Dongseo University
Division of Computer Engineering
Introduction to Programming

Instructor(s): Dae-Ki Kang
Office: UIT706
Phone: 051-320-1724
Email: dkkang@dongseo.ac.kr

Classroom: UIT706
Class Time: 
Office Hours: 

Course Description:
This course provides students with a comprehensive study of the C programming language. Classroom lectures stress the strengths of C, which provide programmers with the means to write efficient, maintainable, and portable code. The lectures are supplemented with non-trivial lab exercises. The course deals with data types, structural programming, function, array, structures, enumerations, typedef, character strings, pointers, bit operations, and file I/O.

Course Goals & Objectives:
At the conclusion of this course, the successful (passing) students will be able to
- Program in C
- Understand program flow
- Understand functions, methods and routines
- Use arguments and return values
- Run a simple C program
- Understand the basics of numeric overflow
- Understand how numbers are encoded as characters in ASCII
- Understand the connection between function return values and variables
- Understand variables when programming in C
- Understand the use of arrays and pointers
- Understand the concept of control flow
- Better understand the use of statements and logic in C programming
- Understand the mechanisms for controlling flow statements such as for, while, and do-while loop
- Implement simple statements in C
- Better understand logical operators
- Use pointers for direct memory access and manipulation in C
- Change the memory address contained within a pointer
- Understand the need to learn pointers
- Understand the char pointer
- Introduce constants and string literals in C;
- Introduce the character string as an array of characters in C
Course Outline:

- **Week 1**
  - Introductions - Course Logistics and Syllabus
  - Learning Objectives
  - Textbooks
  - Labs
  - Grading

  Some Fundamentals
  - Compiling and running your first C program

- **Week 2**
  - Variables, Data Types, and Arithmetic Expressions
    - Working with Variables
    - Understanding Data Types and Constants
      - The Basic Integer Type `int`
      - The Floating Number Type `float`
      - The Extended Precision Type `double`
      - The Single Character Type `char`
    - Storage sizes and ranges
    - Type Specifiers: `long`, `long long`, `short`, `unsigned`, and `signed`
    - Working with Arithmetic Expressions
    - Integer Arithmetic and the Unary Minus Operator
    - The Modulus Operator
    - Integer and Floating-Point Conversions
    - Combining Operations with Assignment: The Assignment Operators

- **Week 3**
  - Program Looping
    - The `for` Statement
    - Relational Operators
    - Nested for Loops
    - Increment Operator
    - Program Input
    - Loop Variants
    - The `while` Statement
    - The `do` Statement
    - The `break` Statement
    - The `continue` Statement
- **Week 4**
  - Making Decisions
  - The if Statement
  - The if-else Construct
  - Logical Operators
  - Boolean Variables
  - Nested if Statements
  - The else if Construct
  - The switch Statement
  - The Conditional Operator

  Character Input/Output
  - Nested for Loops
  - Increment Operator
  - Program Input
  - for Loop Variants
  - The while Statement
  - The do Statement
  - The break Statement
  - The continue Statement

- **Week 5**
  - Arrays
    - The concept of array
    - Defining arrays
    - Initializing arrays
    - Character arrays
    - Multidimensional arrays
    - Variable length arrays

- **Week 6**
  - Functions
    - Defining a Function
    - Arguments and Local Variables
      --- Automatic Local Variables
    - Returning Function Results
    - Declaring a Function Prototype

- **Week 7**
  - Functions (advanced topics)
    - Functions and Arrays
      --- Arrays as parameters
--- Sorting Arrays
--- Multidimensional Arrays
- Global Variables
- Automatic and Static Variables
- Recursive Functions

- Week 8  Mid-Term Week

- Week 9

  Structures
  - Defining and using Structures
  - Functions and Structures
  - Initializing Structures. Compound Literals
  - Arrays of Structures
  - Structures Containing Structures and/or Arrays

  More on Data Types
  - Enumerated Data Types
  - The typedef Statement
  - Data Type Conversions

- Week 10

  Strings
  - Character Arrays/ Character Strings
  - Initializing Character Strings. The null string.
  - Escape Characters
  - Displaying Character Strings
  - Inputting Character Strings
  - String processing:
  --- Testing Strings for Equality
  --- Comparing Strings
  --- Copying Strings
  - Functions in <string.h>
  - String to number conversion functions
  - Character Strings, Structures, and Arrays
  - Example: Simple dictionary program
  --- Sorting the dictionary
  --- A better search in sorted arrays

- Week 11

  Pointers
  - Pointers and Addresses
  - Pointers and Function Arguments
- Pointers and Arrays
- Pointer arithmetics
- Pointers and strings

- **Week 12**
  - Pointers (advanced topics)
  - Dynamic memory allocation
  - Pointer arrays. Pointers to pointers
  - Multidimensional arrays and pointers
  - Structures and pointers

- **Week 13**
  - Binary representation of integer numbers
  - Operations on bits
  - The Bitwise AND Operator
  - The Bitwise Inclusive-OR Operator
  - The Bitwise Exclusive-OR Operator
  - The Ones Complement Operator
  - The Left Shift Operator
  - The Right Shift Operator
  - Binary representation of floating-point numbers

- **Week 14**
  - Standard Input and Output
  - Standard Input and Output (I/O)
    - Buffered/unbuffered input
    - Character I/O

- **Week 15**  
  - Dead Week

- **Week 16**  
  - Final Week

**Textbook(s)**

**Class Website:** e-Class
Course Assignments & Grading:

- Exams: mid-term exam 30%, final-term exam 30%
- Quizzes: N/A
- Course Projects: N/A
- Assignments: 30%
- Bonus Credit: 10%
- Grading: Curved

Grading Policies:

Course Policies:

- Attendance: Mandatory
- Academic Misconduct Policy: