

## Course Outline and Policy

**Objectives & Intended Learning Outcomes:** Students should be able to:

- Analyze a problem statement and develop an algorithm to solve the problem, using program development techniques (flow-charting and pseudo-code);
- Understand the concepts of Structured & Procedural Programming;
- Code the program in C language, test it, debug it (fix errors if any), and document completed program;
- Understand basic Object Oriented Programming concepts and their implementations in C++ (If time allows);
- Develop and debug simple microcontrollers-based systems.

### Course Contents:

- **Introducing C:** Basic features of C, history, strengths, and weaknesses.
- **Problem Solving:** Flowcharting, Algorithms, & Pseudo Coding;
- **C Fundamentals:** Writing simple programs, compiling, debugging, and linking.
- **Formatted Input/Output:** printf, and scanf.
- **Expressions:** Arithmetic operators, assignment operators, increments and decrements, evaluations.
- **Selection Statements:** One-way, Two Way, & Multiple-Way selections (Boolean values, if statement, else, switch, and break statements).
- **Loops:** while, do loop, for loop, exiting from a loop.
- **Basic Types.**
- **Arrays:** one-dimensional arrays, and multi-dimensional arrays
- **Functions:** calling, declarations, arguments, termination, and recursive functions.
- **Program Organization:** local and global variables, blocks, scopes.
- **Pointers:** pointer variables, pointer operators, pointer assignments, pointers as arguments and return values.
- **Pointers and Arrays:** pointer arithmetic, pointers for array processing, array names as pointers, pointers & multidimensional arrays.
- **Strings:** Literals, string variables, reading and writing strings, C String library, arrays of strings, command line arguments.
- **Structures:** Introduction, stricter variables, structure types, nested arrays of structures, enumerations.
- **Advanced uses of pointers:** Dynamically allocated storage, dynamically allocated strings & arrays, pointers to pointers, pointers to functions, dynamic data structures, heap, linked list, queue, stack, ..etc
- **Files:** File pointers, file streams & file operations.
- **Introduction to OOP:** Classes, objects, attributes, constructors, destructors, properties, methods, other OOP concepts (If time allows).
- **Applications In Hardware:** Introduction to Microcontrollers, Introduction to Arduino, I/O ports, Digital and Analog Signals, Arduino IDE Programming.

### Important Notes:

- Don't miss lectures, and be on time.
- Homework should be predicted at the end of each lecture.
- A quiz should be predicted each lecture.
- Honor Code: Don't ever share code!

### Assessments:

- |                                   |     |
|-----------------------------------|-----|
| • Term 1                          | 25% |
| • Term 2                          | 25% |
| • Assignments, Quizzes, & Project | 10% |
| • Final                           | 40% |

### Web site:

For useful materials, announcements about the course, & feedbacks, bookmark this web page: <http://www.e4t.net/c>

### References:

- C Programming: A Modern Approach, K. N. King, Georgia State University, ISBN 0-393-96945-2.
- C++ How to Program, Harvey M. Deitel, Paul J. Deitel, Prentice Hall; 2<sup>nd</sup> Edition, 1997, ISBN: 0-13528910-6.
- Programming in C, A Complete Introduction to the C Programming Language, 3<sup>rd</sup> Edition, Stephen Kochan, ISBN 978-0672326660.
- Problem Solving Using C: Structured Programming Techniques, Uckan, Yuksel, McGraw-Hill, ISBN 0075619369.
- The C Programming Language, 2<sup>nd</sup> Edition, Brian W. Kernighan, Dennis M. Ritchie, Prentice Hall Software, ISBN 978-0131103627.