ACADEMIC CURRICULA

Professional Core Courses

COMPUTER SCIENCE AND ENGINEERING

Regulations - 2018

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Kattankulathur, Kancheepuram, Tamil Nadu, India

Course Code	18CSC201J	Course Name	DATA STRUCTURES AND ALGORITHMS	Course Category	C	Professional Core	L	T	P	C
							3	0	2	4

Pre-requisite Courses	Nil		Co-requisite Courses	Nil	Progressive Courses	18CSC204J
Course Offering Department	Computer Science and Engineering		Data Book / Codes/Standards		Nil	

Course Learning Rationale (CLR):	The purpose of learning this course is to:
CLR-1 :	Utilize the different data types; Utilize searching and sorting algorithms for data search
CLR-2 :	Utilize linked list in developing applications
CLR-3 :	Utilize stack and queues in processing data for real-time applications
CLR-4 :	Utilize tree data storage structure for real-time applications
CLR-5 :	Utilize algorithms to find shortest data search in graphs for real-time application development
CLR-6 :	Utilize the different types of data structures and its operations for real-time programming applications

		Learning			Program Learning Outcomes (PLO)														
		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Course Learning Outcomes (CLO):	At the end of this course, learners will be able to:	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
CLO-1 :	Identify linear and non-linear data structures. Create algorithms for searching and sorting	3	80	70	L	H	-	H	L	-	-	-	L	L	-	H	-	-	-
CLO-2 :	Create the different types of linked lists and evaluate its operations	3	85	75	M	H	L	M	L	-	-	-	M	L	-	H	-	-	-
CLO-3 :	Construct stack and queue data structures and evaluate its operations	3	75	70	M	H	M	H	L	-	-	-	M	L	-	H	-	-	-
CLO-4 :	Create tree data structures and evaluate its types and operations	3	85	80	M	H	M	H	L	-	-	-	M	L	-	H	-	-	-
CLO-5 :	Create graph data structure, evaluate its operations, implement algorithms to identify shortest path	3	85	75	H	H	M	H	L	-	-	-	M	L	-	H	-	-	-
CLO-6 :	Construct the different data structures and evaluate their types and operations	3	80	70	L	H	-	H	L	-	-	-	L	L	-	H	-	-	-

Duration (hour)		15	15	15	15	15
S-1	SLO-1	Introduction-Basic Terminology	Array	Stack ADT	General Trees	Graph Terminology
	SLO-2	Data Structures	Operations on Arrays – Insertion and Deletion	Stack Array Implementation	Tree Terminologies	Graph Traversal
S-2	SLO-1	Data Structure Operations	Applications on Arrays	Stack Linked List Implementation	Tree Representation	Topological sorting
	SLO-2	ADT	Multidimensional Arrays- Sparse Matrix	Applications of Stack- Infix to Postfix Conversion	Tree Traversal	Minimum spanning tree – Prims Algorithm
S-3	SLO-1	Algorithms – Searching techniques	Linked List Implementation - Insertion	Applications of Stack- Postfix Evaluation	Binary Tree Representation	Minimum Spanning Tree - Kruskal's Algorithm
	SLO-2	Complexity – Time , Space Trade off	Linked List- Deletion and Search	Applications of Stack- Balancing symbols	Expression Trees	Network flow problem
S 4-5	SLO-1 SLO-2	Lab 1: Implementation of Searching - Linear and Binary Search Techniques	Lab 4: Implementation of Array – Insertion, Deletion.	Lab 7 :Implementation of stack using array and Linked List	Lab 10: Implementation of Tree using array	Lab 13: Implementation of Graph using Array
S-6	SLO-1	Algorithms - Sorting	Applications of Linked List	Applications of Stack- Nested Function Calls	Binary Tree Traversal	Shortest Path Algorithm- Introduction
	SLO-2	Complexity – Time , Space Trade off	Polynomial Arithmetic	Recursion concept using stack	Threaded Binary Tree	Shortest Path Algorithm: Dijkstra's Algorithm
S-7	SLO-1	Mathematical notations	Cursor Based Implementation – Methodology	Applications of Recursion: Tower of Hanoi	Binary Search Tree :Construction, Searching	Hashing: Hash functions - Introduction

	SLO-2	*Asymptotic notations-Big O, Omega*	*Cursor Based Implementation*	*Queue ADT*	*Binary Search Tree : Insertion and Deletion*	*Hashing: Hash functions*
S-8	SLO-1	*Asymptotic notations - Theta*	*Circular Linked List*	*Queue Implementation using array*	*AVL Trees: Rotations*	*Hashing : Collision avoidance*
	SLO-2	*Mathematical functions*	*Circular Linked List - Implementation*	*Queue Implementation using Linked List*	*AVL Tree: Insertions*	*Hashing : Separate chaining*
S 9-10	SLO-1	*Lab 2: Implementation of sorting Techniques – Insertion sort and Bubble Sort Techniques*	*Lab 5: Implementation of Linked List - Cursor Based Implementation*	*Lab 8: Implementation of Queue using Array and linked list*	*Lab 11: Implementation of BST using linked list*	*Lab 14 :Implementation of Shortest path Algorithm*
	SLO-2					
S-11	SLO-1	*Data Structures and its Types*	*Applications of Circular List -Joseph Problem*	*Circular Queue*	*B-Trees Constructions*	*Open Addressing*
	SLO-2	*Linear and Non-Linear Data Structures*	*Doubly Linked List*	*Implementation of Circular Queue*	*B-Trees Search*	*Linear Probing*
S-12	SLO-1	*1D, 2D Array Initialization using Pointers*	*Doubly Linked List Insertion*	*Applications of Queue*	*B-Trees Deletions*	*Quadratic probing*
	SLO-2	*1D, 2D Array Accessing using Pointers*	*Doubly Linked List Insertion variations*	*Double ended queue*	*Splay Trees*	*Double Hashing*
S-13	SLO-1	*Declaring Structure and accessing*	*Doubly Linked List Deletion*	*Priority Queue*	*Red Black Trees*	*Rehashing*
	SLO-2	*Declaring Arrays of Structures and accessing*	*Doubly Linked List Search*	*Priority Queue - Applications*	*Red Black Trees Insertion*	*Extensible Hashing*
S 14-15	SLO-1	*Lab 3: Implement Structures using Pointers*	*Lab 6: Implementation of Doubly linked List*	*Lab 9: Applications of Stack, Queue*	*Lab 12:Implementation of B-Trees*	*Lab 15 :Implementation of Minimal Spanning Tree*
	SLO-2					

Learning Resources	1. *Seymour Lipschutz, Data Structures with C, McGraw Hill, 2014* 2. *R.F.Gilberg, B.A.Forouzan, Data Structures, 2nd ed., Thomson India, 2005* 3. *A.V.Aho, J.E Hopcroft , J.D.Ullman, Data structures and Algorithms, Pearson Education, 2003* 4. *Mark Allen Weiss, Data Structures and Algorithm Analysis in C, 2nd ed., Pearson Education, 2015*	5. *Reema Thareja, Data Structures Using C, 1st ed., Oxford Higher Education, 2011* 6. *Thomas H Cormen, Charles E Leiserson, Ronald L Revest, Clifford Stein, Introduction to Algorithms 3rd ed., The MIT Press Cambridge, 2014*

Learning Assessment

	Bloom's Level of Thinking	Continuous Learning Assessment (50% weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember Understand	*20%*	*20%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*
Level 2	Apply Analyze	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*
Level 3	Evaluate Create	*10%*	*10%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*
	Total	100 %		100 %		100 %		100 %		100 %	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

Course Designers

Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
1. Dr. Nagaveer, CEO, Campus Corporate Connect, nagaveer@campuscorporateconnect.com	*1. Dr. Srinivasa Rao Bakshi, IITM, Chennai, sbakshi@iitm.ac.in*	*1. Mr. K. Venkatesh, SRMIST*
2. Dr. Sricharan Srinivasan, Wipro Technologies, sricharanms@gmail.com	*2. Dr. Ramesh Babu, N , nrbabu@iitm.ac.in*	*2. Dr.Subalalitha C.N, SRMIST*
	3. Dr.Noor Mahammad, IIITDM, Kancheepuram, noor@iiitdm.ac.in	*3. Ms. Ferni Ukrit, SRMIST*

Course Code	18CSC202J	Course Name	OBJECT ORIENTED DESIGN AND PROGRAMMING	Course Category	*C*	*Professional Core*	L	T	P	C
							3	0	2	4

Pre-requisite Courses	*18CSS101J*	Co-requisite Courses	*Nil*	Progressive Courses	*18CSC207J*
Course Offering Department	*Computer Science and Engineering*	Data Book / Codes/Standards	*Nil*		

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*
CLR-1 :	*Utilize class and build domain model for real-time programs*
CLR-2 :	*Utilize method overloading and operator overloading for real-time application development programs*
CLR-3 :	*Utilize inline, friend and virtual functions and create application development programs*
CLR-4 :	*Utilize exceptional handling and collections for real-time object oriented programming applications*
CLR-5 :	*Construct UML component diagram and deployment diagram for design of applications*
CLR-6 :	*Create programs using object oriented approach and design methodologies for real-time application development*

		Learning			Program Learning Outcomes (PLO)														
		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Course Learning Outcomes (CLO):	*At the end of this course, learners will be able to:*	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
CLO-1 :	*Identify the class and build domain model*	*3*	*80*	*70*	*H*	*H*	*M*	*-*	*-*	*-*	*-*	*-*	*H*	*H*	*-*	*-*	*M*	*H*	*H*
CLO-2 :	*Construct programs using method overloading and operator overloading*	*3*	*85*	*75*	*H*	*H*	*H*	*H*	*H*	*-*	*M*	*-*	*H*	*H*	*-*	*-*	*M*	*H*	*H*
CLO-3 :	*Create programs using inline, friend and virtual functions, construct programs using standard templates*	*3*	*75*	*70*	*H*	*H*	*M*	*H*	*H*	*-*	*M*	*-*	*H*	*H*	*-*	*-*	*M*	*H*	*H*
CLO-4 :	*Construct programs using exceptional handling and collections*	*3*	*85*	*80*	*H*	*H*	*H*	*-*	*-*	*-*	*-*	*-*	*H*	*M*	*-*	*-*	*M*	*H*	*H*
CLO-5 :	*Create UML component diagram and deployment diagram*	*3*	*85*	*75*	*H*	*M*	*M*	*M*	*M*	*M*	*M*	*-*	*H*	*H*	*-*	*M*	*M*	*H*	*H*
CLO-6 :	*Create programs using object oriented approach and design methodologies*	*3*	*80*	*70*	*H*	*H*	*M*	*-*	*-*	*-*	*-*	*-*	*H*	*H*	*-*	*-*	*M*	*H*	*H*

Duration (hour)		15	15	15	15	15
S-1	SLO-1	*Comparison of Procedural and Object Oriented Programming*	*Types of constructor (Default, Parameter)*	*Feature Inheritance: Single and Multiple*	*Generic - Templates : Introduction*	*STL: Containers: Sequence and Associative Container*
	SLO-2	*OOPS and its features*	*Static constructor and copy constructor*	*Inheritance: Multilevel*	*Function templates*	
S-2	SLO-1	*I/O Operations, Data Types, Variables, static*	*Feature Polymorphism: Constructor overloading*	*Inheritance: Hierarchical*	*Example programs Function templates*	*Sequence Container: Vector, List*
	SLO-2	*Constants, Pointers, Type Conversions*	*Method Overloading*	*Inheritance: Hybrid*	*Class Templates*	*Sequence Container: Deque, Array*
S-3	SLO-1	*Features: Class and Objects*	*Example for method overloading*	*Inheritance: Example Programs*	*Class Templates*	*STL : Stack*
	SLO-2	*UML Diagrams Introduction*	*Method Overloading: Different parameter with different return values*		*Example programs for Class and Function templates*	
S 4-5	SLO-1	*Lab 1: I/O operations*	*Lab 4: Constructor and Method overloading*	*Lab 7: Inheritance and its types*	*Lab 10: Templates*	*Lab 13: STL Containers*
	SLO-2					
S-6	SLO-1	*Feature :Class and Objects*	*Operator Overloading and types*	*Advanced Functions: Inline, Friend*	*Exceptional Handling: try and catch*	*Associative Containers: Map, Multimap*
	SLO-2	*Examples of Class and Objects*	*Overloading Assignment Operator*	*Advanced Functions: Virtual, Overriding*	*Exceptional Handling: Multilevel exceptional*	
S-7	SLO-1	*UML Class Diagram and its components*	*Overloading Unary Operators*	*Advanced Function: Pure Virtual function*	*Exceptional Handling: throw and throws*	*Iterator and Specialized iterator*
	SLO-2	*Class Diagram relations and Multiplicity*	*Example for Unary Operator overloading*	*Example for Virtual and pure virtual function*	*Exceptional Handling: finally*	*Functions of iterator*

S-8	SLO-1	*Feature Abstraction and Encapsulation*	*Overloading Binary Operators*	*Abstract class and Interface*	*Exceptional Handling: User defined exceptional*	*Algorithms: find(), count(), sort()*
	SLO-2	*Application of Abstraction and Encapsulation*	*Example for Binary Operator overloading*	*Example Program*	*Example Programs using C++*	*Algorithms: search(), merge()*
S 9-10	SLO-1	*Lab 2: Classes and Objects, Class Diagram*	*Lab 5: Polymorphism : Operators Overloading*	*Lab 8: Virtual Function and Abstract class*	*Lab 11: Exceptional Handling*	*Lab 15: STL Associative containers and algorithms*
	SLO-2					
S-11	SLO-1	*Access specifiers – public, private*	*UML Interaction Diagrams*	*UML State Chart Diagram*	*Dynamic Modeling: Package Diagram*	*Function Object : for_each(), transform()*
	SLO-2	*Access specifiers - protected, friend, inline*	*Sequence Diagram*	*UML State Chart Diagram*	*UML Component Diagram*	*Example for Algorithms*
S-12	SLO-1	*UML use case Diagram, use case, Scenario*	*Collaboration Diagram*	*Example State Chart Diagram*	*UML Component Diagram*	*Streams and Files: Introduction*
	SLO-2	*Use case Diagram objects and relations*	*Example Diagram*	*UML Activity Diagram*	*UML Deployment Diagram*	*Classes and Errors*
S-13	SLO-1	*Method, Constructor and Destructor*	*Feature: Inheritance*	*UML Activity Diagram*	*UML Deployment Diagram*	*Disk File Handling Reading Data and Writing Data*
	SLO-2	*Example program for constructor*	*Inheritance and its types*	*Example Activity Diagram*	*Example Package, Deployment, Package*	
S 14-15	SLO-1	*Lab 3: Methods and Constructor, Usecase*	*Lab 6: UML Interaction Diagram*	*Lab 9: State Chart and Activity Diagram*	*Lab12 : UML Component, Deployment, Package diagram*	*Lab15: Streams and File Handling*
	SLO-2					

Learning Resources	1. *Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Object-Oriented Analysis and Design with Applications, 3rd ed., Addison-Wesley, May 2007* 2. *Reema Thareja, Object Oriented Programming with C++, 1st ed., Oxford University Press, 2015* 3. *Sourav Sahay, Object Oriented Programming with C++, 2nd ed., Oxford University Press, 2017*	4. *Robert Lafore, Object-Oriented Programming in C++, 4th ed., SAMS Publishing, 2008* 5. *Ali Bahrami, Object Oriented Systems Development", McGraw Hill, 2004* 6. *Craig Larmen, Applying UML and Patterns, 3rd ed., Prentice Hall, 2004*

Learning Assessment

	Bloom's Level of Thinking	Continuous Learning Assessment (50% weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember Understand	*20%*	*20%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*
Level 2	Apply Analyze	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*
Level 3	Evaluate Create	*10%*	*10%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*
	Total	100 %		100 %		100 %		100 %		100 %	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc

For the laboratory component the students are advised to take an application and apply the concepts

Course Designers

Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
Mr. Girish Raghavan, Senior DMTS Member, Wipro Ltd.	*1. Dr. Srinivasa Rao Bakshi, IITM Chennai, sbakshi@iitm.ac.in*	*1. Ms. C.G.Anupama, SRMIST*
Ms. Thamilchelvi, Solutions Architect, Wipro Ltd	*2. Dr. Ramesh Babu, N, IITM Chennai, nrbabu@iitm.ac.in*	*2. Mr. C.Arun, SRMIST*
		3. Mr. Geogen George, SRMIST
		4. Mr. Muthukumaran, SRMIST

							L	T	P	C
Course Code	18CSC203J	Course Name	COMPUTER ORGANIZATION AND ARCHITECTURE	Course Category	C	*Professional Core*	3	0	2	4

Pre-requisite Courses	*Nil*	Co-requisite Courses	*Nil*	Progressive Courses	*18CSC207J*
Course Offering Department	*Computer Science and Engineering*	Data Book / Codes/Standards	*Nil*		

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*
CLR-1 :	*Utilize the functional units of a computer*
CLR-2 :	*Analyze the functions of arithmetic Units like adders, multipliers etc.*
CLR-3 :	*Understand the concepts of Pipelining and basic processing units*
CLR-4 :	*Study about parallel processing and performance considerations.*
CLR-5 :	*Have a detailed study on Input-Output organization and Memory Systems.*
CLR-6 :	*Simulate simple fundamental units like half adder, full adder etc*

		Learning			Program Learning Outcomes (PLO)														
		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Course Learning Outcomes (CLO):	*At the end of this course, learners will be able to:*	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
CLO-1 :	*Identify the computer hardware and how software interacts with computer hardware*	*2*	*80*	*70*	*H*	*H*	*-*	*-*	*-*	*-*	*-*	*-*	*M*	*L*	*-*	*M*	*-*	*-*	*-*
CLO-2 :	*Apply Boolean algebra as related to designing computer logic, through simple combinational and sequential logic circuits*	*3*	*85*	*75*	*H*	*H*	*H*	*-*	*H*	*-*	*-*	*-*	*M*	*L*	*-*	*M*	*-*	*-*	*-*
CLO-3 :	*Analyze the detailed operation of Basic Processing units and the performance of Pipelining*	*2*	*75*	*70*	*H*	*H*	*H*	*H*	*-*	*-*	*-*	*-*	*M*	*L*	*-*	*M*	*-*	*-*	*-*
CLO-4 :	*Analyze concepts of parallelism and multi-core processors.*	*3*	*85*	*80*	*H*	*-*	*-*	*H*	*-*	*-*	*-*	*-*	*M*	*L*	*-*	*M*	*-*	*-*	*-*
CLO-5 :	*Identify the memory technologies, input-output systems and evaluate the performance of memory system*	*3*	*85*	*75*	*H*	*-*	*H*	*H*	*-*	*-*	*-*	*-*	*M*	*L*	*-*	*M*	*-*	*-*	*-*
CLO-6 :	*Identify the computer hardware, software and its interactions*	*3*	*85*	*75*	*H*	*H*	*H*	*H*	*H*	*-*	*-*	*-*	*M*	*L*	*-*	*M*	*-*	*-*	*-*

Duration (hour)		15	15	15	15	15
S-1	SLO-1	*Functional Units of a computer*	*Addition and subtraction of Signed numbers*	*Fundamental concepts of basic processing unit*	*Parallelism*	*Memory systems -Basic Concepts*
	SLO-2	*Operational concepts*	*Problem solving*	*Performing ALU operation*	*Need, types of Parallelism*	*Memory hierarchy*
S-2	SLO-1	*Bus structures*	*Design of fast adders*	*Execution of complete instruction, Branch instruction*	*applications of Parallelism*	*Memory technologies*
	SLO-2	*Memory locations and addresses*	*Ripple carry adder and Carry look ahead adder*	*Multiple bus organization*	*Parallelism in Software*	*RAM, Semiconductor RAM*
S-3	SLO-1	*Memory operations*	*Multiplication of positive numbers*	*Hardwired control*	*Instruction level parallelism*	*ROM,Types*
	SLO-2	*Memory operations*	*Problem Solving*	*Generation of control signals*	*Data level parallelism*	*Speed,size cost*
S 4-5	SLO-1	*Lab 1: To recognize various components of PC- Input Output systems Processing and Memory units*	*Lab4:Study of TASM Addition and Subtraction of 8-bit number*	*Lab-7: Design of Half Adder Design of Full Adder*	*Lab-10: Study of Array Multiplier Design of Array Multiplier*	*Lab-13: Study of Carry Save Multiplication Program to carry out Carry Save Multiplication*
	SLO-2					
S-6	SLO-1	*Instructions, Instruction sequencing*	*Signed operand multiplication*	*Micro-programmed control-*	*Challenges in parallel processing*	*Cache memory*
	SLO-2	*Addressing modes*	*Problem solving*	*Microinstruction*	*Architectures of Parallel Systems - Flynn's classification*	*Mapping Functions*
S-7	SLO-1	*Problem solving*	*Fast multiplication- Bit pair recoding of Multipliers*	*Micro-program Sequencing*	*SISD,SIMD*	*Replacement Algorithms*

	SLO-2	Introduction to Microprocessor	Problem Solving	Micro instruction with Next address field	MIMD, MISD	Problem Solving
S-8	SLO-1	Introduction to Assembly language	Carry Save Addition of summands	Basic concepts of pipelining	Hardware multithreading	Virtual Memory
	SLO-2	Writing of assembly language programming	Problem Solving	Pipeline Performance	Coarse Grain parallelism, Fine Grain parallelism	Performance considerations of various memories
S 9-10	SLO-1	Lab-2:To understand how different components of PC are connected to work properly Assembling of System Components	Lab 5: Addition of 16-bit number Subtraction of 16-bit number	Lab-8: Study of Ripple Carry Adder Design of Ripple Carry Adder	Lab-11: Study of Booth Algorithm	Lab-14: Understanding Processing unit Design of primitive processing unit
	SLO-2					
S-11	SLO-1	ARM Processor: The thumb instruction set	Integer division – Restoring Division	Pipeline Hazards-Data hazards	Uni-processor and Multiprocessors	Input Output Organization
	SLO-2	Processor and CPU cores	Solving Problems	Methods to overcome Data hazards	Multi-core processors	Need for Input output devices
S-12	SLO-1	Instruction Encoding format	Non Restoring Division	Instruction Hazards	Multi-core processors	Memory mapped IO
	SLO-2	Memory load and Store instruction in ARM	Solving Problems	Hazards on conditional and Unconditional Branching	Memory in Multiprocessor Systems	Program controlled IO
S-13	SLO-1	Basics of IO operations.	Floating point numbers and operations	Control hazards	Cache Coherency in Multiprocessor Systems	Interrupts-Hardware, Enabling and Disabling Interrupts
	SLO-2	Basics of IO operations.	Solving Problems	Influence of hazards on instruction sets	MESI protocol for Multiprocessor Systems	Handling multiple Devices
S 14-15	SLO-1	Lab -3To understand how different components of PC are connected to work properly Disassembling of System Components	Lab-6: Multiplication of 8-bit number Factorial of a given number	Lab-9: Study of Carry Look-ahead Adder Design of Carry Look-ahead Adder	Lab-12: Program to carry out Booth Algorithm	Lab-15: Understanding Pipeline concepts Design of basic pipeline.
	SLO-2					

Learning Resources	1. Carl Hamacher, ZvonkoVranesic, SafwatZaky, Computer Organization, 5th ed., McGraw-Hill, 2015 2. Kai Hwang, Faye A. Briggs, Computer Architecture and Parallel Processing", 3rd ed., McGraw Hill, 2016 3. Ghosh T. K., Computer Organization and Architecture, 3rd ed., Tata McGraw-Hill, 2011 4. P. Hayes, Computer Architecture and Organization, 3rd ed., McGraw Hill, 2015.	5. William Stallings, Computer Organization and Architecture – Designing for Performance, 10th ed., Pearson Education, 2015 6. David A. Patterson and John L. Hennessy Computer Organization and Design - A Hardware software interface, 5th ed., Morgan Kaufmann,2014

Learning Assessment

	Bloom's Level of Thinking	Continuous Learning Assessment (50% weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember Understand	20%	20%	15%	15%	15%	15%	15%	15%	15%	15%
Level 2	Apply Analyze	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%
Level 3	Evaluate Create	10%	10%	15%	15%	15%	15%	15%	15%	15%	15%
	Total	100 %		100 %		100 %		100 %		100 %	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

Course Designers

Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
1. T. V. Sankar, HCL Technologies Ltd, Chennai, sankar_t@hcl.com	1. Prof. A.P. Shanthi, ANNA University Chennai, a.p.shanthi@cs.annauniv.edu	1.Dr. V. Ganapathy, SRMIST
		2. Dr. C. Malathy, SRMIST
		3. Mrs M.S.Abirami, SRMIST

Course Code	18CSC204J	Course Name	DESIGN AND ANALYSIS OF ALGORITHMS	Course Category	C	*Professional Core*	L	T	P	C
							3	0	2	4

Pre-requisite Courses	*18CSC201J, 18CSC202J*	Co-requisite Courses	*18CSC207J*	Progressive Courses	*Nil*
Course Offering Department	*Computer Science and Engineering*	Data Book / Codes/Standards	*Nil*		

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*
CLR-1 :	*Design efficient algorithms in solving complex real time problems*
CLR-2 :	*Analyze various algorithm design techniques to solve real time problems in polynomial time*
CLR-3 :	*Utilize various approaches to solve greedy and dynamic algorithms*
CLR-4 :	*Utilize back tracking and branch and bound paradigms to solve exponential time problems*
CLR-5 :	*Analyze the need of approximation and randomization algorithms, utilize the importance Non polynomial algorithms*
CLR-6 :	*Construct algorithms that are efficient in space and time complexities*

		Learning			Program Learning Outcomes (PLO)														
		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Course Learning Outcomes (CLO):	*At the end of this course, learners will be able to:*	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
CLO-1 :	*Apply efficient algorithms to reduce space and time complexity of both recurrent and non-recurrent relations*	*3*	*80*	*70*	*L*	*H*	*-*	*H*	*L*	*-*	*-*	*-*	*L*	*L*	*-*	*H*	*-*	*-*	*-*
CLO-2 :	*Solve problems using divide and conquer approaches*	*3*	*85*	*75*	*M*	*H*	*L*	*M*	*L*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*-*	*-*	*-*
CLO-3 :	*Apply greedy and dynamic programming types techniques to solve polynomial time problems.*	*3*	*75*	*70*	*M*	*H*	*M*	*H*	*L*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*-*	*-*	*-*
CLO-4 :	*Create exponential problems using backtracking and branch and bound approaches.*	*3*	*85*	*80*	*M*	*H*	*M*	*H*	*L*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*-*	*-*	*-*
CLO-5 :	*Interpret various approximation algorithms and interpret solutions to evaluate P type, NP Type, NPC, NP Hard problems*	*3*	*85*	*75*	*H*	*H*	*M*	*H*	*L*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*-*	*-*	*-*
CLO-6 :	*Create algorithms that are efficient in space and time complexities by using divide conquer, greedy, backtracking technique*	*3*	*80*	*70*	*L*	*H*	*M*	*H*	*L*	*-*	*-*	*-*	*L*	*L*	*-*	*H*	*-*	*-*	*-*

Duration (hour)		15	15	15	15	15
S-1	SLO-1	*Introduction-Algorithm Design*	*Introduction-Divide and Conquer*	*Introduction-Greedy and Dynamic Programming*	*Introduction to backtracking - branch and bound*	*Introduction to randomization and approximation algorithm*
	SLO-2	*Fundamentals of Algorithms*	*Maximum Subarray Problem*	*Examples of problems that can be solved by using greedy and dynamic approach*	*N queen's problem - backtracking*	*Randomized hiring problem*
S-2	SLO-1	*Correctness of algorithm*	*Binary Search*	*Huffman coding using greedy approach*	*Sum of subsets using backtracking*	*Randomized quick sort*
	SLO-2	*Time complexity analysis*	*Complexity of binary search*	*Comparison of brute force and Huffman method of encoding*	*Complexity calculation of sum of subsets*	*Complexity analysis*
S-3	SLO-1	*Insertion sort-Line count, Operation count*	*Merge sort*	*Knapsack problem using greedy approach*	*Graph introduction*	*String matching algorithm*
	SLO-2	*Algorithm Design paradigms*	*Time complexity analysis*	*Complexity derivation of knapsack using greedy*	*Hamiltonian circuit - backtracking*	*Examples*
S 4-5	SLO-1	*Lab 1: Simple Algorithm-Insertion sort*	*Lab 4: Quicksort, Binary search*	*Lab 7: Huffman coding, knapsack and using greedy*	*Lab 10: N queen's problem*	*Lab 13: Randomized quick sort*
	SLO-2					
S-6	SLO-1	*Designing an algorithm*	*Quick sort and its Time complexity analysis*	*Tree traversals*	*Branch and bound - Knapsack problem*	*Rabin Karp algorithm for string matching*
	SLO-2	*And its analysis-Best, Worst and Average case*	*Best case, Worst case, Average case analysis*	*Minimum spanning tree – greedy Kruskal's algorithm - greedy*	*Example and complexity calculation. Differentiate with dynamic and greedy*	*Example discussion*
S-7	SLO-1	*Asymptotic notations Based on growth functions.*	*Strassen's Matrix multiplication and its recurrence relation*	*Minimum spanning tree - Prims algorithm*	*Travelling salesman problem using branch and bound*	*Approximation algorithm*

	SLO-2	*O,O,ϴ, ω, Ω*	*Time complexity analysis of Merge sort*	*Introduction to dynamic programming*	*Travelling salesman problem using branch and bound example*	*Vertex covering*
S-8	SLO-1	*Mathematical analysis*	*Largest sub-array sum*	*0/1 knapsack problem*	*Travelling salesman problem using branch and bound example*	*Introduction Complexity classes*
	SLO-2	*Induction, Recurrence relations*	*Time complexity analysis of Largest sub-array sum*	*Complexity calculation of knapsack problem*	*Time complexity calculation with an example*	*P type problems*
S 9-10	SLO-1 SLO-2	*Lab 2: Bubble Sort*	*Lab 5: Strassen Matrix multiplication*	*Lab 8: Various tree traversals, Krukshall's MST*	*Lab 11: Travelling salesman problem*	*Lab 14: String matching algorithms*
S-11	SLO-1	*Solution of recurrence relations*	*Master Theorem Proof*	*Matrix chain multiplication using dynamic programming*	*Graph algorithms*	*Introduction to NP type problems*
	SLO-2	*Substitution method*	*Master theorem examples*	*Complexity of matrix chain multiplication*	*Depth first search and Breadth first search*	*Hamiltonian cycle problem*
S-12	SLO-1	*Solution of recurrence relations*	*Finding Maximum and Minimum in an array*	*Longest common subsequence using dynamic programming*	*Shortest path introduction*	*NP complete problem introduction*
	SLO-2	*Recursion tree*	*Time complexity analysis-Examples*	*Explanation of LCS with an example*	*Floyd-Warshall Introduction*	*Satisfiability problem*
S-13	SLO-1	*Solution of recurrence relations*	*Algorithm for finding closest pair problem*	*Optimal binary search tree (OBST)using dynamic programming*	*Floyd-Warshall with sample graph*	*NP hard problems*
	SLO-2	*Examples*	*Convex Hull problem*	*Explanation of OBST with an example.*	*Floyd-Warshall complexity*	*Examples*
S 14-15	SLO-1 SLO-2	*Lab 3: Recurrence Type-Merge sort, Linear search*	*Lab 6: Finding Maximum and Minimum in an array, Convex Hull problem*	*Lab 9: Longest common subsequence*	*Lab 12: BFS and DFS implementation with array*	*Lab 15: Discussion over analyzing a real time problem*

Learning Resources	1. *Thomas H Cormen, Charles E Leiserson, Ronald L Revest, Clifford Stein, Introduction to Algorithms, 3rd ed., The MIT Press Cambridge, 2014* 2. *Mark Allen Weiss, Data Structures and Algorithm Analysis in C, 2nd ed., Pearson Education, 2006*	3. *Ellis Horowitz, Sartajsahni, Sanguthevar, Rajesekaran, Fundamentals of Computer Algorithms, Galgotia Publication, 2010* 4. *S. Sridhar, Design and Analysis of Algorithms, Oxford University Press, 2015*

Learning Assessment

	Bloom's Level of Thinking	Continuous Learning Assessment (50% weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	*20%*	*20%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*
	Understand										
Level 2	Apply	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*
	Analyze										
Level 3	Evaluate	*10%*	*10%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*
	Create										
	Total	100 %		100 %		100 %		100 %		100 %	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

Course Designers

Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
1. G. Venkiteswaran, Wipro Technologies, gvenki@pilani.bits-pilani.ac.in	*1. Mitesh Khapra, IITM Chennai, miteshk@cse.iitm.ac.in*	*1. Mr.K.Senthil Kumar, SRMIST*
2. Dr.Sainarayanan Gopalakrishnan, HCL Technologies, sai.jgk@gmail.com	*2. V. Masilamani. IIITDM, masila@iiitdm.ac.in*	*2. Dr.A.Razia Sulthana, SRMIST*
		3. Mr. V. Sivakumar, SRMIST
		4. Ms. R. Vidhya, SRMIST

Course Code	18CSC205J	Course Name	OPERATING SYSTEMS	Course Category	C	Professional Core	L	T	P	C
							3	0	2	4

Pre-requisite Courses	*Nil*	Co-requisite Courses	*Nil*	Progressive Courses	*Nil*
Course Offering Department	*Computer Science and Engineering*	Data Book / Codes/Standards	*Nil*		

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*	Learning			Program Learning Outcomes (PLO)														
		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CLR-1 :	*Introduce the key role of an Operating system*																		
CLR-2 :	*Insist the Process Management functions of an Operating system*																		
CLR-3 :	*Emphasize the importance of Memory Management concepts of an Operating system*																		
CLR-4 :	*Realize the significance of Device Management part of an Operating system*																		
CLR-5 :	*Comprehend the need of File Management functions of an Operating system*																		
CLR-6 :	*Explore the services offered by the Operating system practically*																		
Course Learning Outcomes (CLO):	*At the end of this course, learners will be able to:*	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
CLO-1 :	*Identify the need of an Operating system*	*1*	*80*	*70*	*H*	*H*	*H*	*H*	*H*	*M*	*L*	*M*	*H*	*M*	*M*	*H*	*H*	*H*	*M*
CLO-2 :	*Know the Process management functions of an Operating system*	*1*	*85*	*75*	*H*	*H*	*H*	*H*	*H*	*M*	*L*	*M*	*H*	*M*	*M*	*H*	*H*	*H*	*M*
CLO-3 :	*Understand the need of Memory Management functions of an Operating system*	*1*	*75*	*70*	*H*	*H*	*H*	*H*	*H*	*M*	*L*	*M*	*H*	*M*	*M*	*H*	*H*	*H*	*M*
CLO-4 :	*Find the significance of Device management role of an Operating system*	*2*	*85*	*80*	*H*	*H*	*H*	*H*	*H*	*M*	*L*	*M*	*H*	*M*	*M*	*H*	*H*	*H*	*M*
CLO-5 :	*Recognize the essentials of File Management part of an Operating system*	*2*	*85*	*75*	*H*	*H*	*H*	*H*	*H*	*M*	*L*	*M*	*H*	*M*	*M*	*H*	*H*	*H*	*M*
CLO-6 :	*Gain an insight of Importance of an Operating system through practical*	*3*	*80*	*70*	*H*	*H*	*H*	*H*	*H*	*M*	*L*	*M*	*H*	*M*	*M*	*H*	*H*	*H*	*M*

Duration (hour)		15	15	15	15	15
S-1	SLO-1	*Operating System Objectives and functions*	*PROCESS SYNCHRONIZATION : Peterson's solution, Synchronization Hardware*	*MEMORY MANAGEMENT: Memory Management: Logical Vs Physical address space, Swapping*	*VIRTUAL MEMORY– Background*	*STORAGE MANAGEMENT :* *Mass storage structure – Overview of Mass storage structure – Magnetic Disks*
	SLO-2	*Gaining the role of Operating systems*	*Understanding the two-process solution and the benefits of the synchronization hardware*	*Understanding the basics of Memory management*	*Understanding the need of demand paging*	*Understanding the Basics in storage management*
S-2	SLO-1	*The evolution of operating system, Major achievements*	*Process synchronization: Semaphores, usage, implementation*	*Contiguous Memory allocation – Fixed and Dynamic partition*	*VIRTUAL MEMORY – Basic concepts – page fault handling*	*Disk Scheduling*
	SLO-2	*Understanding the evolution of Operating systems from early batch processing systems to modern complex systems*	*Gaining the knowledge of the usage of the semaphores for the Mutual exclusion mechanisms*	*Getting to know about Partition memory management and issues: Internal fragmentation and external fragmentation problems*	*Understanding , how an OS handles the page faults*	*Understanding the various scheduling with respect to the disk*
S-3	SLO-1	*OS Design considerations for Multiprocessor and Multicore*	*Classical Problems of synchronization – Readers writers problem, Bounded Buffer problem*	*Strategies for selecting free holes in Dynamic partition*	*Performance of Demand paging*	*FILE SYSTEM INTERFACE: File concept, File access methods*
	SLO-2	*Understanding the key design issues of Multiprocessor Operating systems and*	*Good understanding of synchronization mechanisms*	*Understanding the allocation strategies with examples*	*Understanding the relationship of effective access time and the page fault rate*	*Understanding the file basics*

