



MINOR DEGREE STRUCTURE

Offered by Department of Computer Science and Engineering Minor in Computer Science and Engineering

Computer Science Engineering (CSE) transforms our day-to-day lives through a multitude of innovative technologies and products. The Computer Science Engineering (CSE) minor is intended to expose students from other disciplines to the unlimited opportunities for innovation in this exciting field, and to the methodologies and tools used by Computer Science engineers for the exploration and design of new technologies and products. Undergraduate students of the university who are not majoring in CSE have the option to take a Minor in Computer Science Engineering. The program is expected to accommodate students of diverse backgrounds.

- An **academic major** is the academic discipline to which an undergraduate student formally commits. A student who successfully completes all courses required for the major qualifies for an undergraduate degree.
- **Academic minor** is an academic discipline outside of the student's academic major in which he or she takes a small number of classes.
- An academic major or major field refers to a student's primary focus within their degree program while a minor or minor field refers to his or her secondary focus.
- **Minors are optional.** You may complete multiple minors or none at all.
- Some students will prepare for their intended career with their major, while pursuing personal interests with a minor, for example, majoring in Computer Science while minoring in a foreign language or performing arts or Economics or Management Studies. Other students may pursue a minor to provide specific specialization and thus make themselves more attractive to employers.

Why Choose a Minor?

- Adding depth to your studies
- Adding breadth to your studies
- An area of passion
- An alternative to double majoring

Objectives of this minor program

1. To impart to students basic fundamental and applied knowledge in the Computer Science discipline.
2. To convey an understanding of the continuously increasing contributions of Computer Science to society.
3. To obtain a working knowledge of core Computer Science principles.
4. To understand the current state-of-the-art within the Computer Science discipline, and to bridge between their major discipline of study and that of Computer Science.

Minors: Rules and Regulations

1. Pursuing a major/minor program is a highly individual decision, and should be based on a student's educational and career goals.
2. Minors must be completed simultaneously with a major degree program. You cannot earn a minor after you have already earned your bachelor's degree(s).
3. You need at least one active major in order to pursue a minor.
4. Also be aware that since the minor must be completed with a major, any outstanding minor requirements will prevent the awarding of the degree for your major. If you ultimately decide to graduate without the minor, the minor must be removed from your records before your degree can be processed.
5. The Certification for Minor will NOT be issued until the Major degree is successfully completed, even though all requirements for the Minor have been satisfied.
6. All requirements for the Minor must be completed within a maximum of ONE semester of the completion of the Major degree but NOT LATER THAN the maximum duration of study of the Major degree permitted by UGC i.e., not more than N+2 years where 'N' is the normal duration of study as per regulations.
7. If a student drops from a Minor or is NOT able to fulfil all the requirements for the certification of Minor, within the maximum period of study permitted by UGC, the student will NOT be issued the Certification. Nevertheless, the transcripts for the completed courses will be issued.
8. Courses offered for a Certification in Minor will be treated on par with the regular courses of a Major in respect of attendance requirements, assessment and examination requirements.
9. The number of seats for minor in Computer Science is limited and subject to availability and academic performance.
10. Register for a Minor any time after 3rd Semester of B.Tech
11. The offering Department scrutinises the credentials of registered students and announces the list of selected students who fulfil the minimum eligibility criteria for admission into a Minor Certificate programme.
12. The selected students enrol into the Minor Certificate programme by paying a one-time programme fee of Rs. 75,000/- that includes the tuition fee, examination fee and fee for transcript and certificate.
13. The student is permitted to register for a maximum of 2 courses per semester, over and above the maximum credits permitted (26 credits) for a Major degree of study.
14. A minor requires 18-20 credits to be acquired from the courses offered by the Dept. of Computer Science.

Eligibility Criteria

A student will be permitted to pursue a Minor, if and only if, he / she fulfils the following criteria:

- i. Must have secured a minimum of 7.0 CGPA* at the time of admission to a Minor
- ii. There must NOT be any standing arrears / Backlogs.
- iii. Must be in the active rolls of the department without any break of study or disciplinary action pending against the student
- iv. Must have NO outstanding fee dues

* CGPA cut off is subject to change based on the demand and competition.

MINOR IN COMPUTER SCIENCE ENGINEERING

Category	Course Code	Course Title	L	T	P	C	Pre-Requisite	Credits to be earned
Professional Core	18CSC001J	Data Structures and Algorithms	3	0	2	4		11
	18CSC002J	Object Oriented Design and Programming	3	0	2	4		
	18CSC003J	Computer Organization and Architecture	2	0	2	3		
Professional Electives (To choose any Three)	18CSE001J	Operating Systems	2	0	2	3		09
	18CSE002T	Software Engineering and Project Management	2	1	0	3		
	18CSE003J	Python Programming	2	0	2	3		
	18CSE004T	Formal Language and Automata	3	0	0	3		
	18CSE005J	Computer Networks	2	0	2	3		
	18CSE006J	Database Management Systems	2	0	2	3		
	18CSE007J	Compiler Design	2	0	2	3		
	18CSE008J	Artificial Intelligence	2	0	2	3		
	18CSE009T	Network Security	3	0	0	3		
	18CSE010T	Data Mining and Analytics	3	0	0	3		
Total Credits								20

Course Code	18CSC001J	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	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 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		
1	2	3
Identify linear and non-linear data structures.	Create algorithms for searching and sorting	Construct stack and queue data structures and evaluate its operations
Construct stack and queue data structures and evaluate its operations	Construct tree data structures and evaluate its types and operations	Construct graph data structure, evaluate its operations, implement algorithms to identify shortest path

Program Learning Outcomes (PLO)														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	H	-	H	L	-	-	-	L	L	-	H	-	-	-
M	H	L	M	L	-	-	-	M	L	-	H	-	-	-
M	H	M	H	L	-	-	-	M	L	-	H	-	-	-
M	H	M	H	L	-	-	-	M	L	-	H	-	-	-
H	H	M	H	L	-	-	-	M	L	-	H	-	-	-

Course Learning Outcomes (CLO):	At the end of this course, learners will be able to:
CLO-1:	Identify linear and non-linear data structures. Create algorithms for searching and sorting
CLO-2:	Create the different types of linked lists and evaluate its operations
CLO-3:	Construct stack and queue data structures and evaluate its operations
CLO-4:	Create tree data structures and evaluate its types and operations
CLO-5:	Create graph data structure, evaluate its operations, implement algorithms to identify shortest path

CLO-6 :	Construct the different data structures and evaluate their types and operations	3	8 0	7 0	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	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
	SLO-2					
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	AVLTrees: 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-	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

15	SLO-2				
----	-------	--	--	--	--

Learning Resources	1. Seymour Lipschutz, <i>Data Structures with C</i> , McGraw Hill, 2014	5. Reema Thareja, <i>Data Structures Using C</i> , 1 st ed., Oxford Higher Education, 2011
	2. R.F.Gilberg, B.A.Forouzan, <i>Data Structures</i> , 2 nd ed., Thomson India, 2005	
	3. A.V.Aho, J.E Hopcroft, J.D.Ullman, <i>Data structures and Algorithms</i> , Pearson Education, 2003	
	4. Mark Allen Weiss, <i>Data Structures and Algorithm Analysis in C</i> , 2 nd ed., Pearson Education, 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 %		-	

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, nrababu@iitm.ac.in	2. Dr.Subalalitha C.N, SRMIST
	3. Dr.Noor Mahammad, IITDM, Kancheepuram, noor@iitdm.ac.in	3. Ms. Ferni Ukrit, SRMIST

Course Code	18CSC002J	Course Name	OBJECT ORIENTED DESIGN AND PROGRAMMING	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:	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

