INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad - 500 043

INFORMATION TECHNOLOGY

COURSE DESCRIPTION

Course Title	**AUTOMATA AND COMPILER DESIGN**			
Course Code	**A50513**			
Regulation	**R13**			
Course Structure	Lectures	Tutorials	Practicals	Credits
	4	-	-	4
Course Coordinator	Mr. D Rahul, Assistant Professor			
Team of Instructors	Mr. D Rahul, Assistant Professor			

I. COURSE OVERVIEW:

This course deals with the basic techniques of Compiler Construction and tools that can be used to perform syntax- directed translation of a high –level of a high level programming language into an executable code. This will provide deeper insights into the more advanced semantics aspects of programming languages, code generation, machine independent optimizations, dynamic memory allocation, types and their inferences, object orientation.

II. PREREQUISITES:

Level	Credits	Periods / Week	Prerequisites
UG	4	4	C Programming, Data Structures, Mathematical Foundations of Computer Science

III. COURSE ASSESSMENT METHODS:

a) Marks Distribution

Session Marks	University End Exam Marks	Total Marks
1. There shall be 2 midterm examinations. Each midterm Examination Consists of subjective type and Objective type tests. 2. The subjective test is for 10 marks, with duration of 1 hour. Subjective test of each semester shall contain 4 questions; the student has to answer 2 questions, each carrying 5 marks. 3. The objective type test is for 10 marks with duration of 20 minutes. It consists of 10 multiple choice and 10 objective type questions, the student has to answer all the questions and each carries half mark. 4. First midterm examination shall be conducted for the first four units of syllabus and second midterm examination shall be conducted for the Remaining portion. 5. Five marks are marked for assignments. There shall be two assignments in every theory course. Marks shall be awarded considering the average of two assignments in each course.	75	100

IV. EVALUATION SCHEME:

S. No	Component	Duration	Marks
1	I Mid Examination	80 minutes	20
2	I Assignment		05
3	II Mid Examination	80 minutes	20
4	II Assignment		05
5	External Examination	3 hours	75

V. COURSE OBJECTIVES:

1. **Construct** an insight of fundamental principles of designing compilers
2. **Discuss** the various phases of compiler design
3. **Develop** the skills needed for building compilers
4. **Differentiate** the organization of code optimizer
5. **Evaluate** Machine dependent code optimization

VI. COURSE OUTCOMES:

At the end of the course the students are able to:

1. **Formulate** lexical analysis phase of the compiler.
2. **Understand** the syntax analysis phase.
3. **Rewrite** Parsing using YACC.
4. **Classify** the Semantic Analysis and Intermediate code generation phase.
5. **Define** the Machine code generation phase.

VII. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes		Level	Proficiency assessed by
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	H	Assignments and Tutorials
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	H	Assignments and Tutorials
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	N	--
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.	H	Assignments and Tutorials
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.	N	--
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.	N	--

PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	N	--
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	N	--
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	H	Assignments and Tutorials seminars.
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.	S	--
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 multidisciplinary environments.	S	Projects
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.	S	--
N= None	**S= Supportive**	**H = Highly Related**	

VIII HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes		**Level**	**Proficiency assessed by**
PSO1	**Professional Skills:** The ability to research, understand and implement computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer-based	H	Lectures, Assignments
PSO2	**Software Engineering practices:** The ability to apply standard practices and strategies in software service management using open-ended programming environments with ability to deliver a quality service for business success.	S	Projects
PSO3	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies.	S	Guest Lectures

N - None **S - Supportive** **H - Highly Related**

IX. SYLLABUS:

UNIT – I

Formal Language and Regular Expressions: Languages, Definition Language regular expressions, Finite Automata-DFA, NFA. Conversion of regular expression to NFA, NFA to DFA, Applications of Finite Automata to lexical analysis,lex tools.

Context Free grammars and parsing: Context free grammars, derivation, parse trees, ambiguity LL (K) grammars and LL (1) parsing.

UNIT – II

Bottom up parsing handle pruning LR Grammar Parsing, LALR Parsing, Parsing ambiguous grammars, YACC Programming specification.

Semantics: Syntax directed translation, S-attributed and L-attributed grammars, and Intermediate code-abstract, syntax tree, translation of simple statements and control flow statements.

UNIT – III

Context sensitive features-Chomsky hierarchy of languages and recognizers. Type checking, type conversions, equivalence of type expressions, overloading of functions and operations.

UNIT – IV

Run time storage: Storage organization, storage allocation strategies scope access to now local names, parameters, language facilities for dynamics storage allocation.

Code optimization: Principal sources of optimization of basic blocks, peephole optimization, flow graphs, data flow analysis of flow graphs.

UNIT – V

Code generation: Machine dependent code generation, object code forms, generic code generation algorithm, register allocation and assignment using DAG representation of Block.