		Multicore Operating systems				
S 4-5	SLO-1	LAB 1 : Understanding the booting process of Linux	LAB4 : System admin commands – Basics	LAB7: Shell Programs – Basic level	LAB10 : Overlay concept	LAB13:Process synchronization
	SLO-2					
S-6	SLO-1	PROCESS CONCEPT– Processes, PCB	Classical Problems of synchronization – Dining Philosophers problem (Monitor)	Paged memory management	Copy-on write	File sharing and Protection
	SLO-2	Understanding the Process concept and Maintanance of PCB by OS	Understanding synchronization of limited resources among multiple processes	Understanding the Paging technique.PMT hardware mechanism	Understanding the need for Copy-on write	Emphasis the need for the file sharing and its protection
S-7	SLO-1	Threads – Overview and its Benefits	CPU SCHEDULING : FCFS,SJF,Priority	Structure of Page Map Table	Page replacement Mechanisms: FIFO, Optimal, LRU and LRU approximation Techniques	FILE SYSTEM IMPLEMENTATION : File system structure
	SLO-2	Understanding the importance of threads	Understanding the scheduling techniques	Understanding the components of PMT	Understanding the Pros and cons of the page replacement techniques	To get the basic file system structure
S-8	SLO-1	Process Scheduling : Scheduling Queues, Schedulers, Context switch	CPU Scheduling: Round robin, Multilevel queue Scheduling, Multilevel feedback Scheduling	Example : Intel 32 bit and 64 –bit Architectures	Counting based page replacement and Page Buffering Algorithms	Directory Implementation
	SLO-2	Understanding basics of Process scheduling	Understanding the scheduling techniques	Understanding the Paging in the Intel architectures	To know on additional Techniques available for page replacement strategies	Understanding the various levels of directory structure
S 9-10	SLO-1	LAB2 : Understanding the Linux file system	LAB5: System admin commands – Simple task automations	LAB 8: Process Creation	LAB11: IPC using Pipes	LAB14 : Study of OS161
	SLO-2					
S-11	SLO-1	Operations on Process – Process creation, Process termination	Real Time scheduling: Rate Monotonic Scheduling and Deadline Scheduling	Example : ARM Architectures	Allocation of Frames - Global Vs Local Allocation	FILE SYSTEM IMPLEMENTATION :Allocation methods
	SLO-2	Understanding the system calls – fork(),wait(),exit()	Understanding the real time scheduling	Understanding the Paging with respect to ARM	Understanding the root cause of the Thrashing	Understanding the pros and Cons of various disk allocation methods
S-12	SLO-1	Inter Process communication : Shared Memory, Message Passing ,Pipe()	DEADLOCKS: Necessary conditions, Resource allocation graph, Deadlock prevention methods	Segmented memory management	Thrashing, Causes of Thrashing	FILE SYSTEM IMPLEMENTATION :Free space Management
	SLO-2	Understanding the need for IPC	Understanding the deadlock scenario	Understanding the users view of memory with respect to the primary memory	Understanding the Thrashing	Understanding the methods available for maintaining the free spaces in the disk
S-13	SLO-1	PROCESS SYNCHRONIZATION: Background, Critical section Problem	Deadlocks :Deadlock Avoidance, Detection and Recovery	Paged segmentation Technique	Working set Model	Swap space Management
	SLO-2	Understanding the race conditions and the need for the Process synchronization	Understanding the deadlock avoidance, detection and recovery mechanisms	Understanding the combined scheme for efficient management	Understanding the working set model for controlling the Working set Model	Understanding the Low-level task of the OS
S 14-15	SLO-1	LAB3: Understanding the various Phases of Compilation of a 'C' Program	LAB6 : Linux commands	LAB9: Overlay concept	LAB12: IPC using shared memory and Message queues	LAB15 : Understanding the OS161 filesystem and working with test programs
	SLO-2					

Learning Resources	1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating systems, 9th ed., John Wiley & Sons, 2013 2. William Stallings, Operating Systems-Internals and Design Principles, 7th ed., Prentice Hall, 2012	3. Andrew S. Tanenbaum, Herbert Bos, Modern Operating systems, 4th ed., Pearson, 2015 4. Bryant O'Hallaxn, Computer systems- A Programmer's Perspective, Pearson, 2015

Learning Assessment											
	Bloom's Level of Thinking	Continuous Learning Assessment (50% weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	*20%*	*20%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*
	Understand										
Level 2	Apply	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*
	Analyze										
Level 3	Evaluate	*10%*	*10%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*
	Create										
	Total	100 %		100 %		100 %		100 %		100 %	

Course Designers			
Experts from Industry	Experts from Higher Technical Institutions	Internal Experts	
1. Mr. Balamurugan, Infosys, balams@gmail.com	*1. Dr. Latha Parthiban, Pondicherry University, lathaparthiban@yahoo.com*	*1. Dr.G.Maragatham, SRMIST*	*3. Ms. Aruna S, SRMIST*
		2. Mr. Eliazer M, SRMIST	

Course Code	18CSC206J	Course Name	SOFTWARE ENGINEERING AND PROJECT MANAGEMENT	Course Category	C	*Professional Core*	L	T	P	C
							3	0	2	4

Pre-requisite Courses	*Nil*	Co-requisite Courses	*Nil*	Progressive Courses	*Nil*
Course Offering Department	*Computer Science and Engineering*	Data Book / Codes/Standards	*Nil*		

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*
CLR-1 :	*Familiarize the software life cycle models and software development process*
CLR-2 :	*Understand the various techniques for requirements, planning and managing a technology project*
CLR-3 :	*Examine basic methodologies for software design, development, testing, closure and implementation*
CLR-4 :	*Understand manage users expectations and the software development team*
CLR-5 :	*Acquire the latest industry knowledge, tools and comply to the latest global standards for project management*

		Learning			Program Learning Outcomes (PLO)														
		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Course Learning Outcomes (CLO):	*At the end of this course, learners will be able to:*	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
CLO-1 :	*Identify the process of project life cycle model and process*	*1*	*85*	*80*	*H*	*H*	*L*	*-*	*-*	*-*	*L*	*-*	*H*	*H*	*M*	*M*	*-*	*-*	*-*
CLO-2 :	*Analyze and specify software requirements through a productive working Relationship with project stakeholders*	*2*	*80*	*75*	*H*	*H*	*H*	*H*	*H*	*-*	*M*	*-*	*H*	*H*	*H-*	*M*	*-*	*-*	*-*
CLO-3 :	*Design the system based on Functional Oriented and Object Oriented Approach for Software Design.*	*3*	*85*	*85*	*H*	*H*	*M*	*H*	*H*	*M*	*M*	*L*	*H*	*H*	*M*	*-*	*-*	*-*	*-*
CLO-4 :	*Develop the correct and robust code for the software products*	*3*	*85*	*85*	*H*	*H*	*H*	*-*	*H*	*-*	*-*	*M*	*H*	*M*	*H*	*-*	*-*	*-*	*-*
CLO-5 :	*Perform by applying the test plan and various testing techniques*	*2*	*85*	*75*	*H*	*M*	*M*	*M*	*M*	*M*	*M*	*-*	*H*	*H*	*-*	*M*	*-*	*-*	*-*

Duration (hour)		15	15	15	15	15
S-1	SLO-1	*Introduction to Software Engineering*	*Software Design - Software Design Fundamentals*	*Software Construction*	*Introduction to testing*	*Product Release*
	SLO-2	*Software Project Management - life cycle activities*	*Design Standards - Design Type*	*Coding Standards*	*Verification*	*Product Release*
S-2	SLO-1	*Traditional – Waterfall, V Model*	*Design model – Architectural design, Software architecture*	*Coding Framework*	*Validation*	*Product Release Management*
	SLO-2	*Prototype, Spiral, RAD*	*Software Design Methods*	*Reviews - Desk checks (Peer Reviews)*	*Test Strategy*	*Product Release Management*
S-3	SLO-1	*Conventional – Agile,*	*Top Down , Bottom Up*	*Walkthroughs*	*Planning*	*Implementation*
	SLO-2	*XP, Scrum*	*Module Division (Refactoring)*	*Code Reviews, Inspections*	*Example: Test Strategy and Planning*	*Implementation*
S 4-5	SLO-1	*Lab1:Identify the Software Project, Create Business Case, Arrive at a Problem Statement*	*Lab 4:Prepare Project Plan based on scope, Find Job roles and responsibilities, Calculate Project effort based on resources*	*Lab 7:State and Sequence Diagram, Deployment Diagram, Sample Frontend Design (UI/UX)*	*Lab 10: Module Implementation (Phase 2), Scrum Master to Induce New Issues in Agile Development*	*Lab 13:Manual Testing*
	SLO-2					
S-6	SLO-1	*Introduction to Requirement Engineering*	*Module Coupling*	*Coding Methods*	*Test Project Monitoring and Control*	*User Training*
	SLO-2	*Requirements Elicitation*	*Component level design*	*Structured Programming*	*Test Project Monitoring and Control*	*Maintenance Introduction*
S-7	SLO-1	*Software Project Effort and cost estimation*	*User Interface Design*	*Object-Oriented Programming*	*Test Project Monitoring and Control*	*Maintenance Types - Corrective*
	SLO-2	*Cost estimation*	*Pattern oriented design*	*Automatic Code Generation*	*Test Project Monitoring and Control*	*Adaptive*
S-8	SLO-1	*Cocomo 1 and 2*	*Web application design*	*Automatic Code Generation*	*Test Project Monitoring and Control*	*Perfective*
	SLO-2	*Cocomo 1 and 2*	*Web application design*	*Automatic Code Generation*	*Test Project Monitoring and Control*	*Preventive*

S 9-10	SLO-1	*Lab 2:Stakeholder and User Description, Identify the appropriate Process Model, Comparative study with Agile Model*	*Lab 5:Prepare the Work, Breakdown Structure based on timelines, Risk Identification and Plan*	*Lab 8:Module Description, Module Implementation (phase 1) Using Agile*	*Lab 11:Module Implementation (Phase 3) Scrum Master to Induce New requirements in Agile Development, Scrum Master to Induce New Issues in Agile Development, Code Documentation*	*Lab 14:User Manual, Analysis of Costing, Effort and Resources*
	SLO-2					
S-11	SLO-1	*Risk Management*	*Design Reuse*	*Software Code Reuse*	*Design –Master test plan, types*	*Maintenance Cost*
	SLO-2	*Risk Management*	*Design Reuse*	*Software Code Reuse*	*Design –Master test plan, types*	*Maintenance Process*
S-12	SLO-1	*Configuration management*	*Concurrent Engineering in Software Design*	*Pair Programming*	*Test Case Management*	*life cycle*
	SLO-2	*Configuration management*	*Concurrent Engineering in Software Design*	*Test-Driven Development*	*Test Case Management*	*Software Release*
S-13	SLO-1	*Project Planning – WBC, planning,*	*Design Life-Cycle Management*	*Configuration Management*	*Test Case Reporting*	*Software Maintenance*
	SLO-2	*scope, risk*	*Design Life-Cycle Management*	*Software Construction Artifacts*	*Test Case Reporting*	*Software Release, Software Maintenance*
S 14-15	SLO-1	*Lab 3:Identify the Requirements, System Requirements, Functional Requirements, Non-Functional Requirements*	*Lab 6:Design a System Architecture, Use Case Diagram, ER Diagram (Database), DFD Diagram (process) (Upto Level 1), Class Diagram (Applied For OOPS based Project), Collaboration Diagram (Applied For OOPS based Project) (Software – Rational Rose)*	*Lab 9:Module Implementation, Scrum Master to Induce New requirements in Agile Development*	*Lab 12:Master Test Plan, Test Case Design (Phase 1)*	*Lab 15: Project Demo and Report Submission with the team*
	SLO-2					

Learning Resources	1. *Roger S. Pressman, Software Engineering – A Practitioner Approach, 6th ed., McGraw Hill, 2005* 2. *Ian Sommerville, Software Engineering, 8th ed., Pearson Education, 2010* 3. *Rajib Mall, Fundamentals of Software Engineering, 4th ed., PHI Learning Private Limited, 2014* 4. *Ramesh, Gopalaswamy, Managing Global Projects, Tata McGraw Hill, 2005*	5. *Ashfaque Ahmed, Software Project Management: a process-driven approach, Boca Raton, Fla: CRC Press, 2012* 6. *Walker Royce, Software Project Management, Pearson Education, 1999* 7. *Jim Smith Agile Project Management: Creating Innovative Products,Pearson 2008*

Learning Assessment											
	Bloom's Level of Thinking	Continuous Learning Assessment (50% weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	*20%*	*20%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*
	Understand										
Level 2	Apply	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*
	Analyze										
Level 3	Evaluate	*10%*	*10%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*
	Create										
	Total	100 %		100 %		100 %		100 %		100 %	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

Course Designers		
Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
1. *Mr. Girish Raghavan, Wipro Technologies*	1. *Dr. Latha Parthiban, Pondicherry University, lathaparthiban@yahoo.com*	1. *Mrs. Sasi Rekha Sankar, SRMIST*
2. Dr.Mariappan Vaithilingam, *Amazon, Bangalore*	2. *V. Masilamani. IIITDM, masila@iiitdm.ac.in*	2. *Dr. T.S.Shiny Angel, SRMIST*
		3. *Mr.N.Arivazhagan, SRMIST*
		4. *Mrs K.R.Jansi, SRMIST*

Course Code	18CSC207J	Course Name	ADVANCED PROGRAMMING PRACTICE	Course Category	C	Professional Core	L	T	P	C
							3	0	2	4

Pre-requisite Courses	*18CSC202J*	Co-requisite Courses	*18CSC204J*	Progressive Courses	*Nil*
Course Offering Department	*Computer Science and Engineering*	Data Book / Codes/Standards	*Nil*		