Learning		
1	2	3
l	e	e
e	x	x
v	p	p
i	e	e
l	c	c
c	t	t
e	n	e
f	d	d

Program Learning Outcomes (PLO)														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H	F	I	A	N	S	H		I	C	F	L	F	F	F
n	r	e	n	c	d	n	H	n	c	r	i	F	S	S
g	o	s	a	d	c	v	t	d	n	o	f	C	C	C
i	b	i	l	e	i	i	h	i	n	j	e			
n	l	g	y	r	e	r	i	v	u	e				
e	e	n	s	n	t	o	c	i	n	c	L			
e	n	i		y	n	s	d	i	t					
r	d	s	T	n		u	c				n	l	2	3

CLR-6 : Create programs using object oriented approach and design methodologies for real-time application development

Course Learning Outcomes (CLO): At the end of this course, learners will be able to:

3	80	70	H	H	M	-	-	-	-	-	H	H	-	-	M	H	H
3	85	75	H	H	H	H	H	-	M	-	H	H	-	-	M	H	H
3	75	70	H	H	M	H	H	-	M	-	H	H	-	-	M	H	H
3	85	80	H	H	H	-	-	-	-	-	H	M	-	-	M	H	H
3	85	75	H	M	M	M	M	M	M	-	H	H	-	M	M	H	H
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
	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
	SLO-2	Constants, Pointers, Type Conversions	Method Overloading	Inheritance: Hybrid	Class Templates
S-3	SLO-1	Features: Class and Objects	Example for method overloading	Inheritance: Example Programs	Class Templates
	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
	SLO-2				
S-6	SLO-1	Feature :Class and Objects	Operator Overloading and types	Advanced Functions: Inline, Friend	Exceptional Handling: try and catch
	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	Lab 12 : UML Component, Deployment, Package diagram	Lab 15: Streams and File Handling
	SLO-2					

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

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, nrbbabu@iitm.ac.in	2. Mr. C.Arun, SRMIST
		3. Mr. Geogen George, SRMIST

Course Code	18CSC003J	Course Name	COMPUTER ORGANIZATION AND ARCHITECTURE	Course Category	C	Professional Core			
						L	T	P	C
						2	0	2	3

Pre-requisite Courses	Nil	Co-requisite Courses	Nil	Progressive Courses	Nil
------------------------------	-----	-----------------------------	-----	----------------------------	-----

Course Offering Department	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		
1	2	3
Identify the components of a computer system.	Explain the functions of various components of a computer system.	Execute simple programs on a computer system.
Apply Boolean algebra to design combinational and sequential logic circuits.	Design and simulate combinational and sequential logic circuits.	Implement and test combinational and sequential logic circuits.
Analyze the detailed operation of Basic Processing units and the performance of Pipelining.	Design and simulate Basic Processing units and Pipelining.	Implement and test Basic Processing units and Pipelining.
Analyze concepts of parallelism and multi-core processors.	Design and simulate parallelism and multi-core processors.	Implement and test parallelism and multi-core processors.

Program Learning Outcomes (PLO)														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Identify the computer hardware and how software interacts with computer hardware.	Apply Boolean algebra as related to designing computer logic, through simple combinational and sequential logic circuits.	Analyze the detailed operation of Basic Processing units and the performance of Pipelining.	Analyze concepts of parallelism and multi-core processors.	Apply Boolean algebra to design combinational and sequential logic circuits.	Analyze the detailed operation of Basic Processing units and the performance of Pipelining.	Analyze concepts of parallelism and multi-core processors.	Apply Boolean algebra to design combinational and sequential logic circuits.	Analyze the detailed operation of Basic Processing units and the performance of Pipelining.	Analyze concepts of parallelism and multi-core processors.	Apply Boolean algebra to design combinational and sequential logic circuits.	Analyze the detailed operation of Basic Processing units and the performance of Pipelining.	Analyze concepts of parallelism and multi-core processors.	Apply Boolean algebra to design combinational and sequential logic circuits.	Analyze the detailed operation of Basic Processing units and the performance of Pipelining.
H	H	-	-	-	-	-	-	M	L	-	M	-	-	-
H	H	H	-	H	-	-	-	M	L	-	M	-	-	-
H	H	H	H	-	-	-	-	M	L	-	M	-	-	-
H	-	-	H	-	-	-	-	M	L	-	M	-	-	-

Course Learning Outcomes (CLO):	At the end of this course, learners will be able to:
CLO-1 :	Identify the computer hardware and how software interacts with computer hardware
CLO-2 :	Apply Boolean algebra as related to designing computer logic, through simple combinational and sequential logic circuits
CLO-3 :	Analyze the detailed operation of Basic Processing units and the performance of Pipelining
CLO-4 :	Analyze concepts of parallelism and multi-core processors.

