SIR C.R.REDDY COLLEGE OF ENGINEERING, ELURU

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE HANDOUT

SUBJECT: AUTOMATA THEORY AND COMPILER DESIGN

CLASS: II/IV B.Tech. II SEMESTER, A.Y.2022-23

INSTRUCTOR: SRI CHALLA YALLAMANDA

Course Handout Index

S. No	Description
1	College Vision & Mission
2	Department Vision & Mission
3	Program Educational Objectives (PEOs)
4	Program Outcomes (POs)
5	Program Specific Outcomes (PSOs)
6	JNTUK Academic Calendar
7	Department Academic Calendar
8	Course Description
9	Course Objectives
10	Course Outcomes
11	Lesson Plan
12	Evaluation Pattern
13	Timetable
14	Unit wise Questions

College Vision & Mission

Vision: To emerge as a premier institution in the field of technical education and research in the state and as a home for holistic development of the students and contribute to the advancement of society and the region.

Mission: To provide high quality technical education through a creative balance of academic and industry oriented learning; to create an inspiring environment of scholarship and research; to instill high levels of academic and professional discipline; and to establish standards that inculcate ethical and moral values that contribute to growth in career and development of society in general.

Department Vision & Mission

Vision: To be a premier department in the region in the field of Information Technology through academic excellence and research that enable graduates to meet the challenges of industry and society.

Mission: To Provide dynamic teaching-learning environment to make the students industry ready and advancement in career; to inculcate professional and leadership quality for better employability and entrepreneurship; to make high quality professional with moral and ethical values suitable for industry and society.

Program Educational Objectives (PEOs)

PEO1: Solve real world problems through effective professional skills in Information Technology industry and academic research.

PEO2: Analyze and develop applications in Information Technology domain and adapt to changing technology trends with continuous learning.

PEO3: Practice the profession in society with ethical and moral values.

Program Outcomes (POs)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/Development of Solutions: Design solutions for complex engineering problems and system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, society, and environmental considerations.

PO4: Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional engineering solutions in society and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multi-disciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multi-disciplinary environments.

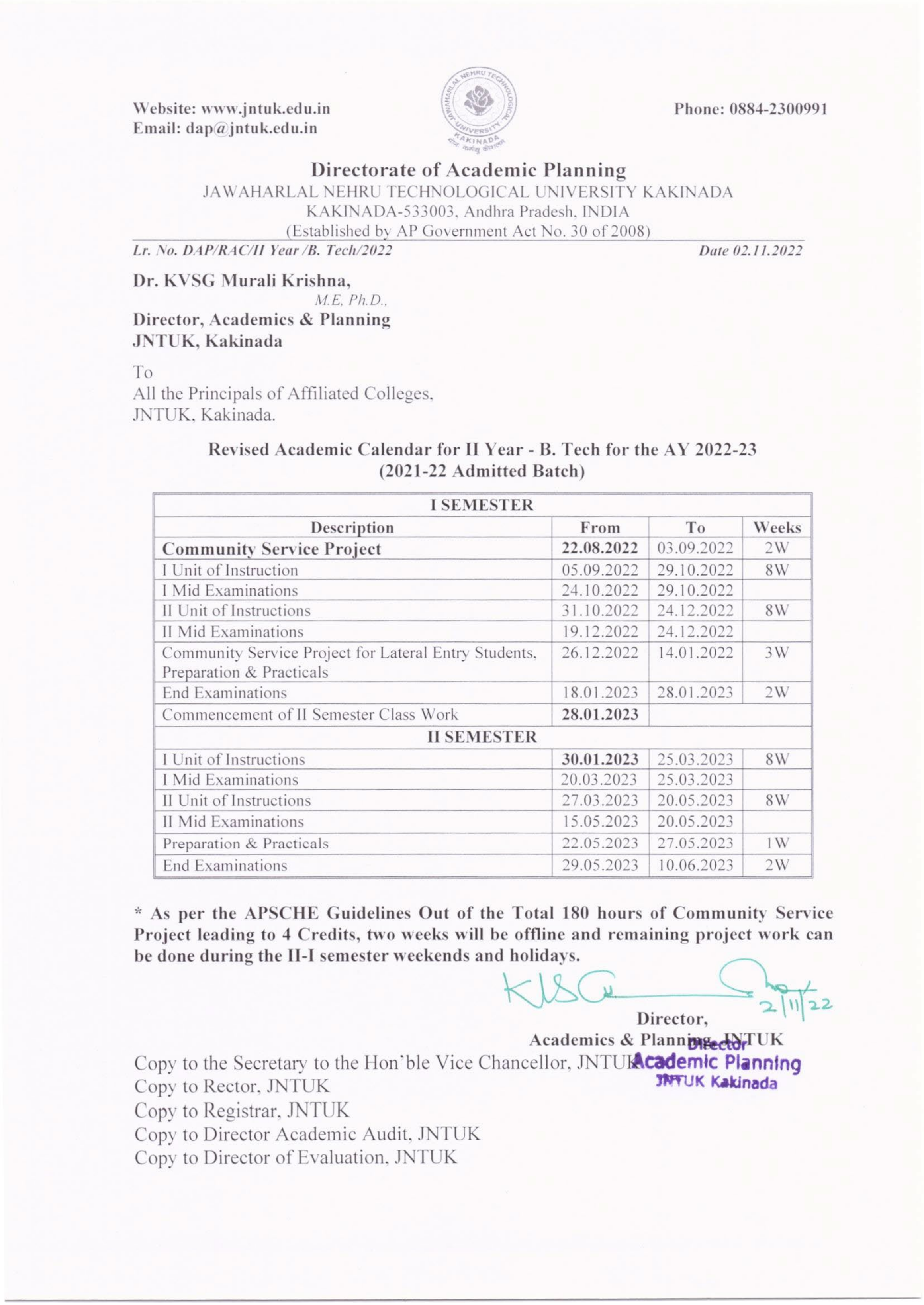
PO12: Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1: Design Skill: Design and develop softwares in the area of relevance under realistic constraints.