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*	Learning			Program Learning Outcomes (PLO)														
		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CLR-1 :	*Create Real-time Application Programs using structured, procedural and object oriented programming paradigms*																		
CLR-2 :	*Create Real-time Application Programs using event driven, declarative and imperative programming paradigms*																		
CLR-3 :	*Create Real-time Application Programs using parallel, concurrent and functional programming paradigms*																		
CLR-4 :	*Create Real-time Application Programs using logic, dependent type and network programming paradigms*																		
CLR-5 :	*Create Real-time Application Programs using symbolic, automata based and graphical user interface program paradigm*																		
CLR-6 :	*Create Real-time Application Programs using different programming paradigms using python language*																		
Course Learning Outcomes (CLO):	*At the end of this course, learners will be able to:*	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
CLO-1 :	*Create Programs using structured, procedural and object oriented programming paradigms*	*3*	*85*	*80*	*H*	*H*	*H*	*H*	*H*	*-*	*-*	*L*	*M*	*M*	*L*	*M*	*-*	*M*	*-*
CLO-2 :	*Create Programs using event driven, declarative and imperative programming paradigms*	*3*	*85*	*80*	*H*	*H*	*H*	*H*	*H*	*-*	*-*	*L*	*M*	*M*	*L*	*M*	*-*	*-*	*-*
CLO-3 :	*Create Programs using parallel, concurrent and functional programming paradigms*	*3*	*85*	*80*	*H*	*H*	*H*	*H*	*H*	*-*	*-*	*L*	*M*	*M*	*L*	*M*	*-*	*-*	*-*
CLO-4 :	*Create Programs using logic, dependent type and network programming paradigms*	*3*	*85*	*80*	*H*	*H*	*H*	*H*	*H*	*-*	*-*	*L*	*M*	*M*	*L*	*M*	*-*	*-*	*-*
CLO-5 :	*Create Programs using symbolic, automata based and graphical user interface programming paradigms*	*3*	*85*	*80*	*H*	*H*	*H*	*H*	*H*	*-*	*-*	*L*	*M*	*M*	*L*	*M*	*-*	*-*	*-*
CLO-6 :	*Create Programs using different programming paradigms using python language*	*3*	*85*	*80*	*H*	*H*	*H*	*H*	*H*	*-*	*-*	*L*	*M*	*M*	*L*	*M*	*-*	*-*	*-*

Duration (hour)		15	15	15	15	15
S-1	SLO-1	*Structured Programming Paradigm*	*Event Driven Programming Paradigm*	*Parallel Programming Paradigm*	*Logic Programming Paradigm*	*Symbolic Programming Paradigm*
	SLO-2	*Programming Language Theory*	*Event Object, handler, bind*	*Multi-threading, Multi-Processing*	*First-class function, Higher-order function, Pure functions, Recursion*	*Symbolic Maths, algebraic manipulations, limits, differentiation, integration, series*
S-2	SLO-1	*Bohm-Jacopini structured program theorem*	*Keypress events, Mouse events*	*Serial Processing, Parallel Processing*	*Packages: Kanren, SymPy*	*SymPy usage for symbolic maths*
	SLO-2	*Sequence, selection, decision, iteration, recursion*	*Automatic events from a timer*	*Multiprocessing module in Python*	*PySWIP, PyDatalog*	*Equation Solving, Matrices*
S-3	SLO-1	*Other languages: C, C++, Java, C#, Ruby*	*Other languages: Algol, Javascript, Elm*	*Process class, Pool class*	*Other languages: Prolog, ROOP, Janus*	*Other languages: Aurora, LISP, Wolfram*
	SLO-2	*Demo: Structured Programing in Python*	*Demo: Event Driven Programming in Python*	*Demo: Parallel Programming in Python*	*Demo: Logic Programming in Python*	*Demo: Symbolic Programming in Python*
S 4-5	SLO-1	*Lab 1: Structured Programming*	*Lab 4: Event Driven Programming*	*Lab 7: Parallel Programming*	*Lab 10: Logic Programming*	*Lab 13: Symbolic Programming*
	SLO-2					
S-6	SLO-1	*Procedural Programming Paradigm*	*Declarative Programming Paradigm*	*Concurrent Programming Paradigm*	*Dependent Type Programming Paradigm*	*Automata Based Programming Paradigm*
	SLO-2	*Routines, Subroutines, functions*	*Sets of declarative statements*	*Parallel Vs Concurrent Programming*	*Logic Quantifier: for all, there exists*	*Finite State Machine, deterministic finite automation (dfa), nfa*
S-7	SLO-1	*Using Functions in Python*	*Object attribute, Binding behavior*	*threading, multiprocessing*	*Dependent functions, dependent pairs*	*State transitions using python-automaton*
	SLO-2	*logical view, control flow of procedural*	*Creating Events without describing flow*	*concurrent.futures, gevent, greenlets,*	*Relation between data and its computation*	*Initial state, destination state, event*

		programming in various aspects		celery		(transition)
S-8	SLO-1	Other languages: Bliss, ChucK, Matlab	Other languages: Prolog, Z3, LINQ, SQL	Other languages: ANI, Plaid	Other Languages: Idris, Agda, Coq	Other languages: Forth, Ragel, SCXML
	SLO-2	Demo: creating routines and subroutines using functions in Python	Demo: Declarative Programming in Python	Demo: Concurrent Programming in Python	Demo: Dependent Type Programming in Python	Demo: Automata Based Programming in Python
S 9-10	SLO-1	Lab 2: Procedural Programming	Lab 5: Declarative Programming	Lab 8: Concurrent Programming	Lab 11: Dependent Type Programming	Lab 14: Automata Programming
	SLO-2					
S-11	SLO-1	Object Oriented Programming Paradigm	Imperative Programming Paradigm	Functional Programming Paradigm	Network Programming Paradigm	GUI Programming Paradigm
	SLO-2	Class, Objects, Instances, Methods	Program State, Instructions to change the program state	Sequence of Commands	Socket Programming: TCP & UDP Connection oriented, connectionless	Graphical User Interface (GUI)
S-12	SLO-1	Encapsulation, Data Abstraction	Combining Algorithms and Data Structures	map(), reduce(), filter(), lambda	Sock_Stream, Sock_Dgram, socket(), bind(), recvfrom(), sendto(), listen()	Tkinter, WxPython, JPython
	SLO-2	Polymorphism, Inheritance	Imperative Vs Declarative Programming	partial, functools	Server-Client; send(), recv(), connect(), accept(), read(), write(), close()	WxWidgets, PyQT5
S-13	SLO-1	Constructor, Destructor	Other languages: PHP, Ruby, Perl, Swift	Other languages: F#, Clojure, Haskell	Other languages: PowerShell, Bash, TCL	Other languages: GTK, java-gnome
	SLO-2	Example Languages: BETA, Cecil, Lava Demo: OOP in Python	Demo: Imperative Programming in Python	Demo: Functional Programming in Python	Demo: Socket Programming in Python	Demo: GUI Programming in Python
S 14-15	SLO-1	Lab 3: Object Oriented Programming	Lab 6: Imperative Programming	Lab 9: Functional Programming	Lab 12: Network Programming	Lab 15: GUI Programming
	SLO-2					

Learning Resources	1. Elad Shalom, A Review of Programming Paradigms throughout the History: With a suggestion Toward a Future Approach, Kindle Edition, 2018 2. John Goerzen, Brandon Rhodes, Foundations of Python Network Programming: The comprehensive guide to building network applications with Python, 2nd ed., Kindle Edition, 2010 3. Elliot Forbes, Learning Concurrency in Python: Build highly efficient, robust and concurrent applications, Kindle Edition, 2017	4. Amit Saha, Doing Math with Python: Use Programming to Explore Algebra, Statistics, Calculus and More, Kindle Edition, 2015 5. Alan D Moore, Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter, Kindle Edition, 2018 6. https://www.scipy-lectures.org/

Learning Assessment											
	Bloom's Level of Thinking	Continuous Learning Assessment (50% weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	20%	20%	15%	15%	15%	15%	15%	15%	15%	15%
	Understand										
Level 2	Apply	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%
	Analyze										
Level 3	Evaluate	10%	10%	15%	15%	15%	15%	15%	15%	15%	15%
	Create										
	Total	100 %		100 %		100 %		100 %		100 %	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

Course Designers		
Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
1. Mr. Sagar Sahani, Amadeus Software Labs, Bangalore, hello.sagarsahni@gmail.com	1. Dr. Rajeev Sukumaran, IIT Madras, rajeev@wmail.iitm.ac.in	1. Dr. R. Annie Uthra, SRMIST
2. Mr. Janmajay Singh, Fuji Xerox R&D, Japan, janmajaysingh14@gmail.com	2. Prof. R. Golda Brunet, GCE, goldabrunet@gcessalem.edu.in	2. Dr. Christhu Raj M R, SRMIST
		3. Ms. K. Sornalakshmi, SRMIST
		4. Mr. C. Arun, SRMIST

ACADEMIC CURRICULA

Professional Core Courses

COMPUTER SCIENCE AND ENGINEERING

Regulations - 2018

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Kattankulathur, Kancheepuram, Tamil Nadu, India

Course Code	18CSC301T	Course Name	FORMAL LANGUAGE AND AUTOMATA	Course Category	C	Professional Core	L	T	P	C
							3	0	0	3

Pre-requisite Courses	*Nil*	Co-requisite Courses	*Nil*	Progressive Courses	*Nil*
Course Offering Department	*Computer Science and Engineering*	Data Book / Codes/Standards	*Nil*		

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*	Learning			Program Learning Outcomes (PLO)														
CLR-1 :	*Utilize the mathematics and engineering principles for the basics of Formal Language*	1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CLR-2 :	*Acquire knowledge of Automata and minimize with Regular language's*																		
CLR-3 :	*Acquire knowledge of Context free Grammar and simplify using normal forms*																		
CLR-4 :	*Gain knowledge to push down automata and apply it with CFL*																		
CLR-5 :	*Analyze the methods of turning machine*																		
CLR-6 :	*Analyze and Design the methods of computational complexity*																		
Course Learning Outcomes (CLO):	*At the end of this course, learners will be able to:*	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
CLO-1 :	*Acquire the knowledge of mathematics and engineering principles for the basics of Formal Language*	*3*	*80*	*70*	*M*	*H*	*-*	*H*	*L*	*-*	*-*	*-*	*L*	*L*	*-*	*H*	*-*	*-*	*-*
CLO-2 :	*Acquire the ability to identify specification of a Regular language's with Automata*	*3*	*85*	*75*	*M*	*H*	*L*	*M*	*L*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*-*	*-*	*-*
CLO-3 :	*Acquire knowledge of Context free Grammar and simplify using normal forms*	*3*	*75*	*70*	*M*	*H*	*M*	*H*	*L*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*-*	*-*	*-*
CLO-4 :	*Understand the concepts of push down automata and CFL.*	*3*	*85*	*80*	*M*	*H*	*M*	*H*	*L*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*-*	*-*	*-*
CLO-5 :	*Apply the knowledge to turning machine and its methods*	*3*	*85*	*75*	*H*	*H*	*M*	*H*	*L*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*-*	*-*	*-*
CLO-6 :	*Design the computational and acceptor machines using FA, PDA and Turing machines*	*3*	*80*	*70*	*L*	*H*	*-*	*H*	*L*	*-*	*-*	*-*	*L*	*L*	*-*	*H*	*-*	*-*	*-*

Duration (hour)		11	9	9	9	7
S-1	SLO-1	*Introduction to Automaton*	*Grammars: Introduction: Types of Grammar*	*Pushdown Automata: Definitions Moves*	*Turing Machines: Introduction*	*Undecidability :Basic definitions*
	SLO-2	*Mathematical concepts*	*Context Free Grammars and Languages*	*Instantaneous descriptions*	*Formal definition of Turing machines, Instantaneous descriptions*	*Decidable problems,*
S-2	SLO-1	*Formal Languages: Strings, Languages, Properties*	*Derivations*	*Deterministic pushdown automata*	*Turing Machine as Acceptors*	*Examples of undecidable problems and Problems*
	SLO-2	*Finite Representation : Regular Expressions*	*Ambiguity*	*Problems related to DPDA*	*Problems related to turning machine as Acceptors*	*Rice's Theorem*
S-3	SLO-1	*Problems related to regular expressions*	*Relationship between derivation and derivation trees*	*Non - Deterministic pushdown automata*	*Problems related to turning machine as Acceptors*	*Undecidable problems about Turing Machine- Post's Correspondence Problem*
	SLO-2	*Finite Automata :Deterministic Finite Automata*	*Problems related to Context free Grammar*	*Problems related to NDPDA*		*Problems related to Post's Correspondence Problem*
S-4	SLO-1	*Nondeterministic Finite Automata*	*Simplification of CFG : Elimination of Useless Symbols*	*Problems related to DPDA and NDPDA*	*Turing Machine as a Computing Device*	*Properties of Recursive and Recursively enumerable languages*
	SLO-2	*Finite Automaton with €- moves*			*Problems related to turning Turing Machine as a Computing Device*	
S-5	SLO-1	*Problems related to Deterministic and Nondeterministic Finite Automata*	*Simplification of CFG : Unit productions*	*Pushdown automata to CFL Equivalence*	*Problems related to turning Turing Machine as a Computing Device*	*Introduction to Computational Complexity: Definitions*
	SLO-2	*Problems related to Finite Automaton with €- moves*	*Simplification of CFG : Null productions*	*Problems related to Equivalence of PDA to CFG*		*Time and Space complexity of TMs*

