|  |  |  |  |
| --- | --- | --- | --- |
| **CMPE 112 Programming Fundamentals** | | | |
| **Department:** Computer Engineering | | | |
| **Instructor Information:**  **Gr 01:** Prof. Dr. Dogu Arifler **E-mail:** [dogu.arifler@emu.edu.tr](mailto:dogu.arifler@emu.edu.tr) **Office:** CMPE 218 **Office Tel:** 1192  **Gr 02:** Prof. Dr. Marifi Güler **E-mail:** [marifi.guler@emu.edu.tr](mailto:marifi.guler@emu.edu.tr) **Office:** CMPE 209 **Office Tel:** 1120 | | | |
| **Assistant Information:** | | | |
| **Meeting Times and Places:**  Gr 01: Mon 08:30-10:30 Lab CMPE 134, Tue 14:30-16:30 Room CMPE 126, Thu 10:30-12:30, Room CMPE 126  Gr 02: Mon 14:30-16:30 Lab CMPE 137, Tue 12:30-14:30 Room CMPE 126, Wed 12:30-14:30 Room CMPE 126 | | | |
| **Program Name:** Computer/Software Engineering | | **Program** **Code:** 25/29 | |
| **Course Code:**  CMPE 112 | **Credits:**  4 | | **Year/Semester:**  2018-2019 Spring |
| Required Course  Elective Course (click on and check the appropriate box) | | | |
| **Prerequisite(s):**  CMPE 101 Foundations of Computer Engineering | | | |
| **Catalog 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. 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. | | | |
| **Course Web Page:** https://staff.emu.edu.tr/doguarifler/en/teaching/cmpe112 | | | |
| **Textbook(s):**  G. J. Bronson, *A First Book of ANSI C*, 4th ed., Course Technology, 2006. | | | |
| **Supplemental Texts:**  K. N. King, *C Programming: A Modern Approach*, 2nd ed., W. W. Norton & Company, 2008.  J. R. Hanly and E. B. Koffman, *Problem Solving and Program Design in C*, 8th ed., Pearson, 2015.  P. Deitel and H. Deitel, *C How to Program*, 8th ed., Pearson, 2015. | | | |
| **Topics Covered and Tentative Class Schedule:**  **(4 hours of lectures per week)**   |  |  | | --- | --- | | **Week 1** | *Introduction, Formatted I/O (Ch. 2)* | | **Week 2** | *Sequential Structures (Ch. 2, 3)* | | **Week 3** | *Selective Structures (Ch. 4)* | | **Week 4** | *Repetitive Structures (Ch. 5)* | | **Week 5** | *Functions (Ch. 6, 7)* | | **Week 6** | *Arrays (Ch. 8)* | | **Week 7** | *Pointers (Ch. 11)* | | **Weeks 8-9** | *MIDTERMS (April 11-22, 2019)* | | **Week 10** | *Pointers (contd.) (Ch. 11)* | | **Week 11** | *Pointers and Arrays (Ch. 11)* | | **Week 12** | *Strings (Ch. 9)* | | **Week 13** | *File I/O (Ch. 10)* | | **Week 14** | *Review* | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lab Schedule:**  *There are no exemptions from labs. Consult the course Web site for details of lab assignments and other lab policies. If you attend at least 7 (out of 8) of the lab experiments and tutorials, you will get full grade. If you attend less, then you will be graded proportional to the number of lab experiments and tutorials.*   * Lab 1 Week of Mar 4 * Lab 2 Week of Mar 11 * Lab 3 Week of Mar 18 * Lab 4 Week of Mar 25 * Lab 5 Week of Apr 29 * Lab 6 Week of May 6 * Lab 7 Week of May 13 * Lab 8 Week of May 20 | | | | |
| **Course Learning Outcomes:**  Upon successful completion of the course, students are expected to have the following competencies:   1. Ability to write a complete C program for solving a problem 2. Ability to use the MS-Visual Studio IDE to edit, compile, and executing C codes 3. Understand problem solving concept using the computer and ability to construct an algorithm and /or flowchart for solving a problem 4. Understand the Basics of C high level programming languages 5. Ability to use if-statement and switch statement to implement selective structure programs 6. Ability to use while-loop, do-while loop, and for-loop to construct repetitive structure 7. Ability to use modular programming for implementing multi-task problem 8. Ability to use arrays concept in C programming 9. Ability to use pointers in C programming 10. Ability to use strings in C programming 11. Ability to use structures in C programming | | | | |
| **Assessment** | **Method** | **No** | | **Percentage** |
| Midterm Exam | 1 | | 40% |
| Final Exam | 1 | | 45% |
| Labs | 8 | | 15% |
| **Attendance and Participation:** Attendance to every lecture is mandatory. | | | | |
| **NG Policy:** Receiving zero from or missing any of the components (midterm, final, labs) used in determination of the letter grade may result in an NG if the accumulated total mark in the course is <50%. | | | | |
| **Make-Up Policy:** Only one **comprehensive** make-up examination will be given for a missed midterm or final **only under exceptional/extenuating circumstances** (e.g., hospitalization, loss of a close relative, etc.). In these cases, students must submit a petition with related official reports to the instructor within the next three working days following the missed exam. Note that minor ailments are not considered as exceptional/extenuating circumstances. Eligibility to take the make-up exam **will be subject to the course coordinator’s final 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. | | | | |
| **Relationship of the course to Student Outcomes**  The course has been designed to contribute to the following student outcomes:   1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics 2. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions | | | | |
| **Prepared by:** Prof. Dr. Dogu Arifler | | | **Date Prepared:** 15 February 2019 | |