EEN218 - Intermediate Programming

3 credits

B.S.E.E.			B.S.Cp.E.	B.S.I.S.E.	
EEN	EAN	WCN	ECN	IT	SE
REQ	**REQ**	**REQ**	**REQ**	**REQ**	**REQ**

2007-8 Catalog Data: Continuation of Programming with emphasis on C++ and the skills required of a capable programmer. Essential data structures and algorithms, and introducing algorithm analysis. Basic sorting, searching, and data management. Dynamic and static memory management. Object oriented programming.

Prerequisites: EEN 118

Texts:

1. C++ programming: Program design including data structures.
   D. S. Malik, Course Technology, inc., ISBN 1418836400, 2006

References: None

Objectives:

1. Create complete working programs making suitable use of any of the well known data structures and algorithms.
2. Make use of the essential features of object oriented programming as provided by C++.
3. Provide an elementary analysis of the time and space complexity of basic algorithms.
4. Understand how memory is organised in standard programming languages, and make use of that understanding.

Topics:

1. Object Oriented Programming and C++:
2. Classes, objects, and data representation
3. Class definition, members, methods, and related functions
4. Constructors and Destructors
5. Encapsulation, Abstraction, and Implementation hiding (protected, public)
6. Essential Algorithms and Associated Data Structures
7. Essential sorting and searching algorithms
8. Linked lists: structures and algorithms
9. Vectors, flexible arrays, bounds-tested arrays, and related structures
10. Programming Techniques:
11. Dynamic (heap) versus stack and global memory allocation
12. Pointer operations, allocation, arithmetic; arrays as pointers
13. Pointers to objects, arrays, arrays of pointers, pointers to arrays, etc: applications
14. Recursive design of functions and data structures
15. Advanced input and output processing
16. Software Engineering:
17. Structured program development, planned programming, handling larger projects
18. Time analysis of algorithms and functions, big-O notation
19. Efficiency analysis: memory usage, memory/time/flexibility trade-offs

Schedule: 150 minutes lecture per week

Professional Component: Engineering topics: 3 credits, design 2½ credits
Students design and implement many algorithms and programs.

Prepared by S. Murrell on 2006-03-04, Last updated 2010-01-02