Duration (hour)		11	9	9	9	7
S-6	SLO-1	*Equivalence of NFA and DFA*	*Problems related to Simplification of CFG*	*Problems related to Equivalence of PDA to CFG*	*Techniques for Turing Machine Construction*	*Complexity classes: Class P, Class NP*
	SLO-2	*Heuristics to Convert NFA to DFA*				
S-7	SLO-1	*Equivalence of NDFA's with and without €-moves*	*Chomsky normal form*	*CFL to Pushdown automata Equivalence*	*Considering the state as a tuple* *Considering the tape symbol as a tuple*	*Complexity classes: Introduction to NP-Hardness*
	SLO-2	*Problems related Equivalence of NDFA's with and without €-moves*	*Problems related to CNF*	*Problems related to Equivalence of CFG to PDA*	*Checking off symbols*	*NP Completeness*
S-8	SLO-1	*Minimization of DFA*	*Greiback Normal form*	*Pumping lemma for CFL*	*Modifications of Turing Machine*	
	SLO-2	*Problems related to Minimization of DFA*			*Multi-tape Turing Machine*	
S-9	SLO-1	*Regular Languages : Equivalence of Finite Automata and Regular Languages*	*Problems related to GNF*	*Problems based on pumping Lemma*	*Non-Deterministic Turing Machine*	
	SLO-2	*Equivalence of Finite Automata and Regular Grammars*			*Semi-Infinite Tape Turing Machine*	
S-10	SLO-1	*Problems related to Equivalence of Finite Automata and Regular Languages and Regular Grammars*				
	SLO-2	*Variants of Finite Automata :Two-way Finite Automaton Mealy Machines*				
S-11	SLO-1	*Properties of Regular Languages: Closure Properties*				
	SLO-2	*Set Theoretic Properties & Other Properties*				
	SLO-3	*Pumping Lemma*				

Learning Resources	1. *Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory, Languages and Computations", Second Edition, Pearson Education, 2008.* 2. *Michael Sipser, "Introduction to the Theory of Computation" Cengage Learning, 2012.*	3. *.John.C.Martin, "Introduction to Languages and the Theory of Computation" McGraw-Hill Education, 01- May-2010.* 4. *Kamala Krithivasan, Rama.R," Introduction to Formal Languages, Automata Theory and Computation", Pearson Education India, 01-Sep-2009.* 5. *Peter Linz , "An introduction to formal languages and automata", Jones & Bartlett Learning, 2001.*

Learning Assessment

	Bloom's Level of Thinking	Continuous Learning Assessment (50% weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember Understand	*40 %*	*-*	*30 %*	*-*	*30 %*	*-*	*30 %*	*-*	*30%*	*-*
Level 2	Apply Analyze	*40 %*	*-*	*40 %*	*-*	*40 %*	*-*	*40 %*	*-*	*40%*	*-*
Level 3	Evaluate Create	*20 %*	*-*	*30 %*	*-*	*30 %*	*-*	*30 %*	*-*	*30%*	*-*
	Total	100 %		100 %		100 %		100 %		100 %	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

Course Designers		
Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
		1. *Dr.R.AnnieUthra*
		2. *Dr.Jeyasudha*

Course Code	18CSC302J	Course Name	COMPUTER NETWORKS	Course Category	C	Professional Core	L	T	P	C
							3	0	2	4

Pre-requisite Courses	Nil	Co-requisite Courses	Nil	Progressive Courses	Nil
Course Offering Department	CSE	Data Book / Codes/Standards		Nil	

Course Learning Rationale (CLR):	The purpose of learning this course is to:
CLR-1 :	Describe the importance of various Internet protocols like ARP, RARP, ICMP, Multicasting and multi routing, SCTP
CLR-2 :	Understand the transport layer protocols , application layer protocol and its characteristics
CLR-3 :	Learn and Understand IPV6 technologies
CLR-4 :	Work with client server sockets and develop related applications to communicate with each other.
CLR-5 :	Understand the wide area network protocols
CLR-6 :	Learn the basics of DSL,ATM,HDLC,MPLS

Course Learning Outcomes (CLO):	At the end of this course, learners will be able to:	Learning			Program Learning Outcomes (PLO)														
		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
CLO-1 :	Identify the basics of different types of network and transport layer protocols	3	80	70	L	H	-	H	L	-	-	-	L	L	-	H	-	-	-
CLO-2 :	Design and implement the socket programming	3	85	75	M	H	-	M	L	-	-	-	M	L	-	H	-	-	-
CLO-3 :	Enumerate the types of application layer protocols	3	75	70	M	H	-	H	L	-	-	-	M	L	-	H	-	-	-
CLO-4 :	Analyze and compare the IPv4 and IPv6 protocols	3	85	80	M	H	-	H	L	-	-	-	M	L	-	H	-	-	-
CLO-5 :	Familiarize with wide area technologies	3	85	75	H	H	-	H	L	-	-	-	M	L	-	H	-	-	-
CLO-6 :	Describe the working of DSL,ATM,PPP,	3	80	70	L	H	-	H	L	-	-	-	L	L	-	H	-	-	-

Duration (hour)		15	15	15	15	15
S-1	SLO-1	IP header	Byte ordering	DNS	IPV6 Overview	DSL
	SLO-2	IP fragmentation	Byte ordering conversion functions	DNS in the Internet,	IPV6 Features	Other DSL Technology
S-2	SLO-1	ARP	System calls	DNS Resolution	IPV6 Addressing Modes	DSL Benefits
	SLO-2	RARP	Sockets	DNS Messages	IPV6 Address Types	Cable Technology
S-3	SLO-1	ICMP –introduction	System calls used with Sockets	TELNET	Introduction	Compare DSL Vs Cable
	SLO-2	ICMP-Messages	Iterative and concurrent server	SSH	Address Space Allocation	Frame Relay, VPN
S 4-5	SLO-1	Study of necessary header files with respect to socket programming.	UDP Echo Client Server Communication	Full Duplex Chat Using TCP/IP	ARP implementation Using UDP	Implementation of VPN
	SLO-2					
S-6	SLO-1	Debugging tools	Socket Interface	FTP	Global Unicast Addresses	ATM Introduction
	SLO-2	ICMP package	Structure and Functions of Socket	TFTP	Auto configuration	ATM Cell Format
S-7	SLO-1	UDP Datagram	Remote Procedure Call	WWW Architecture	Renumbering	ATM Layer
	SLO-2	UDP characteristics	RPC Model, Features	WWW Documents	IPV6 Routing Protocols	AAL Layer
S-8	SLO-1	TCP Header	TCP Client Server Program	HTTP	Introduction	ATM Application
	SLO-2	TCP connection establishment process	Input, Output Processing Module	HTTP Request and Reply	IPV6 Packet Format	PPP
S 9-10	SLO-1	Study of Basic Functions of Socket Programming	Concurrent TCP/IP Day-Time Server	Implementation of File Transfer Protocol	Study of IPV6 Addressing & Subnetting	Communication Using HDLC
	SLO-2					
S-11	SLO-1	TCP Error Control	UDP Client Server Program	DHCP Operation	Comparison between IPV4 and IPV6 Header	PPP Services, Components

Duration (hour)		15	15	15	15	15
	SLO-2	TCP Congestion Control	UDP Control block table & Module	DHCP Configuration	IPV4 to IPV6 Tunneling	PPP frame and byte stuffing
S-12	SLO-1	TCP Flow Control	UDP Input & Output Module	SMTP	IPV4 to IPV6 Translation Techniques	HDLC
	SLO-2	Multicasting	SCTP Sockets	POP3	NAT Protocol Translation	HDLC Transfer Modes, Frame
S-13	SLO-1	Multicasting and Multicast Routing Protocol	SCTP Services and Features, Packet Format	IMAP	IPV6 Mobility	Types of HDLC Frame
	SLO-2	Stream Control Transmission Protocol	SCTP Client/Server	MIME	Protocols Changed to Support IPV6	MPLS
S 14-15	SLO-1 SLO-2	Simple TCP/IP Client Server Communication	Half Duplex Chat Using TCP/IP	Remote Command Execution Using UDP	Implementation of NAT	Communication Using PPP

Learning Resources	1. Behrouz A. Forouzan, "TCP IP Protocol Suite " 4th edition, 2010, McGraw-HillISBN: 0073376043 2. Douglas E. Comer, Internetworking with TCP/IP, Principles, protocols, and architecture,Vol 1 5th Edition,2006 ISBN: 0131876716, ISBN: 978-0131876712	3. Richard Stevens, Unix Network Programming, vol.1, 3rd edition, 2003, McGraw-HillISBN 0-07-246060-1

Learning Assessment											
	Bloom's Level of Thinking	Continuous Learning Assessment (50% weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember Understand	20 %	20%	15 %	15%	15 %	15%	15 %	15%	15 %	15%
Level 2	Apply Analyze	20 %	20 %	20 %	20%	20 %	20%	20 %	20%	20 %	20%
Level 3	Evaluate Create	10 %	10%	15 %	15%	15 %	15%	15 %	15%	15 %	15%
	Total	100 %		100 %		100 %		100 %		100 %	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

Course Designers		
Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
1. Thamaraiselvam.S, Zoho Corporation. thamaraiselvams@gmail.com	1. Dr.Uma,Anna University ,umaramesh@auist.net	1. Dr.K.Venkatesh,SRMIST,2.Dr.G.Usha,SRMIST
2. Mithun,Cognizant, Mithun.SS@cognizant.com	2. Dr.KunvarSingh, NIT Trichy,kunwar@nitt.edu	2. Dr.J.Kalaivani,SRMIST,4.Mr.GodwinPon,SRMIST

Course Code	18CSC303J	Course Name	DATABASE MANAGEMENT SYSTEMS	Course Category	C	Professional Core	L	T	P	C
							3	0	2	4

Pre-requisite Courses	*Nil*	Co-requisite Courses	*Nil*	Progressive Courses	*Nil*
Course Offering Department	*Computer Science and Engineering*	Data Book / Codes/Standards		*Nil*	

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*
CLR-1 :	*Understand the fundamentals of Database Management Systems, Architecture and Languages*
CLR-2 :	*Conceive the database design process through ER Model and Relational Model*
CLR-3 :	*Design Logical Database Schema and mapping it to implementation level schema through Database Language Features*
CLR-4 :	*Familiarize queries using Structure Query Language (SQL) and PL/SQL*
CLR-5 :	*Familiarize the Improvement of the database design using normalization criteria and optimize queries*
CLR-6 :	*Understand the practical problems of concurrency control and gain knowledge about failures and recovery*

		Learning			Program Learning Outcomes (PLO)														
		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Course Learning Outcomes (CLO):	*At the end of this course, learners will be able to:*	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
CLO-1 :	*Acquire the knowledge on DBMS Architecture and Languages*	*3*	*80*	*70*	*H*	*M*	*L*	*L*	-	-	-	-	*L*	*L*	*L*	*H*	-	-	-
CLO-2 :	*Apply the fundamentals of data models to model an application's data requirements using conceptual modeling tools like ER diagrams*	*3*	*85*	*75*	*H*	*H*	*H*	*H*	*H*	-	-	-	*H*	*H*	*H*	*H*	-	-	-
CLO-3 :	*Apply the method to convert the ER model to a database schemas based on the conceptual relational model*	*3*	*75*	*70*	*H*	*H*	*H*	*H*	*H*	-	-	-	*H*	*H*	*H*	*H*	-	-	-
CLO-4 :	*Apply the knowledge to create, store and retrieve data using Structure Query Language (SQL) and PL/SQL*	*3*	*85*	*80*	*H*	*H*	*H*	*H*	*H*	-	-	-	*H*	*H*	*H*	*H*	-	-	-
CLO-5 :	*Apply the knowledge to improve database design using various normalization criteria and optimize queries*	*3*	*85*	*75*	*H*	*H*	*L*	*M*	*L*	-	-	-	*M*	*M*	*M*	*L*	-	-	-
CLO-6 :	*Appreciate the fundamental concepts of transaction processing- concurrency control techniques and recovery procedures.*	*3*	*85*	*75*	*H*	*L*	*L*	*L*	*L*	-	-	-	*H*	*L*	*L*	*L*			

Duration (hour)		15	15	15	15	15
S-1	SLO-1	*What is Database Management System*	*Database Design*	*Basics of SQL-DDL,DML,DCL,TCL*	*Relational Algebra – Fundamental Operators and syntax, relational algebra queries, Tuple relational calculus*	*Transaction concepts, properties of transactions,*
	SLO-2	*Advantage of DBMS over File Processing System*	*Design process*	*Structure Creation, alternation*		
S-2	SLO-1	*Introduction and applications of DBMS*	*Entity Relation Model*	*Defining Constraints-Primary Key, Foreign Key, Unique, not null, check, IN operator*		*Serial izability of transactions, testing for serial inability, System recovery,*
	SLO-2	*Purpose of database system*				
S-3	SLO-1	*Views of data*	*ER diagram*	*Functions-aggregation functions Built-in Functions-numeric, date, string functions, string functions, Set operations,*	*Pitfalls in Relational database, Decomposing bad schema Functional Dependency – definition, trivial and non-trivial FD*	*Concurrency Control*
	SLO-2					
S 4-5	SLO-1	*Lab 1: SQL Data Definition Language Commands on sample exercise * The abstract of the project to construct database must be framed*	*Lab4 Inbuilt functions in SQL on sample Exercise.*	*Lab 7 : Join Queries on sample exercise. * Frame and execute the appropriate DDL,DML,DCL,TCL for the project*	*Lab10: PL/SQL Procedures on sample exercise. * Frame and execute the appropriate Join Queries for the project*	*Lab 13: PL/SQL Exception Handling * Frame and execute the appropriate PL/SQL Procedures and Functions for the project*
	SLO-2					
S-6	SLO-1	*Database system Architecture*	*Keys , Attributes and Constraints*	*Sub Queries, correlated sub queries*	*closure of FD set , closure of attributes irreducible set of FD*	*Two- Phase Commit protocol, Recovery and Atomicity*
	SLO-2					
S-7	SLO-1	*Data Independence*	*Mapping Cardinality*	*Nested Queries, Views and its Types*	*Normalization – 1Nf, 2NF, 3NF,*	*Log-based recovery*
	SLO-2					

