ELE 160 Programming for Engineers
Spring 2015
4 Credits

Lecture: Mondays and Wednesdays 10-11:15 A.M.
Lab: Fridays 1:15-4 P.M.
in Palanakila 122

INSTRUCTOR: Dr. Milica Barjaktarovic
OFFICE: Officially Palanakila 101 but that room has no computer, so I will be either in the classroom or Palanakila 109 (with computers)
OFFICE HOURS: MW 9:30-10 and 11:15-11:30, and by appointment.
TELEPHONE: EMAIL: milica@hawaii.edu

Windward Community College Mission Statement:
Windward Community College offers innovative programs in the arts and sciences and opportunities to gain knowledge and understanding of Hawai‘i and its unique heritage. With a special commitment to support the access and educational needs of Native Hawaiians, we provide O‘ahu’s Ko‘olau region and beyond with liberal arts, career and lifelong learning in a supportive and challenging environment — inspiring students to excellence.

Catalog Description: Introductory course on computer programming and modern computing environments with an emphasis on algorithm and program design, implementation and debugging. Designed for engineering students, this course includes a hands-on laboratory to develop and practice programming skills.
(Note: involves C-programing)

Pre-Requisites: Credit for or registration in Math 140 or consent of instructor

Recommendation: ICS 101

Credits: 4 (3 Lecture and one 3-hour Lab)

Learning Resources: Textbook and Other Required Materials:

Course website: Laulima. Lecture notes, syllabus, tentative schedule, assignment instructions and all other class materials that students are responsible for is posted there. Homework and projects are to be posted there.

**Topics:**
1. using computers as a tool in developing solutions to problems;
2. design process for writing programs to solve problems; using a program like C to organize information and work with simple program constructs and sources of error;
3. work with variables and declarations as well as reading and printing data;
4. using functions, modular programming looping, branching, character and numeric processing; use pointers, array and strings;
5. apply standard programming library and work with the operating system.

**Course Competencies:** A student should understand (i) the basic design process in developing an algorithm for a specific problem, implementing the algorithm in a programming language like C, and testing and verifying programs of small to medium size, (ii) the fundamental programming constructs including variables, scalar and compound data types, functions, and looping and branching, and (iii) the use of library functions and interaction with the operating system. Students should be able to work in teams to meet learning objectives and complete assignments.

**Assignments:**
During this semester, we will do theoretical and hands-on assignments. There will be:

- about 6 concepts assignments based on the reading for the day.
- about 6 programming assignments where the student will apply the specific knowledge gained in the lab to broader problems. Unless otherwise specified, these will be individual projects.
- one final team project.

If an assignment is to be done by a team, one assignment will be turned in per team. Each team member will document their contribution and will receive individual grade.

Lab sessions will be held weekly and are designed to give students the opportunity to become familiar with the systems and language constructs by writing, debugging and testing small programs to be submitted for grading. Unless otherwise specified, these lab assignments must be completed and submitted during the lab period in which they are assigned. In general, lab work will be individual effort; however, students are encouraged to work with their teammates to help each other understand the material covered.

Three exams will be given: two midterms and one final exam. We will have weekly quizzes in class every Wednesday, covering the material necessary to do the lab that week.
Grading is based as follows:

Labs: 20%
Assigned Projects: 20%
Final Project: 10%
Quizzes: 15%
Midterm 1: 8%
Midterm 2: 12%
Final Exam: 15%

Grades for the course will be as follows:

- A 90-100%
- B 80-89%
- C 70-79%
- D 60-69%
- F 0-59%

Grades are posted in Laulima.

Make-up exams / quizzes will be given only for extenuating circumstances and must be initiated by the student before he or she returns to class. Failure to arrange a make up with the instructor will result in a zero (0) for the missed exam. Make-up for missed exams/ quizzes must be done during office hours. Notes are not allowed on make-up quizzes.

