td>
<td>Structures (Ch. 16)</td>
</tr>
<tr>
<td>Week 15</td>
<td>Review and concluding remarks</td>
</tr>
</tbody>
</table>

Reading the textbook is a must for success in this course. Please schedule your reading according to the tentative outline given above.

Topics and related reading assignments from the textbook:

- **Chapter 1**: Introduction (Read: §1.1-1.2)
- **Chapter 2**: Compiling and linking a program, directives, comments, variables and declarations, assignments (Read: §2.1-2.8)
- **Chapter 3**: Formatted I/O (Read: §3.1-3.2)
- **Chapter 4**: Operators, evaluation of expressions (Read: §4.1-4.5)
- **Chapter 5**: Logical expressions, if statement, switch statement (Read: §5.1-5.3)
- **Chapter 6**: while, do, and for statements; break, continue, and goto (Read: §6.1-6.5)
- **Chapter 7**: Types, type conversions (Read: §7.1-7.6)
- **Chapter 8**: Arrays (Read: §8.1-8.2)
- **Chapter 9**: Functions; recursion (Read: §9.1-9.6)
- **Chapter 10**: Local variables, static variables, blocks, scopes (Read: §10.1-10.5)
- **Chapter 11**: Pointers, address and indirection operators (Read: §11.1-11.5)
- **Chapter 12**: Pointers and arrays, pointer arithmetic (Read: §12.1-12.3)
- **Chapter 13**: Strings, C string library (Read: §13.1-13.7)
- **Chapter 16**: Structures (Read: §16.1-16.3)
Course Learning Outcomes (CLOs):
At the end of the course, student must be able to:

1. Design and implement programs in the standard version of C;

2. Understand issues involved in designing, implementing, debugging, and maintaining software;

3. Develop good software engineering skills;

4. Build up the necessary background for advanced courses such as data structures and object-oriented programming.

Contribution of course to meeting the requirements of ABET Criterion 5
The course helps achieving the following components in ABET Criterion-5: B- Engineering Science and Design.

Relationship of course to Program Outcomes
The course supports achievement of the following program objectives: I. identify, formulate and solve computer engineering and science problems through application of an adequate knowledge of mathematics, basic sciences and engineering tools; II. design and implement systems in their particular field; V. apply modern engineering tools and techniques innovatively.

This course is used to assess the following items of Program Outcomes: a) an ability to apply knowledge of mathematics, science, and engineering (CLO 1, 2, 3, 4); c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability (CLO 1, 2, 3); e) an ability to identify, formulate, and solve engineering problems (CLO 1, 2, 3); k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (CLO 1, 2, 3, 4).