Duration (hour)		15	15	15	15	15
S-8	SLO-1	The evolution of Data Models	Extended ER - Generalization, Specialization and Aggregation	Transaction Control Commands Commit, Rollback, Save point	Decomposition using FD- dependency preservation,	concurrent executions of transactions and related problems
	SLO-2					
S 9-10	SLO-1	Lab 2: SQL Data Manipulation Language Commands * Identification of project Modules and functionality	Lab 5: Construct a ER Model for the application to be constructed to a Database	Lab 8: Set Operators & Views. * Frame and execute the appropriate In- Built functions for the project	Lab 11: PL/SQL Functions * Frame and execute the appropriate Set Operators & Views for the project	Lab 14: PL/SQL Trigger * Frame and execute the appropriate PL/SQL Cursors and Exceptional Handling for the project
	SLO-2					
S-11	SLO-1	Degrees of Data Abstraction	ER Diagram Issues Weak Entity	PL/SQL Concepts- Cursors	BCNF	Locking mechanism, solution to concurrency related problems
	SLO-2					
S-12	SLO-1	Database Users and DBA	Relational Model	Stored Procedure, Functions Triggers and Exceptional Handling	Multi- valued dependency, 4NF	Deadlock
	SLO-2					
S-13	SLO-1	Database Languages	Conversion of ER to Relational Table	Query Processing	Join dependency and 5NF	two-phase locking protocol, Isolation, Intent locking
	SLO-2					
S 14-15	SLO-1	Lab 3: SQL Data Control Language Commands and Transaction control commands to the sample exercises * Identify the issues that can arise in a business perspective for the application	Lab 6: Nested Queries on sample exercise * Construction of Relational Table from the ER Diagram	Lab9: PL/SQL Conditional and Iterative Statements * Frame and execute the appropriate Nested Queries for the project	Lab 12: PL/SQL Cursors * Frame and execute the appropriate PL/SQL Conditional and Iterative Statements for the project	Lab 15 : * Frame and execute the appropriate PL/SQL Cursors and Exceptional Handling for the project * Demo of the project
	SLO-2					

Learning Resources	1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, Database System Conceptsǁ, Sixth Edition, Tata McGraw Hill,2011. 2. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systemsǁ, Sixth Edition, Pearson Education,2011. 3. CJ Date,A Kannan,S Swamynathan, An Introduction to Database Systems, Eight Edition, Pearson Education,2006. 4. Rajesh Narang, Database Management Systems, 2nd ed., PHI Learning Private Limited,2011.	4. Martin Gruber, Understanding SQL, Sybex,1990 5. Sharad Maheshwari,Introduction to SQLandPL/SQL,2ded.,LaxmiPublications,2016. 6. RaghuramaKrishnan,Johannes Gehrke,Database Management Systems,3rdEdition,McGrawHill Education,2003.

Learning Assessment											
	Bloom's Level of Thinking	Continuous Learning Assessment (50%weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	20%	20%	15%	15%	15%	15%	15%	15%	15%	15%
	Understand										
Level 2	Apply	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%
	Analyze										
Level 3	Evaluate	10%	10%	15%	15%	15%	15%	15%	15%	15%	15%
	Create										
	Total	100 %		100 %		100 %		100 %		-	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, Conf. Paper etc.,

Course Designers		
Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
1. Dr.Mariappan Vaithilingam, Engineering Leader Amazon, dr.v.m@ieee.org		1. Ms. Sasi Rekha Sankar SRMIST
2. Mr. Badinath, SDET, Amzon, sbadhrinath@gmail.com		2. Mr.Elizer, SRMIST
		3. Mrs. Hemavathy, SRMIST

Course Code	18CSC304J	Course Name	COMPILER DESIGN	Course Category	C	Professional Core	L	T	P	C
							3	0	2	4

Pre-requisite Courses	*18CSC301T*	Co-requisite Courses	*Nil*	Progressive Courses	*Nil*
Course Offering Department	*Computer Science and Engineering*	Data Book / Codes/Standards	*Nil*		

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*
CLR-1 :	*Utilize the mathematics and engineering principles for the Design of Compilers*
CLR-2 :	*Acquire knowledge of Lexical Analyzer from a specification of a language's lexical rules*
CLR-3 :	*Acquire knowledge of Syntax Analyzer for parsing the sentences in a compiler grammar*
CLR-4 :	*Gain knowledge to translate a system into various intermediate codes*
CLR-5 :	*Analyze the methods of implementing a Code Generator for compilers*
CLR-6 :	*Analyze and Design the methods of developing a Code Optimizer*

		Learning			Program Learning Outcomes (PLO)														
		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Course Learning Outcomes (CLO):	*At the end of this course, learners will be able to:*	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
CLR-1 :	*Utilize the mathematics and engineering principles for the Design of Compilers*	*3*	*80*	*70*	*H*	*H*	*H*	*H*	*M*	*L*	*L*	*L*	*M*	*M*	*L*	*H*	*H*	*H*	*H*
CLR-2 :	*Acquire knowledge of Lexical Analyzer from a specification of a language's lexical rules*	*3*	*85*	*75*	*H*	*H*	*H*	*H*	*M*	*L*	*L*	*L*	*M*	*M*	*L*	*H*	*H*	*H*	*H*
CLR-3 :	*Acquire knowledge of Syntax Analyzer for parsing the sentences in a compiler grammar*	*3*	*75*	*70*	*H*	*H*	*H*	*H*	*M*	*L*	*L*	*L*	*M*	*M*	*L*	*H*	*H*	*H*	*H*
CLR-4 :	*Gain knowledge to translate a system into various intermediate codes*	*3*	*85*	*80*	*H*	*H*	*H*	*H*	*M*	*L*	*L*	*L*	*M*	*M*	*L*	*H*	*H*	*H*	*H*
CLR-5 :	*Analyze the methods of implementing a Code Generator for compilers*	*3*	*85*	*75*	*H*	*H*	*H*	*H*	*M*	*L*	*L*	*L*	*M*	*M*	*L*	*H*	*H*	*H*	*H*
CLR-6 :	*Analyze and Design the methods of developing a Code Optimizer*	*3*	*80*	*70*	*H*	*H*	*H*	*H*	*M*	*L*	*L*	*L*	*M*	*M*	*L*	*H*	*H*	*H*	*H*

Duration (hour)		15	15	15	15	15
S-1	SLO-1	*Compilers – Analysis of the source program*	*Syntax Analysis Definition - Role of parser*	*Bottom Up Parsing*	*Intermediate Code Generation*	*Code optimization*
	SLO-2	*Phases of a compiler – Cousins of the Compiler*	*Lexical versus Syntactic Analysis*	*Reductions*	*Intermediate Languages - prefix - postfix*	*Introduction– Principal Sources of Optimization*
S-2	SLO-1	*Grouping of Phases – Compiler construction tools*	*Representative Grammars*	*Handle Pruning*	*Quadruple - triple - indirect triples Representation*	*Function Preserving Transformation*
	SLO-2	*Lexical Analysis – Role of Lexical Analyzer*	*Syntax Error Handling*	*Shift Reduce Parsing*	*Syntax tree- Evaluation of expression - three-address code*	*Loop Optimization*
S-3	SLO-1	*Input Buffering*	*Elimination of Ambiguity, Left Recursion*	*Problems related to Shift Reduce Parsing*	*Synthesized attributes – Inherited attributes*	*Optimization of basic Blocks*
	SLO-2	*Specification of Tokens*	*Left Factoring*	*Conflicts During Shift Reduce Parsing*	*Intermediate languages – Declarations*	*Building Expression of DAG*
S 4-5	SLO-1 SLO-2	*Lab 1 - Implementation of Lexical Analyzer*	*Lab 4Elimation of Ambiguity, Left Recursion and Left Factoring*	*Lab 7 - Shift Reduce Parsing*	*Lab 10-Intermediate code generation – Postfix, Prefix*	*Lab 13 Implementation of DAG*
S-6	SLO-1	*Finite automation - deterministic*	*Top down parsing*	*LR Parsers- Why LR Parsers*	*Assignment Statements*	*Peephole Optimization*
	SLO-2	*Finite automation - non deterministic*	*Recursive Descent Parsing, back tracking*	*Items and LR(0) Automaton, Closure of Item Sets,*	*Boolean Expressions, Case Statements*	*Basic Blocks, Flow Graphs*
S-7	SLO-1	*Transition Tables*	*Computation of FIRST*	*LR Parsing Algorithm*	*Back patching – Procedure calls*	*Next -Use Information*
	SLO-2	*Acceptance of Input Strings by Automata*	*Problems related to FIRST*	*Operator Precedence Parser Computation of LEADING*	*Code Generation*	*Introduction to Global Data Flow Analysis*

Duration (hour)		15	15	15	15	15
S-8	SLO-1	State Diagrams and Regular Expressions	Computation of FOLLOW	Computation of TRAILING	Issues in the design of code generator	Computation of gen and kill
	SLO-2	Conversion of regular expression to NFA – Thompson's	Problems related to FOLLOW	Problems related to LEADING AND TRAILING	The target machine – Runtime Storage management	Computation of in and out
S 9-10	SLO-1	Lab 2 conversion from Regular Expression to NFA	Lab 5 -FIRST AND FOLLOW computation	Lab 8- Computation of LEADING AND TRAILING	Lab 11 Intermediate code generation – Quadruple, Triple, Indirect triple	Lab 14 : Implementation of Global Data Flow Analysis
	SLO-2					
S-11	SLO-1	Conversion of NFA to DFA	Construction of a predictive parsing table	SLR Grammars	A simple Code generator	Parameter Passing.
	SLO-2	Simulation of an NFA	Predictive Parsers LL(1) Grammars	SLR Parsing Tables	Code Generation Algorithm	Runtime Environments
S-12	SLO-1	Converting Regular expression directly to DFA	Transition Diagrams for Predictive Parsers	Problems related to SLR	Register and Address Descriptors	Source Language issues
	SLO-2	Minimization of DFA	Error Recovery in Predictive Parsing	Construction of Canonical LR(1) and LALR	Generating Code of Assignment Statements	Storage Organization
S-13	SLO-1	Minimization of NFA	Predictive Parsing Algorithm	Construction of LALR	Cross Compiler – T diagrams	Activation Records
	SLO-2	Design of lexical analysis (LEX)	Non Recursive Predictive Parser	Problems related to Canonical LR(1) and LALR Parsing Table	Issues in Cross compilers	Storage Allocation strategies
S 14-15	SLO-1	Lab 3 Conversion from NFA to DFA	Lab 6 Predictive Parsing Table	Lab9 Computation of LR(0) items	Lab 12 : A simple code Generator	Lab 15: Implement any one storage allocation strategies(heap, stack, static)
	SLO-2					

Learning Resources	1. AlfredVAho,JefferyDUllman,RaviSethi,"Compilers,Principlestechniquesandtools",Pearson Education2011 2. S.GodfreyWinster,S.ArunaDevi,R.Sujatha,"CompilerDesign",YesdeePublishingPvt.Ltd,2016 3. WilliamM.WaiteandGerhardGoos.CompilerConstruction.Springer-Verlag,New York,2013.	4. K.Muneeswaran,,"CompilerDesign",OxfordHigherEducation,Fourthedition2015 5. DavidGalles,"ModernCompilerDesign",PearsonEducation,Reprint2012. 6. RaghavanV.,"PrinciplesofCompilerDesign",TataMcGrawHillEducationPvt.Ltd.,2010

Learning Assessment											
	Bloom's Level of Thinking	Continuous Learning Assessment (50%weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	20%	20%	15%	15%	15%	15%	15%	15%	15%	15%
	Understand										
Level 2	Apply	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%
	Analyze										
Level 3	Evaluate	10%	10%	15%	15%	15%	15%	15%	15%	15%	15%
	Create										
	Total	100 %		100 %		100 %		100 %		-	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

Course Designers		
Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
		1. Ms.R.Jeya
		2. Mrs.J. Jeyasudha

Course Code	18CSC305J	Course Name	ARTIFICIAL INTELLIGENCE	Course Category	C	Professional Core	L	T	P	C
							3	0	2	4

Pre-requisite Courses	*Nil*	Co-requisite Courses	*Nil*	Progressive Courses	*Nil*
Course Offering Department	*Computer Science and Engineering*	Data Book / Codes/Standards	*Nil*		