SLOs:
Upon successful completion of ELE 160, the student should be able to:
1. Explain the steps involved in the programming process.
2. Solve simple problems and express those solutions as algorithms.
3. Use the fundamental techniques of selection, looping, assignment, input, and output to describe the steps the computer takes to solve a problem.
4. Write algorithms and code in a top-down manner.
5. Work with arrays in searching and sorting applications.
6. Work with structures and unions types.
7. Write, test, and debug small programs.
8. Write functions and use pointers.
10. Work in text based environment like UNIX.
11. Interface with text base using a GUI interface (we will use Microsoft Visual Studio 2013).
## Course Tasks and Student Learning Outcomes Alignment:

<table>
<thead>
<tr>
<th>Student Learning Outcomes (SLOs)</th>
<th>Assignments</th>
<th>Exercises</th>
<th>Exams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explain the steps involved in the programming process.</td>
<td></td>
<td></td>
<td><strong>Exam 1</strong></td>
</tr>
<tr>
<td>Solve simple problems and express those solutions as algorithms.</td>
<td></td>
<td></td>
<td><strong>Exam 1,2,3</strong></td>
</tr>
<tr>
<td>Use the fundamental techniques of selection, looping, assignment, input, and output to describe the steps the computer takes to solve a problem.</td>
<td></td>
<td></td>
<td><strong>Exam 1,2,3</strong></td>
</tr>
<tr>
<td>Write algorithms and code in a top-down manner.</td>
<td></td>
<td></td>
<td><strong>Exam 1,2,3</strong></td>
</tr>
<tr>
<td>Work with arrays in searching and sorting applications.</td>
<td></td>
<td></td>
<td><strong>Exam 2</strong></td>
</tr>
<tr>
<td>Work with structures and unions types.</td>
<td></td>
<td></td>
<td><strong>Exam 3</strong></td>
</tr>
<tr>
<td>Write, test, and debug small programs.</td>
<td></td>
<td></td>
<td><strong>Exam 1,2,3</strong></td>
</tr>
<tr>
<td>Write functions and use pointers.</td>
<td></td>
<td></td>
<td><strong>Exam 2,3</strong></td>
</tr>
<tr>
<td>Work with characters and strings.</td>
<td></td>
<td></td>
<td><strong>Exam 3</strong></td>
</tr>
<tr>
<td>Work in text based environment like UNIX.</td>
<td></td>
<td></td>
<td><strong>Exam 2,3</strong></td>
</tr>
<tr>
<td>Interface with text base using a GUI interface.</td>
<td></td>
<td></td>
<td><strong>Exam 2,3</strong></td>
</tr>
</tbody>
</table>

## Connection with GEOs:
- Develop the ability to perceive how people interact with their cultural and natural environments, through their own worldview and through the worldviews of others, in order to analyze how individuals and groups function in local and global contexts.
- Identify information needed in a variety of situations, and access, evaluate, and use relevant information effectively and responsibly.
- Make judgments, solve problems, and reach decisions using analytical, critical, and creative thinking skills.
- Use written, visual, and oral communication to discover, develop, and communicate meaning, and to respond respectfully to the ideas of others in multiple environments.
Textbook (online) EE 160

Programming in C
Bharat Kinariwala and Tep Dobry

This page is your access to the HTML version of the textbook for EE 160 this semester. It is also available in PDF format. We will cover the first nine chapters of the text (Part I) in this course. Part II (chapters 11-15) will be provided here.

Table of Contents

• Part I - Fundamental C Programming
  •
  • 📚 Chapter 1 Introduction.
  • 1.1 Computer System Organization
    1.1.1 Computer Hardware
    1.1.2 Computer Software -- The Operating System
    1.1.3 Utility Programs
    1.1.4 User Programs and Applications
  • 1.2 Representing Data and Program Internally
    1.2.1 Representing Data
    1.2.2 Main Memory
    1.2.3 Representing Programs
  • 1.3 Designing Programs and the C Language
    1.3.1 Designing The Algorithm
    1.3.2 The C Language
  • 1.4 Summary
  • References
  • 1.5 Exercises
  • 1.6 Problems

