

CS 738: Advanced Compiler Optimizations

1. **Description:** This course aims to teach topics in program analysis and compiler optimizations.
2. **Prerequisites:** CS335 (Compiler Design)
3. **Course Contents:** The course will mainly cover topics from the following list (not necessarily in the same order). Not all topics listed below will be covered, and depending on class feedback, new topics may be added.
 - Introduction, compiler architecture, intermediate representations
 - Control flow analysis, control-flow graphs, basic blocks
 - Dataflow analysis
 - SSA form
 - Classical optimizations (constant folding, CSE, PRE)
 - Pointer and alias analysis
 - Interprocedural analysis
 - Advanced Topics:
 - Garbage Collection
 - Program Synthesis
 - Program Testing and Debugging
 - Types and Programming
4. **Lecture & Venue:** Wednesday, Friday 3:30-4:45 PM KD102
5. **Office Hours:** Walk in
6. **Evaluation Components:**
 - Assignments: 5-10%
 - Quizzes and Class Participation: 5-10%
 - Mid semester exam: 15 - 25%
 - End semester exam: 25 - 35%
 - Course Project: 30 - 45%
7. **Course Policies:** Honesty Practices and Withdrawal – in accordance with the Institute and DOAA norms.
8. **Books & References:** No specific textbook. Relevant references and texts (if needed) will be posted on the course homepage from time to time. Some of the following books may be useful:
 - a. day P. Khedker, Amitabha Sanyal, and Bageshri Karkare, **Data Flow Analysis: Theory and Practice**, CRC Press, USA (2009).
[Indian Edition is available](#), CRC Press, 2013.

- b. Appel, A., **Modern Compiler Implementation in Java** (or ML, or C), Cambridge University Press, 2002.
- c. Cooper, K., Torczon, L., **Engineering a Compiler**, Morgan Kaufmann, 2004
- d. Muchnick, S., **Advanced Compiler Design and Implementation**, Morgan Kaufmann, 1997.
- e. Aho, A., Lam, M., Sethi, R., Ullman, J., **Compilers: Principles, Techniques, & Tools**, Addison Wesley, 2007.
- f. Y. N. Srikant, Priti Shankar, **The Compiler Design Handbook: Optimizations and Machine Code Generation**, CRC Press, 2008
- g. Randy Allen, Ken Kennedy, **Optimizing Compilers for Modern Architectures: A Dependence-based Approach**, Morgan Kaufmann, 2001