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*
CLR-1 :	*Provide a broad understanding of the basic techniques for building intelligent computer systems and an understanding of how AI is applied to problems.*
CLR-2 :	*Gain knowledge in problem formulation and building intelligent agents*
CLR-3 :	*Understand the search technique procedures applied to real world problems*
CLR-4 :	*Understand the types of logic and knowledge representation schemes*
CLR-5 :	*Acquire knowledge in planning and learning algorithms*
CLR-6 :	*Gain knowledge in AI Applications and advances in Artificial Intelligence*

Course Learning Outcomes (CLO):	*At the end of this course, learners will be able to:*	Learning			Program Learning Outcomes (PLO)														
		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO – 3
CLO-1 :	*Formulate a problem and build intelligent agents*	*1*	*80*	*70*	*M*	*M*	*M*	*M*	*H*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*L*	*L*	*L*
CLO-2 :	*Apply appropriate searching techniques to solve a real world problem*	*2*	*85*	*75*	*M*	*H*	*H*	*H*	*H*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*M*	*L*	*M-*
CLO-3 :	*Analyze the problem and infer new knowledge using suitable knowledge representation schemes*	*2*	*75*	*70*	*M*	*H*	*H*	*M*	*H*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*M*	*L*	*M*
CLO-4 :	*Develop planning and apply learning algorithms on real world problems*	*2*	*85*	*80*	*M*	*H*	*M*	*H*	*H*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*M*	*M*	*M*
CLO-5 :	*Design an expert system and implement natural language processing techniques*	*3*	*85*	*75*	*M*	*H*	*H*	*H*	*H*	*-*	*-*	*-*	*M*	*L*	*-*	*H*	*H*	*M*	*H*
CLO-6 :	*Implement advance techniques in Artificial Intelligence*	*3*	*80*	*70*	*L*	*H*	*M*	*M*	*H*	*-*	*-*	*-*	*H*	*L*	*-*	*H*	*H*	*M*	*H*

Duration (hour)		15	15	15	15	15
S-1	SLO-1	*Introduction to AI-AI techniques*	*Searching techniques- Uniformed search-General search Algorithm*	*Knowledge and reasoning-Approaches and issues of knowledge reasoning*	*Planning- Planning problems, Simple planning agent*	*Expert system-Architecture*
	SLO-2	*Problem solving with AI*	*Uniformed search Methods-Breadth first search*	*Knowledge base agents-Logic Basics*	*Planning languages*	*Pros and Cons of expert system*
S-2	SLO-1	*AI Models, Data acquisition and learning aspects in AI*	*Uniformed search Methods-Depth first search*	*Logic-Propositional logic-syntax ,semantics and inferences*	*Blocks world ,Goal stack planning*	*Rule based systems*
	SLO-2	*Problem solving- Problem solving process, Formulating problems*	*Uniformed search Methods-Depth limited search*	*Propositional logic- Reasoning patterns*	*Mean Ends Analysis*	*Frame based expert system*
S-3	SLO-1	*Problem types and characteristics*	*Uniformed search Methods- Iterative Deepening search*	*Predicate logic – Syntax and semantics, instance and is relationship*	*Non-linear Planning*	*Case study*
	SLO-2	*Problem space and search*	*Bi-directional search*	*Unification and Resolution*	*Conditional planning, Reactive planning*	*Case study*
S 4-5	SLO-1 SLO-2	*Lab 1: Implementation of toy problems*	*Lab4: Implementation and Analysis of DFS and BFS for an application*	*Lab 7: Implementation of unification and resolution for real world problems.*	*Lab 10 :Implementation of block world problem*	*Natural language processing-Levels of NLP*
S-6	SLO-1	*Intelligent agent*	*Informed search- Generate and test, Best First search*	*Knowledge representation using rules*	*Learning- Machine learning*	*Syntactic and Semantic Analysis*
	SLO-2	*Rationality and Rational agent with performance measures*	*Informed search-A* Algorithm*	*Knowledge representation using semantic nets*	*Goals and Challenges of machine learning*	*Information retrieval*
S-7	SLO-1	*Flexibility and Intelligent agents*	*AO* research*	*Knowledge representation using frames*	*Learning concepts, models*	*Information Extraction*

Duration (hour)		15	15	15	15	15
	SLO-2	*Task environment and its properties*	*Local search Algorithms-Hill Climbing, Simulated Annealing*	*Inferences*	*Artificial neural network based learning-Back propagation*	*Machine translation*
S-8	SLO-1	*Types of agents*	*Local Beam Search*	*Uncertain Knowledge and reasoning-Methods*	*Support vector machines*	*NLP Applications*
	SLO-2	*Other aspects of agents*	*Genetic Algorithms*	*Bayesian probability and belief network*	*Reinforcement learning*	*NLP Applications*
S 9-10	SLO-1 SLO-2	*Lab 2: Developing agent programs for real world problems*	*Lab 5: Developing Best first search and A* Algorithm for real world problems*	*Lab 8: Implementation of knowledge representation schemes - use cases*	*Lab 11: Implementation of learning algorithms for an application*	*Lab 14:Implementation of NLP programs*
S-11	SLO-1	*Constraint satisfaction problems(CSP)*	*Adversarial search Methods-Game playing-Important concepts*	*Probabilistic reasoning*	*Adaptive learning*	*Advance topics in Artificial Intelligence-Cloud Computing and intelligent agent*
	SLO-2	*Crypto arithmetic puzzles*	*Game playing and knowledge structure*	*Probabilistic reasoning over time*	*Multi agent based learning*	*Business intelligence and analytics*
S-12	SLO-1	*CSP as a search problem-constrains and representation*	*Game as a search problem-Mini max approach*	*Forward and backward reasoning*	*Ensemble learning*	*Sentiment Analysis*
	SLO-2	*CSP-Backtracking, Role of heuristic*	*Mini max Algorithm*	*Other uncertain techniques-Data mining*	*Learning for decision making*	*Deep learning Algorithms*
S-13	SLO-1	*CSP-Forward checking and constraint propagation*	*Alpha beta pruning*	*Fuzzy logic*	*Distributed learning*	*Deep learning Algorithms*
	SLO-2	*CSP-Intelligent backtracking*	*Game theory problems*	*Dempster -shafer theory*	*Speedup learning*	*Planning and logic in intelligent agents*
S 14-15	SLO-1 SLO-2	*Lab 3: Implementation of constraint satisfaction problems*	*Lab 6: Implementation of mini max algorithm for an application*	*Lab 9: Implementation of uncertain methods for an application*	*Lab12: Development of ensemble model for an application*	*Lab 15: Applying deep learning methods to solve an application.*

Learning Resources	1. *Parag Kulkarni, Prachi Joshi, Artificial Intelligence –Building Intelliegent Systems, 1st ed., PHI learning,2015* 2. *DeepakKemhani,FirstcourseinArtificilaIntelligence,McGrawHillPvtLtd,2013* 3. *Stuart J. Russell, Peter Norwig , Artificial Intelligence –A Modern approach, 3rd Pearson Education, 2016*	4. *PrateekJoshi,ArtificialIntelligencewithPhython,1sted.,PacktPublishing,2017* 5. *DenisRothman,ArtificialIntelligencebyExample,Packt,2018*

Learning Assessment											
	Bloom's Level of Thinking	Continuous Learning Assessment (50% weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember Understand	*20%*	*20%*	*10%*	*10%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*
Level 2	Apply Analyze	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*	*20%*
Level 3	Evaluate Create	*10%*	*10%*	*20%*	*20%*	*15%*	*15%*	*15%*	*15%*	*15%*	*15%*
	Total	*100 %*		*100 %*		*100 %*		*100 %*		-	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

Course Designers		
Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
1. Mr.Jagatheeswaran, Lead, Auxo labs jagatheeswarans.iot@auxolabs.in	*1. Dr. Chitrakala, Anna University, au.chitras@gmail.com*	*1. Dr.M.Pushpalatha, SRMIST*
		2. Dr.G..Vadivu, SRMIST
		3. Dr.C.Lakshmi, SRMIST

Course Code	18CSC350T	Course Name	COMPREHENSION	Course Category	C	Professional Core	L	T	P	C
							0	1	0	1

Pre-requisite Courses	*Nil*	Co-requisite Courses	*Nil*	Progressive Courses	*Nil*
Course Offering Department	*Computer Science and Engineering*	Data Book / Codes/Standards		*Nil*	

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*	Learning			Program Learning Outcomes (PLO)														
CLR-1 :	*Acquire skills to solve real world problems in Data Structures and Analysis and Design of Algorithms*	1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CLR-2 :	*Acquire skills to solve real world problems in Object Oriented Design and Programming and advanced programming concepts*	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design,	Modern Tool Usage	Society & Culture	Environment &	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO - 3
CLR-3 :	*Acquire skills to solve real world problems in Operating systems , Computer networking and Formal Language and Automata*																		
CLR-4 :	*Acquire skills to solve real world problems in Compiler Design, Database Management systems and Software Engineering*																		
CLR-5 :	*Acquire skills to solve real world problems for competitive examinations in Mechanical Engineering*																		
CLR-6 :	*Acquire skills to solve real world problems in the broad domain of Mechanical Engineering*																		
Course Learning Outcomes (CLO):	*At the end of this course, learners will be able to:*																		
CLO-1 :	*Practice and gain confidence, competence to solve problems in Data Structures and Analysis and Design of Algorithms*	*3*	*85*	*80*	*H*	*H*	*H*	*H*	*M*	*M*	*L*	*L*	*M*	*M*	*L*	*H*	*M*	*L*	*H*
CLO-2 :	*Practice and gain confidence, competence to solve problems in Object Oriented Design, Programming and advanced programming concepts*	*3*	*85*	*80*	*H*	*H*	*M*	*H*	*H*	*M*	*L*	*L*	*H*	*H*	*M*	*H*	*H*	*H*	*H*
CLO-3 :	*Practice and gain confidence, competence to solve problems in Operating systems , Computer networking, Formal Language and Automata*	*3*	*85*	*80*	*H*	*H*	*M*	*M*	*M*	*M*	*L*	*L*	*M*	*H*	*L*	*M*	*M*	*H*	*M*
CLO-4 :	*Practice and gain confidence, competence to solve problems in Compiler Design, Database Management systems and Software Engineering*	*3*	*85*	*80*	*H*	*H*	*M*	*H*	*H*	*H*	*L*	*L*	*H*	*H*	*H*	*H*	*M*	*H*	*H*
CLO-5 :	*Practice and gain confidence and competence to solve problems for competitive examinations in Computer Science and Engineering*	*3*	*85*	*80*	*H*	*H*	*H*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*M*	*L*	*M*
CLO-6 :	*Practice and gain confidence and competence to solve problems in the broad domain of Computer Science and Engineering*	*3*	*85*	*80*	*H*	*H*	*M*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*M*	*M*	*M*

Duration (hour)		3	3	3	3	3
S-1	SLO-1	*Tutorial on Linear Data Structures*	*Tutorial on Object Oriented Design*	*Tutorial on Operating Systems*	*Tutorial on Compiler Design*	*Problem Solving*
	SLO-2	*Problem Solving*	*Problem Solving*	*Problem Solving*	*Problem Solving*	*Problem Solving*
S-2	SLO-1	*Tutorial on Non Linear Data Structures*	*Tutorial on Object Oriented Programming*	*Tutorial on Computer networking*	*Tutorial on Database Management systems*	*Problem Solving*
	SLO-2	*Problem Solving*	*Problem Solving*	*Problem Solving*	*Problem Solving*	*Problem Solving*
S-3	SLO-1	*Tutorial on Analysis and Design of Algorithms*	*Tutorial on Advanced Programming concepts*	*Tutorial on Formal Language and Automata*	*Tutorial on Software Engineering*	*Problem Solving*
	SLO-2	*Problem Solving*	*Problem Solving*	*Problem Solving*	*Problem Solving*	*Problem Solving*

Learning Resources		
	1. Jushta Jaiswal,Objective “Computer Science & Information Technology”, Source books, , 2015 *2. G.K.Mithal, “Objective Computer Science and Information Technology”,G.K.Publishing, 10th edition, 2016*	*3. R.Agor, “Computer Science Conventional & Objective type solved questions”, Birla Publishing, 2004* *4. Timothy Williams, “MCQs in Computer Science”, McGraw Hill, 5th edition, 2017* *5. Surbhi Mitra, “Computer Science and IT”,Arihant Handbook series, 2013*

Learning Assessment

	Bloom's Level of Thinking	Continuous Learning Assessment (100% weightage)								Final Examination	
		CLA – 1 (20%)		CLA – 2 (30%)		CLA – 3 (30%)		CLA – 4 (20%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	*40%*	-	*30%*	-	*30%*	-	*30%*	-	-	-
	Understand										
Level 2	Apply	*40%*	-	*40%*	-	*40%*	-	*40%*	-	-	-
	Analyze										
Level 3	Evaluate	*20%*	-	*30%*	-	*30%*	-	*30%*	-	-	-
	Create										
	Total	100 %		100 %		100 %		100 %		-	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

Course Designers

Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
1. Dr. Anbu Rathinavel , Chief Design Officer, Design Intellect	*1. Dr. Viraj Kumar, Professor, CSE, PES University*	*1. Dr. B.Amutha, Professor & Head, CSE, SRMIST*
		2. Dr.S.S.Sridhar, Professor,CSE, SRMIST