• 📚 Chapter 2 Basic Concepts.
  2.1 A Simple C Program
    2.1.1 Developing the Algorithm
    2.1.2 Translating the Algorithm to C
    2.1.3 Running the Program
2.2 Organization of C Programs - Simple Statements
  2.2.1 Comment Statements
  Defining a Function --- main()
  2.2.3 Variable Declarations
  2.2.4 The Assignment Statement
  2.2.5 Generating Output
2.3 Testing the Program
  2.3.1 Debugging the Program
  2.3.2 Documenting the Code
2.4 Input: Reading Data
2.5 More C Statements
  2.5.1 Making Decisions with Branches
  2.5.2 Simple Compiler Directives
  2.5.3 More on Expressions
    A Simple Loop --- while
  2.5.5 Controlling Loop Termination
  2.5.6 More Complex Loop Constructs - Nested Loops
2.6 Common Errors
2.7 Summary
2.8 Exercises
• 2.9 Problems
  •
  •
  •
  •
  •

Chapter 3 Designing Programs Top Down.
• 3.1 Designing the Algorithm with Functions
  3.1.1 Implementing the Program with Functions
• 3.2 Defining Functions
  3.2.1 Passing Data to and from Functions
  3.2.2 Call by Value and Local Variables
• 3.3 Coding Programs for Readability
  3.3.1 The C Preprocessor
  3.3.2 Macros
  3.3.3 Including Header Files
  3.3.4 Conditional Compilation
• 3.4 Interacting with the Operating System
  3.4.1 Standard Files and EOF
  3.4.2 Standard Files and Redirection
• 3.5 Debugging Guidelines
• 3.6 Common Errors
• 3.7 Summary
• 3.8 Exercises
• 3.9 Problems
• Chapter 4 Processing Character Data.

4.1 A New Data Type: char
   4.1.1 The ASCII Character Set
   4.1.2 Operations on Characters
   4.1.3 Character I/O Using getchar() and putchar()
   4.1.4 Strings vs Characters

4.2 Sample Character Processing Functions
   4.2.1 Converting Letter Characters
   4.2.2 Converting Digit Characters to Numbers
   4.2.3 Counting Words
   4.2.4 Extracting Words

4.3 New Control Constructs
   4.3.1 The switch Statement
   4.3.2 The break Statement
   4.3.3 The continue Statement

4.4 Mixing Character and Numeric Input

4.5 Menu Driven Programs

4.6 Common Errors

4.7 Summary

4.8 Exercises

4.9 Problems

• Chapter 5 Numeric Data Types and Expression Evaluation.

5.1 Representing Numbers
   5.1.1 Signed and Unsigned Integer Types
   5.1.2 Single and Double Precision Floating Point Numbers

5.2 New Control Constructs
   5.2.1 The for Statement
   5.2.2 The do...while Statement

5.3 Scalar Data Types
   5.3.1 Data Type void
   5.3.2 Enumeration
   5.3.3 Defining User Types: typedef

5.4 Operators and Expression Evaluation
   5.4.1 Precedence and Associativity
   5.4.2 The Data Type of the Result
   5.4.3 Some New Operators

5.5 Common Errors

5.6 Summary

5.7 Exercises
5.8 Problems

Chapter 6 Pointers.
6.1 What is a Pointer?
   6.1.1 Data vs Address
   6.1.2 Indirect Access of Values
6.2 Passing Pointers to Functions
   6.2.1 Indirectly Incrementing a Variable
   6.2.2 Computing the Square and Cube
   6.2.3 A function to Swap Values
6.3 Returning to the Payroll Task with Pointers
6.4 Common Errors
6.5 Summary
6.6 Exercises
6.7 Problems

Chapter 7 Arrays.
A Compound Data Type --- array
   7.1.1 Declaring Arrays
   7.1.2 Character Strings as Arrays
7.2 Passing Arrays to Functions
7.3 Arrays, Pointers, Pointer Arithmetic
   7.3.1 Pointers: Increment and Decrement
   7.3.2 Array Names vs Pointer Variables
7.4 String Assignment and I/O
7.5 Array Initializers
7.6 Arrays for Databases
7.7 Common Errors
7.8 Summary
7.9 Exercises
7.10 Problems

Chapter 8 Functions and Files.
8.1 The C Standard Library
   8.1.1 Character Processing Routines
   8.1.2 Math Routines
8.2 Formatted Input/Output
   8.2.1 Formatted Output: printf()
Chapter 11 String Processing.