CLO-5 :	<i>Identify the memory technologies, input-output systems and evaluate the performance of memory system</i>	3	8 5	7 5	H	-	H	H	-	-	-	-	M	L	-	M	-	-	-
CLO-6 :	<i>Identify the computer hardware, software and its interactions</i>	3	8 5	7 5	H	H	H	H	H	-	-	-	M	L	-	M	-	-	-

Duration (hour)		12	12	12	12	12
S-1	SLO-1	<i>Functional Units of a computer</i>	<i>Addition and subtraction of Signed numbers,</i>	<i>Fundamental concepts of basic processing unit, ALU operation</i>	<i>Parallelism , Need, types of Parallelism</i>	<i>Memory systems -Basic Concepts Memory hierarchy</i>
	SLO-2	<i>Operational concepts</i>	<i>Problem solving</i>	<i>Execution of complete instruction, Branch instruction</i>	<i>applications of Parallelism</i>	<i>Memory technologies</i>
S-2	SLO-1	<i>Bus structures</i>	<i>Design of fast adders, Ripple carry adder</i>	<i>Multiple bus organization</i>	<i>Parallelism in Software</i>	<i>RAM, Semiconductor RAM</i>
	SLO-2	<i>Memory locations and addresses , Memory operations</i>	<i>Carry look ahead adder , Multiplication of positive numbers</i>	<i>Hardwired control , Generation of control signals</i>	<i>Instruction level parallelism , Data level parallelism</i>	<i>ROM, Types , Speed, size cost</i>
S 3-4	SLO-1	<i>Lab 1: To recognize various components of PC-Input Output systems Processing and Memory units</i>	<i>Lab4:Study of TASM Addition and Subtraction of 8-bit number</i>	<i>Lab-7: Design of Half Adder Design of Full Adder</i>	<i>Lab-10: Study of Array Multiplier Design of Array Multiplier</i>	<i>Lab-13: Study of Carry Save Multiplication Program to carry out Carry Save Multiplication</i>
	SLO-2		<i>Signed operand multiplication</i>	<i>Micro-programmed control-</i>	<i>Challenges in parallel processing</i>	<i>Cache memory , Mapping Functions</i>
S-5	SLO-1	<i>Instructions, Instruction sequencing</i>	<i>Fast multiplication- Bit pair recoding of Multipliers</i>	<i>Microinstruction , Micro-program Sequencing, Next address field</i>	<i>Architectures of Parallel Systems - Flynn's classification</i>	<i>Replacement Algorithms</i>
	SLO-2	<i>Addressing modes</i>	<i>Carry Save Addition of summands</i>	<i>Basic concepts of pipelining</i>	<i>SISD,SIMD, MIMD, MISD</i>	<i>Virtual Memory</i>
S-6	SLO-1	<i>Introduction to Microprocessor , Assembly language</i>	<i>Problem Solving</i>	<i>Pipeline Performance</i>	<i>Hardware multithreading , Coarse Grain parallelism, Fine Grain parallelism</i>	<i>Performance considerations of various memories</i>
	SLO-2	<i>Writing of assembly language programming</i>	<i>Lab 5: Addition of 16-bit number Subtraction of 16-bit number</i>	<i>Lab-8: Study of Ripple Carry Adder Design of Ripple Carry Adder</i>	<i>Lab-11: Study of Booth Algorithm</i>	<i>Lab-14: Understanding Processing unit Design of primitive processing unit</i>
S 7-8	SLO-1	<i>Lab-2:To understand how different components of PC are connected to work properly Assembling of System Components</i>	<i>Integer division – Restoring Division</i>	<i>Pipeline Hazards-Data hazards</i>	<i>Uni-processor and Multiprocessors</i>	<i>Input Output Organization , Need for Input output devices</i>
	SLO-2		<i>Non Restoring Division</i>	<i>Methods to overcome Data hazards</i>	<i>Multi-core processors</i>	<i>Memory mapped IO</i>
S-9	SLO-1	<i>ARM Processor: The thumb instruction set</i>	<i>Floating point numbers and operations</i>	<i>Instruction Hazards , Hazards on conditional and Unconditional Branching</i>	<i>Memory in Multiprocessor Systems</i>	<i>Program controlled IO</i>
	SLO-2	<i>Processor and CPU cores , Instruction Encoding format</i>	<i>Solving Problems</i>	<i>Control hazards , Influence of hazards on instruction sets</i>	<i>Cache Coherency and MESI protocol in Multiprocessor Systems</i>	<i>Interrupts-Hardware, Enabling and Disabling Interrupts</i>
S-10	SLO-1	<i>Memory load and Store instruction in ARM</i>	<i>Lab-6: Multiplication of 8-bit number Factorial of a given number</i>	<i>Lab-9: Study of Carry Look-ahead Adder Design of Carry Look-ahead Adder</i>	<i>Lab-12: Program to carry out Booth Algorithm</i>	<i>Lab-15: Understanding Pipeline concepts Design of basic pipeline.</i>
	SLO-2	<i>Basics of IO operations.</i>				
S 11-12	SLO-1	<i>Lab -3To understand how different components of PC are connected to work properly Disassembling of System Components</i>				
	SLO-2					

