MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG230 Introduction to C Programming – Course Syllabus Spring 2012-2013

Credits/ECTS: (2-2) 3 / 4

Catalog Description: Introduction. Constants, variables, expressions, statements. Selective structures. Repetitive structures and arrays. Functions. Pointers. Multi-dimensional arrays. (Offered to non-CNG students only).

Instructors: Dr. İdil Candan (sections: 1,2) , Office: S-142, E-mail: cidil@metu.edu.tr
Dr. S. Esen Yuksel(section: 3), Office: S-208, E-mail: esen@metu.edu.tr
Assoc.Prof. Dr. Tolga Can(section: 4), Office: TZ-30, E-mail: tcan@metu.edu.tr
Lecture Hours: Section 1: Monday 13:40-15:30 in TAZ-08
Section 2: Monday 10:40-12:30 in TAZ-09
Section 3: Monday 15:40-17:30 in T-105
Section 4: Wednesday 15:40-17:30 in TAZ-09
Laboratory Hours: Section 1: Thursday 8:40-10:30 in I-103
Section 2: Wednesday 10:40-12:30 in I-106
Section 3: Thursday 10:40-12:30 in I-106
Section 4: Friday 15:40-17:30 in I-103

Office Hours: Tolga Can (Wednesdays 9:00-10:00, Thursdays 14:30-15:30)

Teaching Assistants: Şukru Eraslan (Sections: 1,2), TBA (Section:3,4)

Textbook: H. M. Deitel and D. J. Deitel, C: How to Program, Pearson Education, 5th Edition, 2006 (earlier editions can also be used).

Course Prerequisites: None

Learning Outcomes: CNG 230 is a service course offered to non-departmental students. The course aims to introduce students basic programming and to give them the ability to solve some simple problems using C programming language.
Having successfully completed this course, the student will be able to:

a. write syntactically correct simple C programs
b. make efficient use of an integrated development environment

Teaching Format: Weekly lectures and tutorials in the laboratories.

- This course will have two hours of lecture and two hours of labs per week as marked in the table given below.
- Lectures will focus on covering new material. There will be 2 midterm and 1 final exams. There may be a number of pop quizzes as well.
- Labs will focus on program development. You are expected to work on the given lab questions on your own and submit your answer to your TA at the end of the lab session. There will be 4 lab tests during semester. A lab task will be given and is to be solved during lab hours. Lab grade will mainly depend on it. Programming exercises will be given starting from week3 (Mar 4).

Grading

35 %	15% for Midterm 1 20 % for Midterm 2
35 %	Final
24 %	24 % for 4 Lab Tests (6% each)
6 %	Pop Quizzes

Attendance

We will take attendance for both lectures and tutorials. Students have to attend at least 75% of lectures and 75% of tutorials/labs in order to be admitted to the final exam. Students whose attendance is below 75% for either lectures or tutorials will not be admitted to the final exam. Valid medical leave with a medical certificate from the METU NCC Health Center will count towards this attendance requirement.

Course Outline (Tentative):

Week of	**Lecture**	**Lab**	**Reading**
Feb 18	Introduction: syllabus, computers, Internet and the Web		Ch 1
Feb 25	Introduction to C Programming – 1 (structure of C programs, simple print statements)		Ch 2
March 4	Introduction to C Programming – 2 (variables and C expressions)	LAB 1 - Introduction & Demo 1 – First exercises	
March 11	Structured Program Development in C (IF, IF…ELSE Statements)	LAB 2 - Introduction & Demo 2 – Exercises	

March 18	Structured Program Development in C (WHILE Repetition Statements)	LAB 3 – Basic Exercises	Ch 3
March 25	C Program Control (FOR Loops, counter-controlled repetition)	LAB 4- LabTest1 (Basic C statements, If, If…else statements)	Ch 4
Apr 1	C Program Control (SWITCH statement, DO…WHILE Loops)	LAB 5	
Apr 8	C Functions-1, Review	**MT1**	Ch 5
Apr 15	C Functions -2	LAB 6 – LabTest2 (while loop, for loop)	
Apr 22	C Functions -3	LAB 7	
Apr 29	C Arrays – 1	LAB 8 - LabTest3 (functions)	Ch 6
May 6	C Arrays - 2	**MT2**	
May 13	C Characters and Strings	LAB 9	Ch 8
May 20	REVIEW	LAB 10 – Lab Test 4 (arrays)	

Computer Usage

N/A

Category Content

Mathematics and Basic Sciences	20%
Engineering Sciences	80%
Humanities and Social Sciences	0 %
Departmental	0%

Relationship to Program Outcomes

This course contributes to fulfillment of the following program outcomes:

(i) Adequate knowledge in mathematics, science and engineering subjects pertaining to the relevant discipline; ability to use theoretical and applied information in these areas to model and solve engineering problems.

(ii) Ability to identify, formulate, and solve basic engineering problems ; ability to select and apply proper analysis and modeling methods for this purpose.

(iii) Ability to devise, select, and use modern techniques and tools needed for engineering practice; ability to employ information technologies effectively.

Academic Integrity

Copying, communicating, or using disallowed materials during an exam is cheating, of course. Students caught cheating on a midterm or final exam will be reported to the campus disciplinary committee.

Academic integrity is a more complicated issue for programming assignments, but one we take very seriously. Students naturally want to work together, and it is clear they learn a great deal by doing so. Getting help is often the best way to interpret error messages and find bugs, even for experienced programmers. In response to this, the following rules will be in force for programming assignments:

a. Students are allowed to work together in designing algorithms, in interpreting error messages, and in discussing strategies for finding bugs, but NOT in writing code.
b. Students may not share code, may not copy code, and may not discuss code in detail (line-by-line or loop-by-loop) while it is being written or afterwards.
c. Similarly, students may not receive detailed help on their code from individuals outside the course. This restriction includes tutors, students from prior terms, Internet resources, etc.
d. Students may not show their code to other students as a means of helping them. Sometimes good students who feel sorry for struggling students are tempted to provide them with "just a peek" at their code. Such "peeks" often turn into extensive copying, despite prior claims of good intentions.
e. Students may not leave their code (either electronic versions or printed copies) in publicly accessible areas. Students may not share computers in any way when there is an assignment pending.

We use various code comparison tools including automated tools to help spot assignments that have been submitted in violation of these rules. The tool takes all assignments from all sections and all prior terms and compares them, highlighting regions of the code that are similar.

We (the instructors and the teaching assistants) check flagged pairs of assignments very carefully ourselves, and make our own judgment about which students violated the rules of academic integrity on programming assignments. When we believe an incident of academic dishonesty has occurred, we contact the students involved. All students caught cheating on a programming assignment (both the copier and the provider) will receive an automatic 0 for that assignment. No excuses, no discussions, no exceptions!

For non-programming assignments, students have to use their own ideas, words, algorithms, tables, and figures. If citation is needed, proper and accurate citation for the used information sources must be given. Any sort of plagiarism will not tolerated. This means no copying, no rewarding, no paraphrasing, or giving false or in accurate information sources.

For more details about plagiarism, please see http://www.plagiarism.org/plag_article_what_is_plagiarism.html.

For further details on METU NCC’s rules concerning academic code of ethics, please see

http://www.ncc.metu.edu.tr/academic/acadcode-of-ethics.php