11.1 The Data Type STRING

11.2 Library String Functions
   11.2.1 String I/O: gets() and puts()
   11.2.2 String Manipulation: strlen() and strcpy()
   11.2.3 String Operations: strcmp() and strcat()
   11.2.4 String Conversion Functions
   11.2.5 File I/O with Strings

11.3 More Example Programs
   11.3.1 Palindromes
   11.3.2 Words
   11.3.3 Substrings

11.4 Common Errors

11.5 Summary

11.6 Exercises

11.7 Problems

Chapter 12 Structures and Unions.

12.1 Structures
   12.1.1 Declaring and Accessing Structure Data
   12.1.2 Using Structure Tags
   12.1.3 Structures and Functions
   12.1.4 Pointers to Structures

12.2 Arrays of Structures
12.3 Sorting Arrays of Structures
12.4 Unions
12.5 Common Errors
12.6 Summary
12.7 Exercises
12.8 Problems

Chapter 13 Files and the Operating System.

13.1 Block Input/Output
13.2 Tell and Seek
13.3 A Small Database Example
13.4 Operating System Interface
13.5 Summary
• **13.6 Problems**

• **Chapter 14** Storage Class and Scope.
  
  **14.1 Storage Classes**
  14.1.1 Automatic Variables
  14.1.2 Register Variables
  14.1.3 External Variables
  14.1.4 Variable Definition vs Declaration
  14.1.5 An Example: Lexical Scanner
  14.1.6 Static Variables
  14.1.7 Storage Class for Functions
  14.1.8 Stack vs Heap Allocation

  **14.2 Dynamic Memory Allocation**
  14.2.1 Library Functions for Dynamic Allocation
  14.2.2 Dynamic Arrays

  **14.3 Pointers to Functions**
  14.3.1 Function Pointers as Returned Values

  **14.4 Summary**
  **14.5 Problems**

• **Chapter 15** Engineering Programming Examples.
  
  **15 Engineering Programming Examples**
  
  **15.1 Matrices**
  15.1.1 Matrix Operations: Transforms
  15.1.2 Matrix Operations: Sums and Products

  **15.2 Complex Numbers**
  15.2.1 Complex Numbers and Vectors
  15.2.2 Roots of Algebraic Equations
  15.2.3 Impedance of Electrical Circuits

  **15.3 Integrals**
  **15.4 Summary**
  **References:**
  **15.5 Problems**

• **Appendix**.
  
  **C Language Reference**
  Identifiers
Declarations and Definitions
Expressions and Operators
Statements and Control Constructs
Compiler Directives

- **ANSI C vs Old C**
- **The C Standard Library**
  - Common System Routines by Category
    - Input and Output Routines: `<stdio.h>`
    - Character class tests: `<ctype.h>`
    - String Functions: `<string.h>`
    - Utility Functions: `<stdlib.h>`
    - Mathematical Functions `<math.h>`
    - Limits `<limits.h>`, `<float.h>`
    - Diagnostics `<assert.h>`
  - Others
  - Alphabetic Listing of System Routines

- **About this document ...**

Comments about the online textbook to:
*tep@wiliki.eng.hawaii.edu*

Copyright © B. Kinariwala and T. Dobry. All rights reserved.

**Additional Information**

One of the biggest factors in academic success is attendance. One of the primary causes of failure is chronic absenteeism.

You are expected to attend class regularly and actively participate. Coming to class and participating in class and group activities are easy ways to be successful in class. Conversely, if students are either not present or do not participate in class activities, they will miss out on valuable skills and information, and their grade will suffer.