PSO2: New Technology: Adapt new and fast emerging technologies in the field of Information Technology.

JNTUK Academic Calendar



Website: www.jntuk.edu.in
Email: dap@jntuk.edu.in

Phone: 0884-2300991

Directorate of Academic Planning

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
KAKINADA-533003, Andhra Pradesh, INDIA
(Established by AP Government Act No. 30 of 2008)

Lr. No. DAP/RAC/II Year /B. Tech/2022 *Date 02.11.2022*

Dr. KVSG Murali Krishna,
M.E, Ph.D.,
Director, Academics & Planning
JNTUK, Kakinada

To
All the Principals of Affiliated Colleges,
JNTUK, Kakinada.

Revised Academic Calendar for II Year - B. Tech for the AY 2022-23
(2021-22 Admitted Batch)

I SEMESTER			
Description	**From**	**To**	**Weeks**
Community Service Project	**22.08.2022**	03.09.2022	2W
I Unit of Instruction	05.09.2022	29.10.2022	8W
I Mid Examinations	24.10.2022	29.10.2022	
II Unit of Instructions	31.10.2022	24.12.2022	8W
II Mid Examinations	19.12.2022	24.12.2022	
Community Service Project for Lateral Entry Students, Preparation & Practicals	26.12.2022	14.01.2022	3W
End Examinations	18.01.2023	28.01.2023	2W
Commencement of II Semester Class Work	**28.01.2023**		
II SEMESTER			
I Unit of Instructions	**30.01.2023**	25.03.2023	8W
I Mid Examinations	20.03.2023	25.03.2023	
II Unit of Instructions	27.03.2023	20.05.2023	8W
II Mid Examinations	15.05.2023	20.05.2023	
Preparation & Practicals	22.05.2023	27.05.2023	1W
End Examinations	29.05.2023	10.06.2023	2W

*** As per the APSCHE Guidelines Out of the Total 180 hours of Community Service Project leading to 4 Credits, two weeks will be offline and remaining project work can be done during the II-I semester weekends and holidays.**

2/11/22

Director,
Academics & Planning, JNTUK

Director
Academic Planning
JNTUK Kakinada

Copy to the Secretary to the Hon'ble Vice Chancellor, JNTUK
Copy to Rector, JNTUK
Copy to Registrar, JNTUK
Copy to Director Academic Audit, JNTUK
Copy to Director of Evaluation, JNTUK

Course Description

This course explores the principles, algorithms, and automata theory involved in the design and construction of compilers. Topics include finite-state machines, lexical analysis, context-free grammars, pushdown parsers, LR and LALR parsers, other parsing techniques, symbol tables, error recovery, and an introduction to intermediate code generation. Students are provided a skeleton of a functioning compiler in C to which they can add functionality.

