Engineering Faculty Document No. 9-05

March 3, 2006

TO: The Faculty of the College of Engineering

FROM: The Faculty of the School of Electrical and Computer Engineering

RE: Title Changes to Graduate-Level Course

The faculty of the School of Electrical and Computer Engineering has approved the following title changes of a graduate-level course. This action is now submitted to the Engineering Faculty with a recommendation for approval.

From: ECE 663 Compiler Code Generation, Optimization, and Parallelization

To: ECE 663 Advanced Optimizing Compilers

ECE 663 Advanced Optimizing Compilers
Class 3, Lab 0, Credit 3
Offered fall of odd years
Prerequisites: ECE 573 and ECE 565

Course Description: This course presents the concepts and techniques to design and implement advanced, optimizing compilers. The course includes topics in program parallelization and scalar optimizations.

Reason: The title of this course was changed to better reflect the evolving course content.

Course History: The course has been and continues to be offered every two years.

Mark J.T. Smith

Professor and Head

Text: (Optional) Fischer and LeBlanc, Crafting a Compiler with C, Benjamin/Cummings, 1991, ISBN 0-8053-2166-7. Course notes and research papers will be used. Background texts: Michale Wolfe, High Performance Compilers for Parallel Computing, Addison-Wesley, ISBN 0-8053-2730-4. Utpal Banerjee, Dependence Analysis, Kluwer, ISBN 0-7923-9809-2. Ken Kennedy and John R. Allen, Optimizing Compilers for Modern Architectures: A Dependence-based
Approach, Morgan Kaufmann Publishers, ISBN 1558602860. Cooper and Torczon, Engineering a Compiler, Morgan Kaufmann, 2004, ISBN 1-55860-698-X.

Course Outcomes:

A student who successfully fulfills the course requirements will have demonstrated an ability to understand and use

- concepts and techniques of advanced optimizing compilers. In particular,
- the various passes of an optimizing compiler, including program analysis, dependence analysis, enabling transformations, loop restructuring, instruction level parallelism, parallel code generation, and issues in the compilation of object oriented languages,
- program analysis techniques used to determine the legality and profitability of transformations,
- open research issues related to these techniques, known solutions, and differences between alternative solutions,
- implementation methods and performance characteristics of these concepts and techniques.

Assessment Methods:

There will be at least one midterm and a final exam. 50% of the final grade will reflect the performance on a class project that each student will propose and conduct.