Students, who are ill or have other reasons for missing class, should email me for an excused absence. The student is responsible the material covered in class and any in-class work missed. In-class work for excused absences may be turned in within one week of the class missed. Any assignments due at the beginning of class should be turned in online or at the start of the next class.

Although I will be happy to help should you have difficulty understanding work missed during an absence, you are ultimately responsible for obtaining notes, handouts,
assignments, etc. Ignorance of an assignment is not an acceptable excuse for not having done it.

If you know beforehand you will be absent from class, a quick call, email or message to my office explaining your absence, its length and cause would be appreciated.

**Student conduct**

During class time, students are expected to dedicate their full attention to class activities. I expect all students to conduct themselves in a fashion appropriate to a professional setting. Class time will be conducted interactively and students are expected to help each other, especially on projects, just like professionals would do in a job setting. Therefore, it is essential that each student respect classmates by actively listening, cooperating on group tasks, participating in the class until the end of the semester, and expressing positive attitudes.

Students should familiarize themselves with the student conduct code in the most current WCC General Catalog. Any manifestation of violation, such as alcohol or drug abuse or online/in-class harassment or misconduct will be dealt with in accordance with the WCC Code of Student Conduct. We can have fun, but this is not a party. All class activities will be conducted in a way that is conducive to learning in an academic setting. This is an institution of higher education, and students will behave in an appropriate, respectful manner.

**Plagiarism**

Plagiarism is unprofessional and can cost you your job and your career. Plagiarism is the copying of another’s work either in part or whole, word-for-word, or paraphrased, without giving the author due credit (including work found on the WWW). You must do your own work. If students are using another's words or ideas, they must reference them appropriately. When in doubt, ask.

Accidental plagiarism will result in the student redoing the assignment. Deliberate plagiarism (and I will be the judge) will result in an F for the assignment, for all students involved. Continued deliberate plagiarism may result in an F for the class and possible expulsion from the college for all students involved.

**Miscellaneous**

Response time expectation: please give me 24 hours to respond to your email before you email me again. Or give me the entire weekend if applicable.

This class participates in MySuccess program. At Windward Community College we want every student to be successful. MySuccess is a system-wide effort that seeks to support
students early in the semester when they first begin experiencing difficulty in a class. If I feel that you are having difficulty in my class within the first few weeks of the semester (e.g. missing class, missing assignments, or low test scores), and working together to address your challenges shows that you would really benefit from being connected to resources outside of the classroom, I may refer you to your assigned counselor. Once referred, MySuccess will:

- Send an email to your hawaii.edu account to let you know about my referral; and
- Have a counselor follow-up with you by phone or by email to find out what kinds of help you might need, to connect you with the necessary resources, and to help you devise a strategy for success.

I will not refer you without telling you. However, if I do refer you, know that I am doing so in an effort to connect you with all the help you may need to do well this semester as your success is important to me.

**AMERICANS WITH DISABILITIES ACT COMPLIANCE**

Students with documented disability will be provided reasonable accommodations. If you have a physical, sensory, health, cognitive, or mental health disability that could limit your ability to fully participate in this class, and/or if you have a documented disability and have not voluntarily disclosed the nature of your disability and the support you need, you are invited to contact the Disability Specialist Counselor to discuss reasonable accommodations that will help you succeed in this class. Ann Lemke can be reached at 235-7448, lemke@hawaii.edu, or you may stop by Hale ‘Akoakoa 213 for more information. Please inform me of your special needs and then I will work with you and your disability counselor to assist you appropriately.
Presented here is a tentative schedule that is subject to change upon the discretion of the instructor. The most updated schedule with complete assignment due dates is available at the ELE 160 Laulima Website. This site is updated almost every week.

Please read the chapter shown in the subject area before coming to class and make sure to keep up.