Prerequisite

Students are expected to have completed an undergraduate knowledge of automata theory, context free languages, computer architecture, data structures and simple graph algorithms, logic or algebra.

Course Objectives

- To learn fundamentals of Regular and Context Free Grammars and Languages
- To understand the relation between Contexts free Languages, PDA and TM
- To study the various phases in the design of a compiler
- To understand the design of top-down and bottom-up parsers
- To understand syntax directed translation schemes
- To learn to develop algorithms to generate code for a target machine

Course Outcomes

Students are able to

CO No's	Cos	Level
CO1	**Understand** the language processors, finite Automata and compiler design phases.	**L2**
CO2	**Apply** various finite Automata techniques and compiler design techniques for a given problem.	**L3**
CO3	**Analyze** various compiler design techniques for given grammar.	**L4**
CO4	**Evaluate** various compiler phases for the given grammars.	**L5**

Lesson Plan

S. No	Unit	Topic	Teaching Aids	CO
1	I	Introduction to Unit I: **Formal Language and Regular Expressions**	BB	CO1
2		Languages	BB	CO1
3		Definition Languages regular Expressions	BB	CO1
4		Finite Automata - DFA, NFA	BB	CO1
5		Conversion of regular expression to NFA	BB	CO1
6		NFA to DFA	PPT	CO2
7		Applications of Finite Automata to lexical analysis	PPT	CO2
8		Lex tools	PPT	CO3
9	II	Introduction to Unit-II: **Context Free grammars and parsing**	BB	CO1
10		Context free grammars	BB/PPT	CO1
11		Derivation and parse trees	BB/PPT	CO1
12		Ambiguity LL(K) grammars	BB/PPT	CO1
13		LL(1) parsing	BB	CO1
14		Bottom up parsing	BB	CO1

15		Handle pruning LR Grammar Parsing	BB	CO1
16		LALR parsing	BB	CO1
17		Parsing ambiguous grammars	BB	CO1
18		YACC programming specification	BB	CO2
19	III	Introduction to Unit III: **Semantics**	BB	CO2
20		Syntax directed translation	BB	CO4
21		S-attributed and L-attributed grammars	BB/PPT	CO4
22		Intermediate code	BB	CO3
23		Abstract syntax tree	BB/PPT	CO4
24		Translation of simple statements and control flow statements	BB/PPT	CO3
25		**Context Sensitive features**	BB	CO2
26		Chomsky hierarchy of languages and recognizers	BB	CO3
27		Type checking	BB	CO4
28		Type conversions and equivalence of type expressions	BB/PPT	CO3
29		Overloading of functions and operations	BB	CO4
30	IV	Introduction to Unit IV: **Run time storage**	BB	CO2
31		Storage organization, storage allocation strategies	BB	CO4
32		Scope access to now local names, parameters	BB/PPT	CO4
33		Language facilities for dynamics storage allocation	BB	CO2
34		**Code optimization**	BB/PPT	CO2
34		Principal sources of optimization	BB	CO3
35		Optimization of basic blocks	BB/PPT	CO4
36		Peephole optimization	BB	CO2
37		Flow graphs	BB/PPT	CO2
38		Data flow analysis of flow graphs	BB	CO3
39	V	Introduction to Unit V: **Code generation**	BB	CO3

40		Machine dependent code generation	BB	CO2
41		Object code forms, generic code generation algorithm	BB/PPT	CO1
42		Register allocation and assignment	BB	CO2
43		Using DAG representation of Block	BB/PPT	CO2

TEXT BOOKS:

1) Introduction to Automata Theory, Languages and Computation, J. E. Hopcroft, R. Motwani and J. D. Ullman, 3rd Edition, Pearson, 2008.
2) Compilers Principles, Techniques and Tools Aho, Ullman, Ravisethi, Pearson Education.