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

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. Shanthy, ANNA University Chennai, a.p.shanthy@cs.annauniv.edu	1. Dr. V. Ganapathy, SRMIST
		2. Dr. C. Malathy, SRMIST
		3. Mrs M.S.Abirami, SRMIST

Course Code	18CSE001J	Course Name	OPERATING SYSTEMS	Course Category	E	<i>Professional Elective</i>	L	T	P	C
							2	0	2	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:
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

Learning		
1	2	3
I	E	E
e	x	x
v	p	p

Program Learning Outcomes (PLO)														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H	F	I	A	N	S	E	E	I	C	F	I	F	F	F
n	r	e	n	c	c	n	t	n	c	r	i	S	S	S
g	d	s	a	d	c	v	h	d	n	c	f	C	C	C

	Create								
	Total	100 %	100 %	100 %	100 %				-

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

Course Code	18CSE002T	Course Name	SOFTWARE ENGINEERING AND PROJECT MANAGEMENT	Course Category	E	Professional Elective	L	T	P	C
							2	1	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:
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

Course Learning Outcomes (CLO):	At the end of this course, learners will be able to:
---------------------------------	--

Learning		
1	2	3
Identify the process of project life cycle model and process	8/5	8/0
Analyze and specify software requirements through a productive working Relationship with project stakeholders	8/0	7/5
Design the system based on Functional Oriented and Object Oriented Approach for Software Design.	8/5	8/5
Develop the correct and robust code for the software products	8/5	8/5
Perform by applying the test plan and various testing techniques	8/5	7/5

Program Learning Outcomes (PLO)														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H	H	L	-	-	-	L	-	H	H	M	M	-	-	-
H	H	H	H	H	-	M	-	H	H	H	M	-	-	-
H	H	M	H	H	M	M	L	H	H	M	-	-	-	-
H	H	H	-	H	-	-	M	H	M	H	-	-	-	-
H	M	M	M	M	M	M	-	H	H	-	M	-	-	-