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Read before class</th>
<th>In-class</th>
<th>Due at the start of class (papers on Mondays, code on Fridays)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M Jan 12</td>
<td>Introduction to Computers, Programs, and C (Ch. 1 in the online textbook)</td>
<td>Intro:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Digital Literacy Basics</td>
<td>• syllabus</td>
<td></td>
</tr>
<tr>
<td></td>
<td>W Jan 14</td>
<td>Basic Programming Concepts</td>
<td>• introductions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>F Jan 16</td>
<td>Install MS Visual Studio 2013 at home. Professional Version is free for students.</td>
<td>• class overview</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Express version is free to anyone.</td>
<td>• Laulima login</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• student profile</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lab: MS Visual Studio demo:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Write and run hello world! program</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>M Jan 19– MLK day</td>
<td>Basic Concepts (Ch. 2 in the online book)</td>
<td>Concept assignment #1:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>W Jan 21</td>
<td>Intro to Programming</td>
<td>Write down 3 questions you have for Weeks 1 and 2.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fr Jan 23</td>
<td>I/O screen/keyboard</td>
<td>Concept assignments may be handwritten, if written legibly.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Binary Numbers and ASCII</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Read before class</td>
<td>In-class</td>
<td>Due at the start of class (papers on Mondays, code on Fridays)</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>3</td>
<td>M Jan 26</td>
<td>Lab: Run and test given code (sphere)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>W Jan 28</td>
<td>Variables</td>
<td>Quiz: programming concepts</td>
<td>Concept assignment #2: Write down 3 questions you have for the Basics and answer questions at the end of lecture notes.</td>
</tr>
<tr>
<td></td>
<td>Fr Jan 30</td>
<td>Foundations of Algorithms (Ch. 3 in the online book)</td>
<td>Lab: Develop algorithms</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Review: Programming in imperative languages</td>
<td>Lab: finish sphere, test variables</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>M Feb 2</td>
<td>Quiz: IF</td>
<td>Programming assignment #1: triangle code</td>
</tr>
<tr>
<td></td>
<td>W Feb 4</td>
<td>Conditional Statements: IF (Ch. 2.5 in the online book)</td>
<td>Lab: Run and test conditionals</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fr Feb 6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>M Feb 9</td>
<td>Lab: experiment with C standard library and math functions.</td>
<td>Concept assignment #3: Write down 3 questions you have about functions and I/O. Write down your idea how you would call function ______</td>
</tr>
<tr>
<td></td>
<td>W Feb 11</td>
<td>Functions, C standard libraries (Ch. 3 in the online book)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>F Feb 13</td>
<td>Common functions (exponential, log, etc.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Random numbers</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>M Feb 16 NO CLASS President’s day</td>
<td>Quiz: functions</td>
<td>Programming assignment #2: a function with IF and C Standard Library (check for validity, generate a number)</td>
</tr>
<tr>
<td></td>
<td>W Feb 18</td>
<td>Functions (continued) I/O read/write to/from file (Ch. 3 in the online book)</td>
<td>Lab: functions and I/O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>F Feb 20</td>
<td></td>
<td>Lab: solve 2 linear equations with 2 unknowns</td>
<td></td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Read before class</td>
<td>In-class</td>
<td>Due at the start of class (papers on Mondays, code on Fridays)</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>-------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td>M Feb 23</td>
<td>I/O (continued)</td>
<td></td>
<td>Programming assignment #3: triangle code as function; input read from a file, output stored in a file (due on Wednesday)</td>
</tr>
<tr>
<td></td>
<td>W Feb 25</td>
<td>Review</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>F Feb 27</td>
<td>Exam #1 (on Friday)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>M Mar 2</td>
<td>Overview of Loops</td>
<td>Lab: loops and nested loops</td>
<td>Concept assignment #4: Write down 3 questions you have about loops and nested loops</td>
</tr>
<tr>
<td></td>
<td>W Mar 4</td>
<td>Loops (Ch. 2.5 and 5.2 in the online book )</td>
<td>Lab: Taylor series</td>
<td></td>
</tr>
<tr>
<td></td>
<td>F Mar 6</td>
