Syllabus for
CSC 453—Compiler Construction
3 Credit Hours
Spring 2006

I. COURSE DESCRIPTION

A study of the details of automatic programming language translation. Topics include program language structures, translation, loading, execution, storage allocation, compilation of simple expressions, statements, organization of a compiler including compile-time and run-time symbol tables, lexical scan, syntax scan, object code generation, error diagnostics, object code optimization techniques, overall design and use of compiler writing languages, and bootstrapping.
Prerequisite: CSC 353.
Academic technology fee: $45.

II. COURSE GOALS

This course is designed to enable the student to do the following:

A. Gain a knowledge of the theoretical aspects of compiling, including formal language theory, grammar, syntax trees, finite-state automata, bottom-up and top-down parsing methods, and precedence relations.

B. Study the major components of the compilation process, including lexical analysis, syntax analysis, semantic analysis, symbol table management, storage allocation, code generation, optimization, and error recovery.

C. Learn how to design and implement a working compiler.

III. STUDENT LEARNING OUTCOMES FOR THIS COURSE

A. Terminal Objectives
As a result of successfully completing this course, the student will be able to do the following:
1. Explain the concepts of compiler and interpreters.
2. Discuss the environment and phases of a compiler.
3. Describe parsing techniques.
4. Utilize precedence relations for parsing arithmetic expressions.
5. Design and develop a simple working compiler with the concepts of scanner, symbol table manager, and syntax analyzer.

B. Unit Objectives
1. Unit I
As a result of successfully completing Unit 1, the student will be able to do the following:
a) Discuss the logical parts of a compiler.
b) Define the terms used in formal language theory.
c) Derive a syntax tree for a given sentence and grammar.
d) Define precisely the meaning of "parse."
e) Identify relations that are reflexive and/or transitive.
f) Form the transitive closure of a given relation.
g) Define a regular grammar.
h) Draw a state diagram that will recognize sentences from a regular grammar
i) Define a finite state automaton

j) Write a scanner for a given language
k) Write a symbol table manager for a compiler. .

2. Unit 2

As a result of successfully completing Unit II, the student will be able to do the following:

a) Describe the difference between top-down and bottom-up parsing techniques.
b) Explain the method of recursive descent.
c) Explain and illustrate the use of precedence relations.
d) Parse arithmetic expressions using the method of operator precedence.
e) Describe methods for storage of elementary and structured data types.
f) Discuss techniques for establishing actual-formal parameter correspondence.
g) Write a syntax analyzer for a given language.

3. Unit 3

As a result of successfully completing Unit III, the student will be able to do the following:

a) Write appropriate semantic routines for a given source language construct.
b) Discuss the detection and recovery from syntax errors.
c) Describe the function of an interpreter.
d) Illustrate and describe a method for handling storage allocation in a block structured language.
e) Write a compiler for a simple language.

IV. TEXTBOOK

Required Textbook
Parsons, Thomas W., Introduction to Compiler Construction. New York: Computer Science Press, 1992.

V. POLICIES AND PROCEDURES

A. University Policies and Procedures

1. Attendance at each class or laboratory is mandatory at Oral Roberts University. Excessive absences can reduce a student's grade or deny credit for the course.
2. Double cuts are assessed for absences immediately preceding or following holidays.
3. Students taking a late exam because of an unauthorized absence are charged a late exam fee.
4. Students and faculty at Oral Roberts University must adhere to all laws addressing the ethical use of others' materials, whether it is in the form of print, video, multimedia, or computer software. By submitting an assignment in any form, the student gives permission for the assignment to be checked for plagiarism, either by submitting the work for electronic verification or by other means.
5. Final exams cannot be given before their scheduled times. Students need to check the final exam schedule before planning return flights or other events at the end of the semester.
6. Students are to be in compliance with University, school, and departmental policies regarding ePortfolio requirements. Students should consult the ePortfolio handbooks for requirements regarding general education and the students' majors.

a. The penalty for not submitting electronically or for incorrectly submitting an ePortfolio artifact is a zero for that assignment.
b. By submitting an assignment, the student gives permission for the assignment to be assessed electronically.

B. Department Policies and Procedures

1. Each Student who uses the computer is given access to the appropriate computer resources. These limited resources and privileges are given to allow students to perform course assignments. Abuse of these privileges will result in their curtailment. Students should note that the contents of computer directories are subject to review by instructors and the computer Administrative staff.
2. A fee of $15.00 will be assessed for all late exams. This policy applies to all exams taken without notifying the professor prior to the regularly scheduled exam time, and to all exams taken late without an administrative excuse.

C. Course Policies and Procedures

1. ePortfolio Requirements
   This course does not require an ePortfolio artifact.
2. Evaluation Procedures
   a. The final grade is based approximately 20% on homework, 30% on programs, 30% on exams, and 20% on the final exam.
   b. This course does not require an ePortfolio artifact.
3. Assignments
   Homework assignments and programming problems are given regularly in class. Details of specific requirements are given at that time.

VI. COURSE CALENDAR

Unit	Lesson	Topic
I	1-2	Introduction
	3-6	Finite State Machines
	7-9	Lexical Analysis
	10-12	Grammars and Languages
	13	**Exam I**
II	14-17	Top-down Parsing
	18-23	Bottom-up Parsing
	24-27	Code generation
	28	**Exam II**
III	29-44	Compiler Projects
	45	**Final Exam**

Course Inventory for ORU's Student Learning Outcomes

CSC 453 - Compiler Construction
Spring 2006

This course contributes to the ORU student learning outcomes as indicated below:
Significant Contribution – Addresses the outcome directly and includes targeted assessment.
Moderate Contribution – Addresses the outcome directly or indirectly and includes some assessment.
Minimal Contribution – Addresses the outcome indirectly and includes little or no assessment.
No Contribution – Does not address the outcome.

The Student Learning Glossary at http://ir.oru.edu/doc/glossary.pdf defines each outcome and each of the proficiencies/capacities.

OUTCOMES & Proficiencies/Capacities	Significant Contribution	Moderate Contribution	Minimal Contribution	No Contribution
1 **Outcome #1 – Spiritually Alive** Proficiencies/Capacities				
1A Biblical knowledge				X
1B Sensitivity to the Holy Spirit				X
1C Evangelistic capability				X
1D Ethical behavior			X	
2 **Outcome #2 – Intellectually Alert** Proficiencies/Capacities				
2A Critical thinking		X		
2B Information literacy			X	
2C Global & historical perspectives				X
2D Aesthetic appreciation			X	
2E Intellectual creativity		X		
3 **Outcome #3 – Physically Disciplined** Proficiencies/Capacities				
3A Healthy lifestyle				X
3B Physically disciplined lifestyle				X
4 **Outcome #4 – Socially Adept** Proficiencies/Capacities				
4A Communication skills		X		
4B Interpersonal skills			X	
4C Appreciation of cultural & linguistic differences				X
4D Responsible citizenship				X
4E Leadership capacity			X	