Department of Information Technology		LP: IT18602 Rev. No: 00 Date: 11-01-2021
~~B.E~~/B.Tech/~~M.E/M.Tech~~ : INT	Regulation: 2018	
PG Specialisation : NA		
Sub. Code / Sub. Name : IT18602 / AUTOMATA AND COMPILER DESIGN		
Unit : I		

Unit Syllabus:

UNIT I INTRODUCTION TO FINITE AUTOMATA 9 +3

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:

To understand deterministic and non-deterministic machines.

Session No *	Topics to be covered	Ref	Teaching Aids
1	Introduction to Finite Automata, Basic Mathematical Notation and techniques	1-Ch-1; Pg 2-5,27-30 3–Ch-2; Pg 10-14	online
2	Finite State systems, Basic Definitions	1-Ch-2; Pg 35-43 3–Ch-3; Pg 16-25	online
3	Finite Automaton – DFA	1-Ch-2; Pg 43-52	online
4	Finite Automaton –NDFA	1-Ch-2; Pg 53-64 4- Ch-2; Pg 15-18	online
5	Finite Automaton with €- moves	1-Ch-2; Pg 68-75	online
6	Regular Languages	1-Ch-4; Pg 122-135	online
7	Regular Expression	1-Ch-3; Pg 79-85 4- Ch 2; Pg 10-15	online
8	Equivalence of NFA and DFA, Equivalence of NDFA's with and without €-moves	1-Ch-4; Pg 143-148	online
9	Minimization of DFA	1-Ch-4; Pg 148-152 4-Ch- 2; Pg 30-35	online
10	Tutorial		online
11	Tutorial		online
12	Tutorial		online
Content beyond syllabus covered (if any):			

* Session duration: 50 minutes

Sub. Code / Sub. Name: IT18602 / AUTOMATA AND COMPILER DESIGN

Unit : II

Unit Syllabus :

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.

Session No *	Topics to be covered	Ref	Teaching Aids
13	Lexical Analysis- Introduction to Translators – The Language Processing Systems	2-Ch-1; Pg 1-4	online
14	The Phases of Compiler, Compiler Construction Tools, Errors Encountered in Different Phases	2-Ch-1; Pg 4-10,12 2-Ch-3; Pg 113-114 2-Ch-4; Pg 194-195 4- Ch-1; Pg 2-3	online
15	Role of Lexical Analyzer, Specification of Tokens – Regular Expressions	2-Ch-3; Pg 103-124	online
16	Recognition of Tokens – NFA, DFA, State Transition Diagrams	2-Ch-3; Pg 128-140	online
17	Converting Regular Expression to DFA(Thomson’s methods) - Steps, Algorithms, Problems	2-Ch-3; Pg 152-166	online
18	LEX	2-Ch-3; Pg 140-146	online
19	Syntax Analysis-Role of the Parser, Top Down Parsing, Predictive Parser-LL(1) Parser	2-Ch-4; Pg 192-193, 217-231 4-Ch-3; Pg 79-84	online
20	Shift Reduce Parser-LR Parser, SLR Parser	2-Ch-4; Pg 241-259	online
21	YACC	2-Ch-4; Pg 287-297	online
22	Tutorial		online
23	Tutorial		online
24	Tutorial		online
Content beyond syllabus covered (if any):			

* Session duration: 50 mins

Sub. Code / Sub. Name: IT18602 / AUTOMATA AND COMPILER DESIGN

Unit : III

Unit Syllabus:

UNIT III SYNTAX DIRECTED TRANSLATION 9 + 3

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

Objective:

To learn and understand the design principles of syntax analyzer

Session No *	Topics to be covered	Ref	Teaching Aids
25	Syntax Directed Definitions- Inherited and Synthesized attributes,	2-Ch-5; Pg 304-306	online
26	Evaluating an SDD	2-Ch-5; Pg 306-309	online
27	Intermediate Code Generation-Representation and Implementation	2-Ch-6; Pg 357-369	online
28	Types And Declarations – Type Expressions, Type Equivalence, Storage Layout for local names, Field in	2-Ch-6; Pg 370-378	online
29	Type Checking- Rules, Type Conversions, Overloading, Type Inference, Algorithm for	2-Ch-6; Pg 386-399	online
30	Control Flow Statements – Boolean Expressions, Short Circuit code	2-Ch-6; Pg 399-403	online
31	Control Flow statements-Translation, Avoiding redundant codes, Boolean Values	2-Ch-6; Pg 403-410	online
32	Back Patching – One-pass code generation, Flow of control statements, Break, Continue and Goto	2-Ch-6; Pg 410-418	online
33	Intermediate code for Procedures	2-Ch-6; Pg 422-424	online
34	Tutorial		online
35	Tutorial		online
36	Tutorial		online

Content beyond syllabus covered (if any):

* Session duration: 50 minutes

Sub. Code / Sub. Name: IT18602 / AUTOMATA AND COMPILER DESIGN

Unit : IV

Unit Syllabus:

