

## COURSE DESCRIPTION

Department and Course Number	<b>CS 402</b>	Course Coordinator	<b>Barrett Bryant</b>
Course Title	<b>Compiler Design</b>	Total Credits	<b>3</b>

### Current Catalog Description

*Lexical and syntactical scan, semantics, code generation and optimization, dataflow analysis, parallelizing compilers, automatic compiler generation.*

### Textbooks

*Compilers: Principles, Techniques, and Tools, by Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, 2<sup>nd</sup> ed., Addison-Wesley Longman, 2007.*

### References

*None*

### Course Goals

*The goal of this class is for students to become knowledgeable about the theory and practice of compiler design and implementation.*

### Prerequisites by Topic

*Programming Languages*

### Major Topics Covered in the Course

*Lexical Analysis, Syntax Analysis, Syntax-Directed Translation, Intermediate Code Generation, Run-Time Environments, Code Generation, Code Optimization*

### Laboratory projects (specify number of weeks on each)

*Lexical Analyzer (1 week)*

*Syntax Analyzer (3 weeks)*

*Type Checker (2 weeks)*

*Symbol Table Construction (2 weeks)*

*Intermediate Code Generator (3 weeks)*

*Code Generator or Interpreter (3 weeks)*

## Estimate CSAB Category Content

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	_____	_____ 2 _____	Computer Organization and Architecture	_____	_____ 4 _____
Algorithms Software Design	_____	_____ 20 _____	Concepts of Programming Languages	_____	_____ 4 _____

### Oral and Written Communications

*None*

### Social and Ethical Issues

*None*

### Theoretical Content

*The primary theoretical content is in the theory of programming language syntax, parsing and translation. Approximately 20 hours are spent on these topics.*

### Problem Analysis

*All course sections require a project to implement a compiler for a simple programming language. The language specification must be thoroughly analyzed before the compiler can be designed and implemented.*

### Solution Design

*All course sections require the design of a compiler for a simple programming language. This design is normally divided into different aspects of language implementation, such as lexical analysis, syntax analysis, symbol table construction, type checking, intermediate code generation, and final code generation.*