



**Birla Institute of Technology & Science, Pilani**  
Hyderabad Campus

## SECOND SEMESTER 2020-2021

### Course Handout Part II

Date: 16-01-2021

In addition to part-I (General Handout for all courses appended to the time table) this portion gives further specific details regarding the course.

Course No. : CS F363  
Course Title : **Compiler Construction**  
Instructor-in-Charge : **Prof. Aruna Malapati** (arunam@hyderabad.bits-pilani.ac.in)  
Instructors : **Prof.R.Gururaj, Dr. Jabez Christopher, Dr. Raghunath Reddy, Mr.Gourish Godur, Mr.Chillara Anil Kumar**

### **Scope and Objective of the Course:**

This course is an introductory course to compiler construction. In this course, students will learn the important basic elements of compilation to use the techniques effectively to design and build a working compiler. Topics include lexical analysis, parsing techniques, syntax directed translation, symbol table, intermediate code generation, data flow analysis, code generation, code optimization, error detection and recovery. Students will also participate in small teams in developing the building blocks of a compiler through compiler project. This course also includes a lab to provide hands-on experience on tools for implementing a compiler using Lex/Flex,Yacc.

- Gain an understanding of how compilers translate source code to machine executable code.
- Utilize tools to automate compiler construction.
- Comprehend how to perform parsing (top down and bottom up).
- Be familiar with techniques for simple code optimizations.
- Have the knowledge to design, implement, and test a compiler for a simple language, to include:
  - o Implementing efficient mechanisms for lexical analysis.
  - o Creating a parse table from a Context Free Grammar.
  - o Implementing an efficient symbol table during the parsing phase.
  - o Perform elementary semantic analysis checks on an abstract syntax tree.
  - o Generating code for a target assembly language

### **Textbooks:**

**T1. Aho, Sethi and Ullman.Compilers Principles, Techniques, and Tools.** Pearson Education. Low Price Edition. Second Edition

### **Reference books**

**R1.** Andrew Appel, Modern Compiler Implementation in Java. Cambridge University Press. (Foundation Books, New Delhi.) Rev. Ed. 2000.

**R2.** VINU V. DAS, Compiler Design Using FLEX and YACC,Prentice-Hall India



## Course Plan:

Lecture No.	Learning objectives	Topics to be covered	Chapter in the Text Book
1	<ul style="list-style-type: none"> <li>To understand the context and use of a compiler.</li> </ul>	Introduction to Course. Structure and Components of a compiler.	T1 Ch1(1.2)
2	<ul style="list-style-type: none"> <li>To identify tokens and lexems and also to implement a lexer given a context free grammar</li> </ul>	Tokens, Lexer functionality and its implementation	T1 Ch. 3
	<ul style="list-style-type: none"> <li>To list and identify various data structures that can be used in the implementation of symbol table</li> </ul>	Data Structures for symbol table organization	T1 Ch 2 2.7
3-13	<ul style="list-style-type: none"> <li>To be able to compare and identify proper use of the parsers based on the grammar.</li> <li>To be able to identify the appropriate parser given a context free grammar</li> </ul>	Parsing, Parser Generator functionality	T1 Ch. 4
		Grammar Transformations for different types of parsers	Notes
		Top Down Parser / Recursive descent parser , LL(1) parser LL(1) Grammar LL(1) Parse algorithm Computing first and follow sets	T1 Ch4 (4.4)
		Bottom Up parsers -LR(0), CLR(1), SLR,LALR	T1 Ch4 (4.5)
14-16	<ul style="list-style-type: none"> <li>To be able to formulate their own semantic grammar based on the task.</li> </ul>	Inherited and Synthesized Attributes	T1 Ch. 5
17-20	<ul style="list-style-type: none"> <li>To be able to apply the knowledge of semantic grammar to generate 3AC for various programming language constructs like if statements, loops, functions etc.</li> </ul>	3AC, Syntax Trees, Translation of Expressions, Type Checking, Control Flow	T1 Ch. 6
21-24	<ul style="list-style-type: none"> <li>To be able to perform optimization given a high level language program.</li> <li>To be able to apply appropriate optimization under different conditions.</li> </ul>	Basic blocks, Flow graphs	T1 Ch. 8.4,8.5.1
		Directed Acyclic Graphs (DAG)	T1 Ch. 8.5
		Loop optimizations	
		Global data flow analysis	T1 Ch. 8.5
25-28	<ul style="list-style-type: none"> <li>To understand the implementation of a back end of a compiler - Code Generation and Register allocation.</li> </ul>	Basic Blocks and Traces, Issues in code generation, Approach to code generation	T1 Ch. 8

**Evaluation Scheme:**

Component	Duration	Weightage (%)	Date & Time	Nature of Component
Lab Exam (Post Mid)		10%	April	Open
Project 5% (Lexer Pre Mid) 10%(Syntax Analysis and others Post Mid)	Take home	15%	TBA	Open
Mid-semester Exam	90 Mins	35%	03/03 9.00 - 10.30AM	Open
Comprehensive Exam	2 hrs	40%	06/05 FN	Open

**Chamber Consultation Hour:**

**Notices:** All notices related to the course will be displayed on CMS.

**Make-up Policy:**

- Prior Permission of the Instructor-in-Charge is required to get make-up for the Test-1/Test-2, or lab exam Only on producing documentary proof of possible absence, which proves that student would be physically unable to appear for the test/exam, the decision of granting the make-up will be taken.
- Prior Permission of Dean, AUGSD is required to get make-up for the comprehensive exam.
- Instructor's / Dean's decision in the matter of granting Make-up would be final.

**Academic Honesty and Integrity Policy:** Academic honesty and integrity are to be maintained by all the students throughout the semester and no type of academic dishonesty is acceptable.

**INSTRUCTOR-IN-CHARGE**  
**CS F363**