TEXT BOOKS

1. Principles of compiler design -A.V. Aho . J.D.Ullman; Pearson Education.
2. Modern Compiler Implementation in C- Andrew N. Apple, Cambridge University Press.

REFERENCES

1. Lex & yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
2. Modern Compiler Design- Dick Grune, Henry E. Bal, Cariel T. H. Jacobs,Wiley dreamtech.
3. Engineering a Compiler-Cooper & Linda, Elsevier.
4. Compiler Construction, Louden, Thomson
5. "Introduction to Automat a Theory Languages and Computation". Hopcroft H.E. and Ullman J.D. Pearson Education

IX. COURSE PLAN:

At the end of the course, students are able to achieve the following learning outcomes.

Lecture No.	Course Learning Outcomes	Topics to be covered	Reference
1	**Understand** the basic compilers and compilation process	Introduction, structure, Phases of Compilation.	T1:1.1-1.8
2-10	Relate regular grammar to programming feature	Construction of regular expressions, NFA,DFA, conversions	T1:3.5-3.7
11	**Differentiate** Pass and Phases of translation	Concept of pass and difference between pass and phase.	T1:3.3,T2:2.3
12	**Design** of compiler for a language	Bootstrapping and types of compiler.	T1:4.1

13	**Identify** Data structure in compilation Using lexical analyzer	Lex-Lexical analyzer generator Derivations and parse tree regular expressions v/s context free grammar.	T1:4.1
14-16	**Understand** Top down parsing techniques	Backtracking, LL(1),Recursive decent parsing Finding FIRST and FOLLOW.	T1:3.8,T2:2.5 T1:4.1, T2:3.1
17-20	**Construct** the parsing table for given inputs	Construction of parse tables, Predictive parsing.	T1:4.3
21-22	**Understand** bottom up parsing techniques	Shift reduce parsing, operator precedence parsing	T1:4.4
23-25	**Differentiate** types of LR(0) parsers	LR-SLR,LR(0)	T1:4.5
26-28	**Differentiate** types of LR(1) parsers	LALR,CLR.	T1: 4.6
29	**Construct** a parse tree for ambiguous grammar	Description of error recovery	T1: 4.7
30	**Implement** parser generator	YACC parser generator	T1:4.8
31-32	**Implement** the construction of syntax trees	Abstract syntax tree, three address code	T1:4.9
33-35	**Recognize** the semantics of grammar	Introduction to attributes grammars Syntax directed definitions, applications of SDD, implementing L-attributed SDD's	T1:5.2
36-37	**Describe** the forms of intermediate code generation phases	Control flow, back patching, switch statements	T1:5.1.5.3,5.4
38-40	**List** different types of language constructs	Rules, type conversions, Overloading, type inference and polymorphic functions.	T1:8.1-8.6
41-43	**Summarize** the symbol table	Symbol table format, ordered and unordered symbol tables. Organization for block structures languages	T1:6.1-6.6
44-46	**List** different types of storage allocation	Static, runtime stack and heap storage allocations	T1:7.6
47-48	**Understand** storage allocations for data structures	Storage allocation for arrays, strings and records	T1: 7.7
48-50	**Understand** Various optimization techniques	Introduction for optimization. Local, global and scope optimization	T1:7.8-7.9
51-53	I**mplementation** of basic block optimization techniques	Basic blocks, flow graphs, loops, code motion, induction variables, reduction in strength	T1:10.1-10.2
54-55	**Construction** of DAG	DAG construction, applications	T1:10.3-10.4
55-57	**Understand** the Data flow analysis	Data flow analysis of flow graphs. Flow graph, loops in flow graphs Representing data flow information, data flow equations for programming constructs	T1:10.5
58-60	**Implement** optimization on data flow graphs	Examples for sub expression elimination, Live variable analysis copy propagation and examples	T1:10.6-10.8 T2:9.1
61-62	**Understand** various code generation techniques	Introduction, issues in code generation, object code forms Need of machine dependent code optimization, peephole optimization	T1:10.9-10.13

63-65	**Implement** machine dependent optimizations	Global register allocation, register assignment for outer loops Rearranging the order, heuristic ordering for DAGs, optimal ordering and labeling algorithm	T1:10.12

X. MAPPING COURSE OBJECTIVES LEADING TO THE ACHIEVEMENT OF THE PROGRAM OUTCOMES:

Course Objectives	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	H		H	S					S				H		S
2	H									H			S		
3		H	S	H					H				S		H
4		S								S				H	
5			H						S						S

XI. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF THE PROGRAM OUTCOMES:

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	H	H	H	H						S			H	S	H
2		H		S											H
3	H		H						H				H	S	
4	S	S								H					
5	S	S	H	S					S	S			S	H	H

S =Supportive **H = Highly Related**

Prepared by: Mr D Rahul, Assistant Professor

Date: 07/07/2017

HOD, IT