

FE1008 Computing

[Lecture: 26 hrs; Tutorial: 12 hrs; Laboratory: 12 hrs Pre-requisite: Nil; Academic Unit: 3.0]

Learning Objective

This course introduces fundamental concepts in computing, with an emphasis on applications in engineering. The students taking the course will acquire the software “literacy” that is indispensable to working creatively in an engineering workplace. Initially an overview on the need of computer programming and a brief introduction to basic computer architecture would be provided. This is followed by problem solving, software development and programming techniques including the use of standard library routines. With this as background knowledge, the course focuses on algorithm design and documentation, larger program development and testing in an modular and structured manner. The student will learn how to use computer programming in solving problems in engineering contexts.

Content

C Programming for an Engineering Career. Introduction to Computers. C Language Fundamentals. Flow of Control. Functions and Libraries. Large Program Development. Arrays, Strings and Pointers. Basic File Processing. Further Topics.

Course Outline

1. C Programming for an Engineering Career
Why use C as a language to learn programming? Importance of computer programming in engineering carrier opportunities. Interdisciplinary applications of programming in Bioinformatics, AI, Forensics, Cryptography and Mechatronics. The relation to computer science, computer engineering, Information systems, Information technology, Software engineering majors. Introduction to modern programming paradigm – object oriented programming.
2. Introduction To Computers
Basic Computer Organization: brief overview of computer architecture. Operating systems: managing processes, peripherals, memory and filestore. Programs: statements, sequential execution, compilation and execution.
3. C Language Fundamentals
Program format; concepts: constants, variables, data type, declaration and definition, statements. Assignments; operations: arithmetic, expressions; precedence rules; truncation rules. Essential standard library functions: I/O.
4. Flow of Control
Decision-making: the relational/logical operators, and the if–else structure, switch and break. Iteration structures: while, for, and do while.

5. Functions and Libraries
More Standard Library Functions: I/O, mathematical. Programmer-defined Functions: prototypes, return values, parameters, and parameter passing by value.

6. Large Program Development
Structured Programming: steps in program development. Top-down stepwise refinement. Introduction to testing: black box testing and white box testing. Good programming practice: simplicity, comments, self documentation (symbolic constants, meaningful identifiers, consistent conventions), ease of understanding (small variable scopes, explicit function call effects), module and subroutine/function.

7. Arrays, Strings and Pointers
1-D arrays, introduction to 2-D arrays, strings, pointers

8. Basic File Processing
File and disk I/O: open, close, read, write. Default I/O.

9. Further Topics
(ANSI-C sections that have not been covered.) These would include: enumerated data types, string functions, recursions, structures, unions, dynamic data structures. A brief introduction to OOP, C++ and other relevant subjects such as Software engineering, Data Structures and Algorithms.

10. Lab Description
Throughout this course, the student will have access to computers equipped with the necessary tools for developing programs using the C language. In the initial part of the laboratory session, programming concepts would be thought as normally done in a tutorial class. During the latter part of the session a greater emphasis would be placed on student writing, testing and debugging C programs. This will be done through the use of given programming exercises. How problems can be solved using readily available library routines would be discussed.

Learning Outcome

Upon completion of the subject, the student should be able to:

- Appreciate the relationship among components in a computer system, namely hardware, operating system and application software.
- Understand the elementary components of a C program, namely, variable types, statements, operations and control flow.
- Write simple programs that use library functions including I/O and Math functions.
- Utilise simple data structures within the programs.
- Incorporate basic file processing techniques into the programs.

- Develop programs using good programming techniques.
- Conduct simple testing procedures on the programs
- Appreciate the importance of computer programming in an engineering career though the use of related problems discussed during lectures as well as in the laboratory.

Textbooks

1. H. H. Cheng, C for Engineers and Scientists: An Interpretive Approach, McGraw-Hill, 2009 (ISBN-10: 0073376051, ISBN-13: 978-0073376059)

References

1. P. J. Deitel and H. M. Deitel, C: How to Program, 6th Edition, Prentice-Hall, 2010 (ISBN-10: 0-13-705966-3, ISBN-13: 978-0-13-705966-9)
2. J. R. Hanly and E. B. Koffman, Problem Solving and Program Design in C, 6th Edition, Addison-Wesley, 2010 (ISBN-10: 0-321-60151-3, ISBN-13: 978-0-321-60151-3, QA76.73.C15H241 2010)
3. D. M. Etter, Engineering Problem Solving with C, 3rd Edition, Prentice-Hall, 2005 (ISBN: 0-13-142971-X, QA76.73.C153E56 2004)