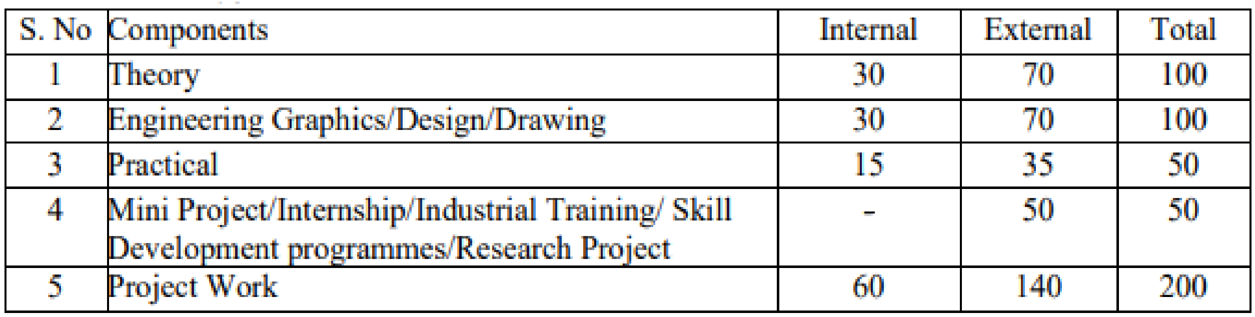
REFERENCES:

1) Louden: “Compiler Construction, Principles & Practice”, 1st Edition, Thomson Press, 2006.
2) Tremblay J P, Sorenson G P: “The Theory & Practice of Compiler writing”, 1st Edition, BSP publication, 2010.
3) Theory of Computation, V. Kulkarni, Oxford University Press, 2013

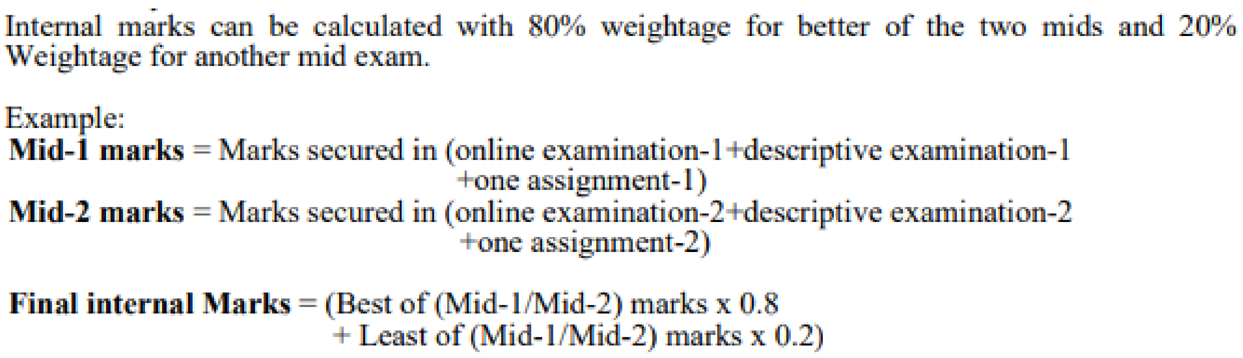
e-Resources:

1) https://nptel.ac.in/courses/106/104/106104028/
2) https://nptel.ac.in/courses/106/104/106104123/

Evaluation Pattern



S. No	Components	Internal	External	Total
1	Theory	30	70	100
2	Engineering Graphics/Design/Drawing	30	70	100
3	Practical	15	35	50
4	Mini Project/Internship/Industrial Training/ Skill Development programmes/Research Project	-	50	50
5	Project Work	60	140	200