<td>Numerical calculations: series and approximations Roundoff errors, precision vs. accuracy</td>
<td>Triangle code with a loop (in class)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nested Loops</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>M Mar 9</td>
<td>Single-Dimension Arrays (Ch. 7 in the online book )</td>
<td>Quiz: loops</td>
<td>Concept assignment #5: Write down 3 questions you have about arrays</td>
</tr>
<tr>
<td></td>
<td>W Mar 11</td>
<td>Two-dimensional arrays (Ch. 9 in the online book )</td>
<td></td>
<td>Programming assignment #4: compute a series (e.g. compute pi using Gregory/Leibniz formula, trig functions using Taylor series,</td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Read before class</td>
<td>In-class</td>
<td>Due at the start of class (papers on Mondays, code on Fridays)</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>--------------------------------------------------------</td>
<td>----------------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>10</td>
<td>F Mar 20</td>
<td>Pointers (Ch. 6 in the online book)</td>
<td>Quiz: arrays</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M Mar 16</td>
<td>Functions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>W Mar 18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Concept assignment #6: Write down 3 questions you have about pointers and functions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Programming assignment #5: arrays. Read grades from a file, display them, calculate grades, print to a file.</td>
</tr>
<tr>
<td></td>
<td>M-F Mar 23-27</td>
<td>SPRING BREAK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>F Apr 3 – NO CLASS Easter break</td>
<td>Object-Oriented Thinking: struct</td>
<td>Quiz on Pointers and functions (on Wed)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M Apr 30</td>
<td>Pointers (continued)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>W Apr 1</td>
<td>Functions (continued)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>W Apr 8</td>
<td>C code organization (header files)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>F Apr 10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>M Apr 13</td>
<td>Review</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>W Apr 15</td>
<td>Exam #2 (cumulative)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Read before class</td>
<td>In-class</td>
<td>Due at the start of class (papers on Mondays, code on Fridays)</td>
</tr>
<tr>
<td>------</td>
<td>---------------</td>
<td>--------------------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>F Apr 17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>M Apr 27</td>
<td><strong>Characters</strong> (Ch. 4 in the online book)</td>
<td>Lab: character arrays (aka strings)</td>
<td>Concept assignment #7: Write down 3 questions you have for character data and 2-D arrays.</td>
</tr>
<tr>
<td></td>
<td>W Apr 29</td>
<td>Putting it all together: <strong>final project</strong></td>
<td>Lab: arrays of strings</td>
<td></td>
</tr>
<tr>
<td></td>
<td>F May 1</td>
<td></td>
<td>Lab: header files</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><em>Pick your team and your project</em></td>
</tr>
<tr>
<td>15</td>
<td>M Apr 27</td>
<td></td>
<td>Work on final project</td>
<td>Concept assignment #8: Write down your first draft of your final project.</td>
</tr>
<tr>
<td></td>
<td>W Apr 29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>F May 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>M May 4</td>
<td></td>
<td><strong>Present your final project in class</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>W May 6:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>last day of class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Final exams week</td>
<td><strong>Final Exam</strong> (<em>Cumulative, emphasizing chapters after last midterm</em>)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


The online textbook content is in slightly different order, but all that we cover is in there:

- **Part I - Fundamental C Programming**
  - [Chapter 1](#) Introduction.
  - [Chapter 2](#) Basic Concepts.
    - 2.5 IF
    - WHILE loop
    - Nested loops
  - [Chapter 3](#) Designing Programs Top Down.
    - 3.2 Functions
    - I/O
  - [Chapter 4](#) Processing Character Data.
  - [Chapter 5](#) Numeric Data Types and Expression Evaluation.
    - 5.2 FOR and DO WHILE loops
  - [Chapter 6](#) Pointers.
  - [Chapter 7](#) Arrays.
  - [Chapter 8](#) Functions and Files.
  - [Chapter 9](#) Two Dimensional Arrays.

- **Part II - Some Advanced Features of C**
  - [Chapter 10](#) Sorting and Searching.
  - [Chapter 11](#) String Processing.
  - [Chapter 12](#) Structures and Unions.
  - [Chapter 13](#) Files and the Operating System.
  - [Chapter 14](#) Storage Class and Scope.
  - [Chapter 15](#) Engineering Programming Examples.