BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, Pilani
Pilani Campus
Instruction Division

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 Number:	CS F363
Course Title:	Compiler Construction
Instructor-in-charge:	SHASHANK GUPTA (shashank.gupta@pilani.bits-pilani.ac.in)
Instructor:	Dr. Vinti Agarwal (vinti.agarwal@pilani.bits-pilani.ac.in)
	Yemineni Ashok (p20190024@pilani.bits-pilani.ac.in)
Course Website:	http://nalanda.bits-pilani.ac.in/

1. Objective

To expose the students to a class of special algorithms to process languages to translate high level user programs to low level assembly code. This includes algorithms and data structures handling both front end and back end of the compiler.

2. Scope 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 required for designing 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.

3. Pre-requisite courses

The course on Compiler Construction uses the concepts learnt in earlier courses such as '*Theory of Computation*', '*Data Structures and Algorithms*' and '*Principles of Programming Languages*' which are also the pre-requisite courses done by the students. Students will be expected to have sufficient understanding of the concepts of finite automata theory e.g. DFA (Deterministic Finite Automata), NFA (Non-deterministic Finite Automata), State Minimization Algorithm, CFG (Context Free Grammar), parse tree, ambiguity, push down automata etc. Also, each student should have a prior knowledge of data structures and experience in implementing linked list, stacks, trees, hash table, dynamic arrays etc.

4. Books

Text Book

T1 Compilers Principles, Techniques, and Tools.
Authors: Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffery D. Ullman
Publisher: Pearson Education. Second Edition.

Reference Books

R1 Programming Languages - Concepts and Constructs

Author: Ravi Sethi
Publisher: Pearson Education.

R2 Introduction to Algorithms
Authors: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
Publisher: Prentice Hall of India

R3 Modern Compiler Implementation in C.
Author: Andrew W. Appel
Publisher: Cambridge University Press. (Foundation Books, New Delhi.) .

R4 Modern Compiler Implementation in Java.
Author: Andrew W. Appel
Publisher: Cambridge University Press. (Foundation Books, New Delhi.) .

R5 Concepts in Programming Languages
Author: Robert W. Sebesta
Publisher: Pearson Education Inc.

5. Course Plan

Modules and Learning Objectives

Module	Title	Learning Objective(s)
C1	Introduction to Compilers	To understand the context and use of a compiler.
C2	Front End of a Compiler	To understand the implementation of a front end of a compiler - scanning, parsing and semantic analysis.
C3	Back End of a Compiler	To understand the implementation of a back end of a compiler - run time environments, Code Generation and Register allocation.
C4	Special aspects of compilers and runtime	To understand some special aspects of compilers and runtime such as code optimization, garbage collection etc.

Lecture Schedule

Lec.	Topic	**Module**	**To be taught by***
1-2	Overview of the course-language processors-compiler and interpreters, Phases of compiler, need for different compilers, attributes of a good compiler, source and target code, a language processing system, structure of a compiler. Course administration overview-evaluation components, project details etc.	C1	SG

3-5	Lexical Analysis-Tokens, patterns, lexemes, lexical errors, input buffering, regular expressions, recognition of tokens, Transition diagrams, recognition of reserved words and identifiers, architecture of a transition diagram based lexical analyzer, look ahead operator, Pattern matching based on NFA, DFA for lexical analyzer, Optimization of DFA based pattern matcher, State minimization in lexical analyzers, DFA simulation.	C2	SG
6-7	Syntax Analysis: Role of a parser, representative grammars, syntax error handling, error recovery strategies, CFG, parse trees, derivations, ambiguity, syntactic correctness, parsing process, Recursive Descent Parsing, left recursion elimination, left factoring.	C2	SG
8-10	Syntax Analysis: Top Down Parsing- first and follow sets, LL(1) Grammars, construction of predictive parsing table, Non Recursive Predictive parsing, error recovery in predictive parsing.	C2	SG
11-13	Syntax Analysis: Bottom Up Parsing- LR parsing, reductions, handle pruning, shift-reduce parsing, conflicts, Simple LR parsing, Items and LR(0) Automaton, LR parsing algorithm, SLR parsing tables, LR(1) Parsers: CLR(1) and LALR(1) Parsers, parser generators.	C2	SG
14-17	Syntax directed translation, inherited and synthesized attributes, evaluation order, dependency graphs, evaluation of attributes, construction of Abstract Syntax Tree (AST)	C2	SG
18-20	Type Checking and Type Inferencing- type expressions, type equivalence	C2	SG
21-22	Intermediate Representation; Intermediate Language , three address code, quadruples, triples, semantic rules to generate intermediate code for various constructs.	C3	VA
23-24	Code Generation-target language, program and instruction cost, addresses in the target code, run time addresses for names, register and address descriptors, code generation algorithm.	C3	VA
25-26	Run Time Environments: Storage organization, Stack allocation of space, contents of an activation record, memory	C4	VA

	models, calling sequences, variable length data on the stack, access to non-local data on the stack, data access without nested scopes, issues with nested procedures, nesting depth, access links.		
27-29	Code Optimization-Basic blocks, next use information, flow graphs, optimization of basic blocks, Liveness analysis.	C3	VA
30-32	Register Allocation, Instruction selection	C3	VA
33-34	Code optimization- Peephole optimization, redundant code elimination, flow of control optimizations.	C3	VA
35-36	Machine Independent Optimizations-Global Common Sub-expressions, copy propagation, dead code elimination, induction variables and reduction in strength etc.	C4	VA
37-38	Garbage Collection-Mark and sweep algorithm, reference counting algorithm	C4	VA
39-40	Discussion on few existing compilers	-	VA

* SG: Dr. Shashank Gupta *VA: Dr. Vinti Agarwal

6. **Evaluation Scheme:** The overall weightage is 200.

S. No.	Component	Mode	Duration	Date	Weight
1	Mid Semester Test	Closed Book	90 min	TBA (visit AUGSD website)	**30%**
2.	Quiz 1	Open Book	TBA	(20 Feb- 28 Feb) TBA	**15%**
3.	Quiz 2	Open Book	TBA	(15 April -29 April) TBA	**15%**
4.	Comprehensive Exam	Partially Open Book	60 Min	May 15th (FN)	**40%**

7. Makeup Policy:

- Quiz 1 and Quiz 2 would have only one combined make up exam. This means, that a student can abstain herself/himself (on valid reasons) in one Quiz. This make up would be conducted at last and would include complete syllabus. Of course, the student would apply for the make up in advance with documentary proof.
- 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 AUGSD is required to get make-up for the comprehensive exam.

8. Notices and Announcements

- Necessary notices, course announcements, uploading of marks of each component will be done on BITS-Nalanda. You are requested to check this website periodically. e-mail will be used as and when required.

9. Consultation Hour

Instructor	Hour
Shashank Gupta	Friday 5:00 PM – 6:00 PM
Vinti Agarwal	Tuesday 5:00 PM – 6:00 PM

**It is requested to the students to drop an email regarding an appointment for the concerned Instructor before joining the consultation hours.

10. Open Book Policy

- The prescribed text book, reference books listed in the handout and hand written (student's own) class notes are the only materials that will be allowed.

Instructor-in-charge

CS F363