UNIT IV CODE OPTIMIZATION AND RUN TIME ENVIRONMENT 9 + 3

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 learn how to optimize the code

Session No *	Topics to be covered	Ref	Teaching Aids
37	Code Optimization -Principal Sources of Optimization	2-Ch-9; Pg 584-595	online
38	Directed Acyclic Graph- DAG representation of basic blocks	2-Ch-8; Pg 533-534	online
39	Optimization of Basic Blocks	2-Ch-8; Pg 534-542	online
40	Global Data Flow Analysis- Abstraction, Schema, Basic blocks, Live variable Analysis, Available expressions	2-Ch-9; Pg 597-600	online
41	Run-time Environment- Source Language Issues-Storage Organization	2-Ch-7; Pg 427-430	online
42	Storage Allocation – Activation trees, Activation records	2-Ch-7; Pg 430-436	online
43	Parameter Passing	2-Ch-7; Pg 436-437	online
44	Symbol Tables	2-Ch-7; Pg 423-425	online
45	Dynamic Storage Allocation	2-Ch-7; Pg 452-463	online
46	Tutorial		online
47	Tutorial		online
48	Tutorial		online

Content beyond syllabus covered (if any):

* Session duration: 50 minutes

Sub. Code / Sub. Name: IT18602 / AUTOMATA AND COMPILER DESIGN

Unit : V

Unit Syllabus:

UNIT V CODE GENERATION 9 + 3

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

Session No *	Topics to be covered	Ref	Teaching Aids
49, 50	Issues in the design of code generator	2-Ch-8; Pg 506-512	online
51	Addresses in the Target Code – static allocation, stack allocation, Run-Time addresses for names	2-Ch-8; Pg 518-525	online
52	Basic Blocks	2-Ch-8; Pg 525-529	online
53	Flow Graphs	2-Ch-8; Pg 529-532	online
54, 55	Simple Code Generator-Register and address descriptors, Algorithm, Design of the function getReg	2-Ch-8; Pg 542-549	online
56	Peephole Optimization- Eliminating Redundant loads and stores, Unreachable code, Flow-of-control	2-Ch-8; Pg 549-553	online
57	Machine Independent Optimization	2-Ch-9, Pg 639-655	online
58	Tutorial		online
59	Tutorial		online
60	Tutorial		online
Content beyond syllabus covered (if any):			

* Session duration: 50 minutes

Sub Code / Sub Name: IT18602 / AUTOMATA AND COMPILER DESIGN

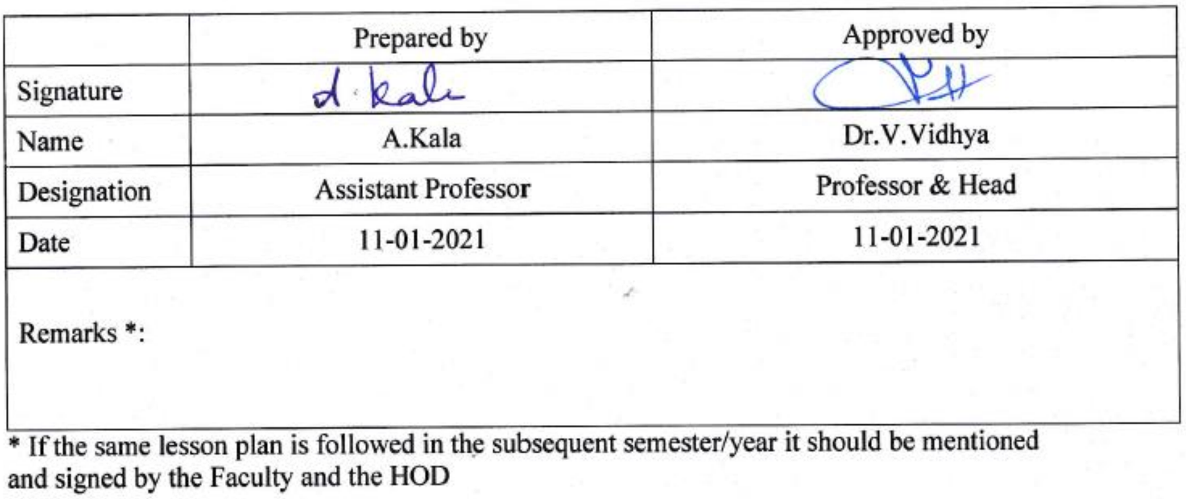
REFERENCES:

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.

3. Alexander Meduna, Petr Zemek,” Regulated Grammars and Automata”, Springer, 2014.

4.Torben Mogensen, “Basics of Compiler Design”, Springer, 2010.



	Prepared by	Approved by
Signature	A. Kala	
Name	A.Kala	Dr.V.Vidhya
Designation	Assistant Professor	Professor & Head
Date	11-01-2021	11-01-2021
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