SRI VENKATESWARA COLLEGE OF ENGINEERING

COURSE DELIVERY PLAN - THEORY

Department of Information Technology	LP: **IT16604** Rev. No: 00 Date: 14.12.2018
B.E/B.Tech/M.E/M.Tech : Information Technology Regulation:2016 PG Specialisation : NA Sub. Code / Sub. Name : IT16604 -AUTOMATA AND COMPILER DESIGN Unit : I	

Unit Syllabus:

UNIT I INTRODUCTION TO FINITE AUTOMATA 9

Introduction- Basic Mathematical Notation and techniques- Finite State systems – Basic Definitions – Finite Automaton – DFA & NDFA – Finite Automaton with €- moves – Regular Languages- Regular Expression – Equivalence of NFA and DFA – Equivalence of NDFA's with and without €-moves – Equivalence of finite Automaton and regular expressions –Minimization of DFA.

Objective:

Session No *	Topics to be covered	Ref	Teaching Aids
1.	Introduction To Finite Automata	1-Ch. 1; Pg:1-3	BB/LCD
2.	Basic Mathematical Notation and techniques	1-Ch. 1; Pg:4-5	BB/LCD
3.	Finite State systems	1-Ch. 1; Pg:6-13	BB/LCD
4.	Basic Definitions	1-Ch. 1; Pg:14-23	BB/LCD
5.	Finite Automaton – DFA & NDFA	1-Ch. 1; Pg:28-33, 1-Ch.2; Pg:37-60	BB/LCD
6.	Finite Automaton with €- moves	1-Ch. 2; Pg:72-80	BB/LCD
7.	Regular Languages	1-Ch. 3; Pg: 83-86	BB/LCD
8.	Regular Expression	1-Ch. 3; Pg: 87-90	BB/LCD
9.	Equivalence of NFA and DFA	1-Ch.1; Pg:60-64	BB/LCD
10.	Equivalence of NDFA's with and without €-moves	7-Ch. 2; Pg:26-27	BB/LCD

Sub. Code / Sub. Name: IT16604 /AUTOMATA AND COMPILER DESIGN
Unit : II

UNIT II LEXICAL AND SYNTAX ANALYSIS **12**

Lexical Analysis-Translators -The Phases of Compiler-Errors Encountered in Different Phases-compiler Construction Tools, Role of Lexical Analyzer-Specification and Recognition of Tokens-Thompson Construction –LEX.

Syntax Analysis-Role of the Parser- Top Down Parsing - Predictive Parser-LL(1) Parser-Shift Reduce Parser-LR Parser-SLR Parser, YACC.

Objective:

- To learn the design principles and tools of the compiler.

	Topics to be covered	Ref	Teaching Aids
13.	Introduction to Translators-The Language processing systems	2 – Ch.1;Pg.1-4	BB/LCD
14.	The Phases of Compiler	2– Ch.1;Pg.4-10	BB/LCD
15.	Errors Encountered in Different Phases	2 – Ch.3;Pg.113 – 114 2– Ch.4;Pg.194-195	BB/LCD
16.	Compiler Construction Tools	2 – Ch.1;Pg.12-12	BB/LCD
17.	Role of Lexical Analyzer	2 – Ch.3;Pg.109 – 110	BB/LCD
18.	Specification of Tokens- Regular Expressions	2 – Ch.3;Pg.116 – 124	BB/LCD
19.	Recognition of Tokens-NFA,DFA,State Transition Diagrams	2 – Ch.3;Pg.128 – 140	BB/LCD
20.	Converting regular expression to DFA(Thomson's method)Steps,algorithms	2 – Ch.3;Pg.152 – 166	BB/LCD
21.	Converting regular expression to DFA(Thomson's method)problems	2 – Ch.3;Pg.152 – 166	BB/LCD
22.	LEX.	2 – Ch.3;Pg.140 – 146	BB/LCD

Sub. Code / Sub. Name: IT16604 /AUTOMATA AND COMPILER DESIGN
Unit : III

Unit Syllabus :

UNIT III SYNTAX DIRECTED TRANSLATION 9

Syntax Directed Definitions--Intermediate Code Generation-Representation and Implementation - Types And Declarations —Type Checking –Control Flow Statements-Back Patching –Procedures.