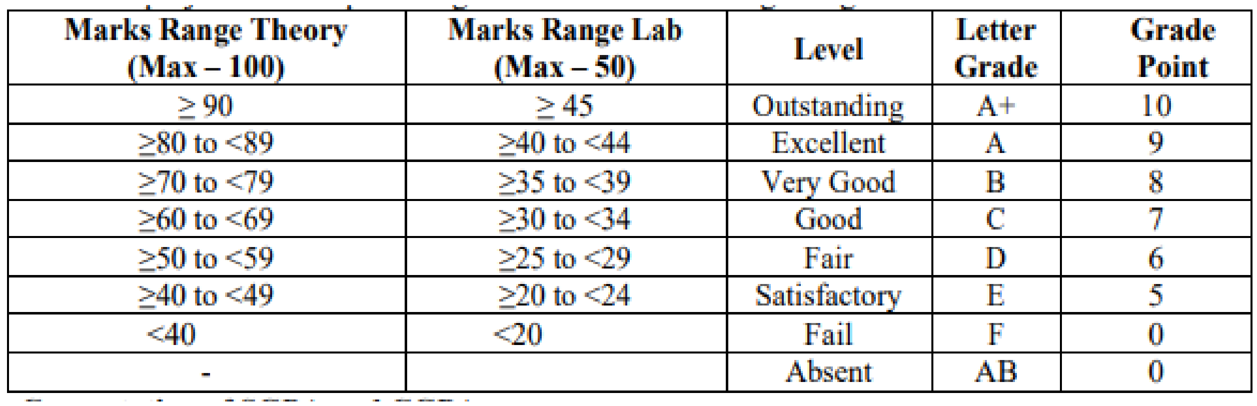
Internal marks can be calculated with 80% weightage for better of the two mids and 20% Weightage for another mid exam.

Example:

Mid-1 marks = Marks secured in (online examination-1+descriptive examination-1 +one assignment-1)

Mid-2 marks = Marks secured in (online examination-2+descriptive examination-2 +one assignment-2)

Final internal Marks = (Best of (Mid-1/Mid-2) marks x 0.8 + Least of (Mid-1/Mid-2) marks x 0.2)



Marks Range Theory (Max – 100)	Marks Range Lab (Max – 50)	Level	Letter Grade	Grade Point
≥ 90	≥ 45	Outstanding	A+	10
≥80 to <89	≥40 to <44	Excellent	A	9
≥70 to <79	≥35 to <39	Very Good	B	8
≥60 to <69	≥30 to <34	Good	C	7
≥50 to <59	≥25 to <29	Fair	D	6
≥40 to <49	≥20 to <24	Satisfactory	E	5
<40	<20	Fail	F	0
-		Absent	AB	0

Timetable

Day/Time	09.00-09.50	09.50-10.40	11.00-11.50	11.50-12.40	01.40-02.30	02.30-03.20	03.20-04.10	04.10-05.00
Mon			A		B			A&B(M)
Tue		B						
Wed	A			B				
Thu	B		A(T)					
Fri		A		B(T)				
Sat		A			A&B(R)			

*T – Tutorial Class

**R – Remedial Class

***M – Makeup Class

Unit wise Questions

Unit 1:

1. Explain the Phases of Compiler or Structure of Compiler.
2. Discuss the language specification for lexical analyzer.
3. Define Automata. Mention the various types of automata.
4. Conversion RE to NFA and NFA to DFA.
5. Summarize the applications of finite automata.

Unit 2:

6. Explain the CFG and how to derive the CGF?
7. Check whether the given grammars LL(1) or not. (Grammars: E grammars and S Grammar)
8. Construct the SLR parsing table for the following grammars. (Grammars: E grammars and S Grammar)
9. Construct the CLR parsing table for the following grammars. (Grammars: E grammars and S Grammar)
10. Construct the LALR parsing table for the following grammars. (Grammars: E grammars and S Grammar)
11. What is an ambiguous grammar? How to eliminate ambiguity for the grammar?
12. What is FIRST and FOLLOW? What are the rules for FIRST and FOLLOW?

Unit 3:

13. Differentiate between S and L Attributes.
14. Implement the various Implementation of three address code statement.
15. Discuss the Chomsky hierarchy of languages and recognizers.
16. What is an SDT? Write an STD for Control flow Statements.
17. Explain SDT for type conversions, equivalence of type expressions.

Unit 4:

18. Summarize the various storage allocation strategies.
19. Explain language facilities for dynamics storage allocation.
20. Discuss the principals of source code optimization.
21. Explain the optimization of basic block.
22. Summarize the peephole optimization.
23. Explain Data flow analysis of flow graphs.

Unit 5:

24. Explain the issues with code generation phases.
25. Discuss the generic code generation algorithm, Register allocation and assignment.
26. Draw the DAG representation of basic block.