|  |  |
| --- | --- |
| **ELEMENT** | **CONTENT** |
| DEPARTMENT | CIS |
| AUTHOR (S) | Peter Chapin, Craig Damon |
| COURSE NUMBER | **CIS 4050** |
| COURSE TITLE | **Compiler Design** |
| SHORT TITLE | Compiler Design |
| COURSE LEVEL | 4000 |
| DATE CREATED |  |
| CHECKED/CHANGED | 6/8/2017 |
| PREREQUISITES | CIS 3030, 3050 |
| COREQUISITES |  |
| RESTRICTIONS |  |
| SPECIAL FEES | No |
| CREDITS | 3 |
| HOURS | 3 hours of lecture per week |
| SEMESTER | Spring |
| COURSE DESCRIPTION | This course familiarizes the student with how computer languages are implemented. Throughout the course, the student writes a small compiler for a simplified programming language specified by the instructor, using compiler construction tools such as lexical analyzer generators and parser generators as well as creating some hand-built components. Although some theory is presented, the emphasis is on implementation. The programming is done in C or Java at the instructor's discretion |
| SUGGESTED TEXTS | *Engineering a Compiler*; Keith D. Cooper & Linda Torczon |
| OPTIONAL TEXTS |  |
| COURSE OUTCOMES | The successful student will be able to:   1. Use lexical analyzer generator and a parser generator to build a lexical analyzer and a parser 2. Read, understand, and design BNF grammars 3. Create a handmade regular expression matching library and a recursive decent parser 4. Use the techniques of syntax directed translation to build a program that can perform a useful transformation on a simple source language 5. Write a program that can perform basic type analysis (or some other form of semantic analysis) 6. Understand the principles of code generation and optimization and be able to implement at least some of those principles |
| COURSE CONTENT | 1. Lexical analysis, finite automata, and using a lexical analyzer generator 2. Context-free grammars and BNF 3. Recursive decent parsing and using a parser generator 4. Syntax directed translation 5. Semantic analysis, type systems 6. Intermediate code generation and optimization 7. Final code generation, register allocation 8. Optimization methods |
| LAB/STUDIO OUTCOMES |  |
| LAB/STUDIO CONTENT |  |
| LECTURE CAPACITY | 21 |
| LAB CAPACITY |  |
| GRADED OR P/NP | Graded |
| EVALUATION |  |
| DELIVERY METHOD | ONL |
| ROOM REQUIREMENTS |  |
| AUTHOR’S NOTES |  |