Objective:

To learn the design principles of syntax analyzer

Session No *	Topics to be covered	Ref	Teaching Aids
28.	Syntax Directed Definitions	2– Ch.5;Pg.304-309	BB/LCD
29.	Intermediate Code Generation	2– Ch.6;Pg.357-358	BB/LCD
30.	Intermediate Code Representation	2– Ch.6;Pg.363-369	BB/LCD
31.	Intermediate Code Implementation	2– Ch.6;Pg.363-369	BB/LCD
32.	Types And Declarations Type conversion.	2– Ch.5;Pg.370-373	BB/LCD
33.	Construction of syntax tree	2– Ch.5;Pg.318-322	BB/LCD
34.	Bottom up evaluation of S-Attribute definitions,	2– Ch.5;Pg.348-352	BB/LCD
35.	Specification of a simple type checker	2– Ch.5;Pg.310-316	BB/LCD
36.	Control Flow Statements	2– Ch.6;Pg.399-408	BB/LCD
37.	Back Patching	2– Ch.6;Pg.410-417	BB/LCD
38.	Equivalence of type expression	2– Ch.6;Pg 352-359	BB/LCD
39.	Procedures	2– Ch.6;Pg.422-424	BB/LCD

Content beyond syllabus covered (if any):

* Session duration: 50 mins

Sub. Code / Sub. Name: IT16604 /AUTOMATA AND COMPILER DESIGN
Unit : IV

Unit Syllabus :

UNIT IV CODE OPTIMIZATION AND RUN TIME ENVIRONMENT **9**

Code Optimization -Principal Sources of Optimization-DAG- Optimization of Basic Blocks-Global Data Flow Analysis.

Run-time Environment- Source Language Issues-Storage Organization-Storage Allocation-Parameter Passing-Symbol Tables-Dynamic Storage Allocation

Objective:

To study the syntax directed translation and storage allocation issues

Session No *	Topics to be covered	Ref	Teaching Aids
40.	Principal Sources of Optimization	2– Ch.9;Pg.584-595	BB/LCD
41.	Directed Acyclic Graph	2– Ch.8;Pg.533-540	BB/LCD
42.	Optimization of Basic Blocks	2– Ch.8;Pg.533-542	BB/LCD
43.	Optimization of Basic Blocks	2– Ch.8;Pg.533-542	BB/LCD
44.	Global Data Flow Analysis.	2– Ch.9;Pg.597-600	BB/LCD
45.	Efficient Data Flow Algorithms	2– Ch.9;Pg.600-614	BB/LCD

Sub. Code / Sub. Name: IT16604 /AUTOMATA AND COMPILER DESIGN

Unit : V

Unit Syllabus :

UNIT V CODE GENERATION **6**

Issues -Design of Code Generator -Addresses in the Target Code -Basic Blocks in Flow Graph -Simple Code Generator -Peephole Optimization -Machine Independent Optimization

Objective:

- To learn how to obtain specific object code from source language after using various optimization techniques.

Session No *	Topics to be covered	Ref	Teaching Aids
52.	Issues in Design of a Code Generator	2 – Ch.8;Pg 506-516	BB/LCD
53.	A Simple Code Generator Algorithm	2 – Ch.8;Pg 542-548	BB/LCD
54.	Addresses in the Target Code	2 - Ch 8;Pg 518-525	BB/LCD
55.	Basic Blocks in Flow Graph	2 - Ch 8;Pg 525-532	BB/LCD
56.	Peephole Optimization	2 - Ch 8;Pg 549-553	BB/LCD
57.	Peephole Optimization	2 - Ch 8;Pg 549-553	BB/LCD
58.	Peephole Optimization	2 - Ch 8;Pg 549-553	BB/LCD
59.	Machine Independent Optimization	2 - Ch 9;Pg 655-669	BB/LCD
60.	Machine Independent Optimization	2 - Ch 9;Pg 655-669	BB/LCD

Content beyond syllabus covered (if any):

* Session duration: 50 mins

Sub Code / Sub Name: IT16604 /AUTOMATA AND COMPILER DESIGN

Course Outcome 1: Understand deterministic and non-deterministic machines.
Course Outcome 2: Formulate and Understand the analysis phase of the compiler
Course Outcome 3: Apply the various optimization techniques.
Course Outcome 4: Use the different compiler construction tools
Course Outcome 5: Design code generators for the specified machine.

Text Books:

1. John. E. Hopcroft, Rajiv Motwani and Jeffrey D Ullman, “Introduction to Automata Theory, Languages and Computation”, Third Edition, Pearson Education, 2014.
2. Alfred Aho, Ravi Sethi and Jeffrey D Ullman, “Compilers Principles, Techniques and Tools”, Second Edition, Pearson Education, 2013.

References:

1. Alexander Meduna, Petr Zemek,” Regulated Grammars and Automata”, Springer, 2014.
2. Torben Mogensen, “Basics of Compiler Design”, Springer, 2010.

	Prepared by	Approved by
Signature	1. SKK. 2. Rohini	26/12/18
Name	1. Ms. S. KALAVATHI 2. Ms. A. ROHINI	Dr. V. VIDHYA
Designation	Professor (IT)	Professor & HOD/IT i/c
Date	17/12/2018	17/12/2018
Remarks *:		
Remarks *:		

* If the same lesson plan is followed in the subsequent semester/year it should be mentioned and signed by the Faculty and the HOD