Course Code	21CSC304J	Course Name	COMPILER DESIGN	Course Category	C	PROFESSIONAL CORE	L	T	P	C
							2	0	2	3

Pre-requisite Courses	*Nil*	Co- requisite Courses	*Nil*	Progressive Courses	*Nil*
Course Offering Department	*School of Computing*	Data Book / Codes / Standards		*Nil*	

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*
CLR-1:	*outline the implementation of Lexical Analyzer*
CLR-2:	*learn the various parsing techniques*
CLR-3:	*familiarize the intermediate code generation and run-time environment*
CLR-4:	*learn the implementation of code generator*
CLR-5:	*identify the various methods for Code Optimizer*

Course Outcomes (CO):	*At the end of this course, learners will be able to:*	Program Outcomes (PO)												Program Specific Outcomes		
		1	2	3	4	5	6	7	8	9	10	11	12			
		Engineering Knowledge	Problem Analysis	Design/development of solutions	Conduct investigations of complex problems	Modern Tool Usage	The engineer and society	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO-1	PSO-2	PSO-3
CO-1:	*acquire knowledge of Lexical Analyzer from a specification of a language's lexical rules*	*3*	*2*	*-*	*-*	*2*	*-*	*-*	*-*	*-*	*-*	*-*	*-*	*-*	*1*	*-*
CO-2:	*apply different parsing algorithms to develop the parsers for a given grammar*	*3*	*3*	*-*	*-*	*2*	*-*	*-*	*-*	*-*	*-*	*-*	*-*	*-*	*1*	*-*
CO-3:	*gain knowledge to translate a system into various intermediate codes*	*3*	*2*	*-*	*-*	*2*	*-*	*-*	*-*	*-*	*-*	*-*	*-*	*-*	*1*	*-*
CO-4:	*analyze the methods of implementing a Code Generator for compilers*	*3*	*2*	*-*	*3*	*-*	*-*	*-*	*-*	*-*	*-*	*-*	*-*	*-*	*1*	*-*
CO-5:	*design the methods of developing a Code Optimizer*	*3*	*-*	*2*	*3*	*-*	*-*	*-*	*-*	*-*	*-*	*-*	*-*	*-*	*1*	*-*

Unit-1 - Introduction	***12 Hour***
Compilers-Phases of Compiler-Cousins of the Compiler-Grouping of Phases-Compiler construction tools- Lexical Analysis-Role of Lexical Analyzer-Input Buffering -Specification of Tokens -LEX -Finite Automata-Regular Expressions to Automata -Minimizing DFA.	
Unit-2 – Top Down Parsing	***12 Hour***
Role of Parser-Grammars-Error Handling-Context-Free Grammars-Writing a grammar- Elimination of Ambiguity-Left Recursion- Left Factoring-Top Down Parsing — Recursive Descent Parser- Predictive Parser-LL(1) Parser- Computation of FIRST-Computation of FOLLOW-Construction of a predictive parsing table-Predictive Parsers LL(1) Grammars- Predictive Parsing Algorithm- Problems related to Predictive Parser - Error Recovery in Predictive Parsing-.	
Unit-3 – Bottom-Up Parsing	***12 Hour***
Bottom Up Parsing-Reductions-Handle Pruning-Shift Reduce Parser-Problems related to Shift Reduce Parsing-Operator Precedence Parser, LEADING, TRAILING -LR Parser- LR Parsers- Need of LR Parsers-LR (0)Item-Closure of Item Sets- Construction of SLR Parsing Table -Problems related to SLR-Construction of Canonical LR(1)- Problems related to CLR - LALR Parser — Problems related to LALR-YACC.	
Unit-4 – Code Generation	***12 Hour***
Intermediate Code Generation- prefix – postfix notation- Quadruple - triple - indirect triples Representation- Syntax tree- Evaluation of expression - Three-address code- Synthesized attributes – Inherited attributes - Intermediate languages – Declarations- Assignment Statements- Boolean Expressions- Case Statements- Back patching – Procedure calls- Code Generation- Issues in the design of code generator- The target machine – Runtime Storage management- A simple Code generator- Code Generation Algorithm- Register and Address Descriptors.	
Unit-5 – Code Optimization	***12 Hour***
Code optimization -Principal Sources of Optimization- Function Preserving Transformation- Loop Optimization- Peephole optimization — DAG- Basic Blocks- Flow Graphs- Global Data Flow Analysis — Efficient Data Flow Algorithm- Runtime Environments- Source Language issues- Storage Organization- Activation Records- Storage Allocation strategies.	

Lab Experiments	
Lab 1 - Implementation of Lexical Analyzer *Lab 2 conversion from Regular Expression to NFA* *Lab 3 Conversion from NFA to DFA* *Lab 4 Elimation of Ambiguity, Left Recursion and Left Factoring* *Lab 5 -FIRST AND FOLLOW computation* *Lab 6 Predictive Parsing Table* *Lab 7 - Shift Reduce Parsing* *Lab 8- Computation of LEADING AND TRAILING*	*Lab 9 Computation of LR (0) items* *Lab 10-Intermediate code generation – Postfix, Prefix* *Lab 11 Intermediate code generation – Quadruple, Triple, Indirect triple* *Lab 12: A simple code Generator* *Lab 13 Implementation of DAG* *Lab 14: Implementation of Global Data Flow Analysis* *Lab 15: Implement any one storage allocation strategies (heap, stack, static)*

Learning Resources	1. *Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles, Techniques and Tools, Second Edition, Pearson Education, 2011.* 2. *S. Godfrey Winster, S. Aruna Devi, R.Sujatha,"Compiler Design", Yesdee Publishing Pvt.Ltd, 2016.* 3. *K .Muneeswaran,"CompilerDesign", Oxford Higher Education, Fourth Edition, 2015.*	4. *David Galles, "Modern Compiler Design", Pearson Education, Reprit 2012.* 5. *Raghavan V., "Principles of CompilerDesign", Tata McGraw Hill Education Pvt. Ltd., 2010.*

Learning Assessment

	Bloom's Level of Thinking	*Continuous Learning Assessment (CLA)*				*Summative Final Examination (40% weightage)*	
		Formative CLA-1 Average of unit test (45%)		*Life-Long Learning CLA-2 (15%)*			
		Theory	*Practice*	*Theory*	*Practice*	*Theory*	*Practice*
Level 1	*Remember*	*15%*	-	-	*15%*	*15%*	-
Level 2	*Understand*	*25%*	-	-	*20%*	*25%*	-
Level 3	*Apply*	*30%*	-	-	*25%*	*30%*	-
Level 4	*Analyze*	*30%*	-	-	*25%*	*30%*	-
Level 5	*Evaluate*	-	-	-	*10%*	-	-
Level 6	*Create*	-	-	-	*5%*	-	-
	Total	*100 %*		*100 %*		*100 %*	

Course Designers

Experts from Industry	**Experts from Higher Technical Institutions**	**Internal Experts**
1. *Saranya Baskar, Lead Software Testing Engineer, EPAM Systems India Private Limited, Hyderabad. saranya_baskar@epam.com*	1. *Dr. E. Ilavarasan, Professor, Department of Computer Science and Engineering, Puducherry Technological University, Puducherry.*	1. *Dr. M. Baskar, SRMIST*
	2. *Dr. M. Shyamala Devi, Professor, Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R & D Institute of Science and Technology, Chennai.*	2. *Dr. Godfrey Winster S SRMIST*