Level 2	Apply	40%	-	40%	-	40%	-	40%	-	40%	-
	Analyze										
Level 3	Evaluate	20%	-	30%	-	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. Mr. Girish Raghavan, Wipro Technologies	1. Dr. LathaParthiban, 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	18CSE003J	Course Name	PYTHON PROGRAMMING	Course Category	E	Professional Elective	L	2	T	0	P	2	C	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):	<i>The purpose of learning this course is to:</i>
CLR-1 :	<i>Create an environment for programming in Python with basic structures and syntax</i>
CLR-2 :	<i>To learn and apply the concepts of Lists, Modules and Packages in Python</i>
CLR-3 :	<i>To learn and apply the concepts of Tuples, Dictionaries, Sets and Files in Python</i>
CLR-4 :	<i>To learn and use Exception handling and Object Oriented concepts in Python</i>
CLR-5 :	<i>To learn and apply the advanced concepts like Iterator, Generator, Regular expressions in Python</i>
CLR-6 :	<i>To learn and apply the Graphical User Interface modules in Python</i>

Course Learning Outcomes (CLO):	<i>At the end of this course, learners will be able to:</i>
--	---

Learning		
1	2	3
1	2	3
3	8	8
3	8	8
3	8	8
3	8	8
3	8	8
3	8	8

Program Learning Outcomes (PLO)														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H	H	H	H	H	-	-	L	M	M	L	M	-	M	-
H	H	H	H	H	-	-	L	M	M	L	M	-	-	-
H	H	H	H	H	-	-	L	M	M	L	M	-	-	-
H	H	H	H	H	-	-	L	M	M	L	M	-	-	-
H	H	H	H	H	-	-	L	M	M	L	M	-	-	-

CLO-1 :	<i>Create Programs using basic structures and syntax in Python</i>	3	8	8
CLO-2 :	<i>Create Programs using the concepts of Lists, Modules and Packages in Python</i>	3	8	8
CLO-3 :	<i>Create Programs using the concepts of Tuples, Dictionaries, Sets and Files in Python</i>	3	8	8
CLO-4 :	<i>Create Programs using Exception handling and Object Oriented concepts in Python</i>	3	8	8
CLO-5 :	<i>Create Programs using the advanced concepts like Iterator, Generator, Regular expressions in Python</i>	3	8	8
CLO-6 :	<i>Create Programs using the Graphical User Interface modules in Python</i>	3	8	8

Duration (hour)	12	12	12	12	12
S-1	SLO-1	<i>Using the Python Interpreter Invoking the Interpreter,</i>	<i>Introduction to ListsAnd List Operations</i>	<i>Introduction to Tuples</i>	<i>Errors - Syntax Errors</i>
	SLO-2	<i>Argument Passing Interactive Mode</i>	<i>list slices, list methods, list loop</i>	<i>Tuple Operations</i>	<i>Exceptions - Handling Exceptions</i>

S-2	SLO-1	Introduction to Python Using Python as a Calculator , Numbers, Strings , Lists Keywords and Identifiers	mutability, aliasing, cloning lists, list parameters	tuple assignment , tuple as return value	Raising Exceptions User-defined Exceptions	Closure
	SLO-2	Statements & Comments Variables , Datatypes, Type Conversion, I/O and import Operators , Namespace	Arbitrary Argument Lists	Examples in Tuples	Defining Clean-up Actions - Predefined Clean-up Actions	Decorators
S 3-4	SLO-1	Lab-1 : Using the Python Interpreter- Simple problems in Input output	Lab-4: Problems in Lists	Lab-7: Problems on Tuples	Lab-10: Problems in Exception handling	Lab-13: Problems using Iterator, Generator
	SLO-2					
S-5	SLO-1	Flow Control - if...else	Using Lists as Stacks	Introduction to Dictionaries	Classes Definition	Context Managers
	SLO-2	for Loop while Loop	Using Lists as Queues	Examples on Dictionaries	Names and Objects - Method Objects	RegEx
S-6	SLO-1	break and continue pass Statements	List Comprehensions	Introduction to Set	Class and Instance Variables	String manipulation
	SLO-2	The range() Function	Nested List Comprehensions	Examples on Set	Python Scopes and Namespaces	Unicode
S 7-8	SLO-1	Lab-2 : Simple problems in flow control	Lab-5: Problems sing List as Stacks and Queue	Lab-8: Problems in Dictionaries and Set	Lab-11: Problems using Classes and Objects	Lab-14: Problems using Regular Expressions
	SLO-2					
S-9	SLO-1	Defining Functions Default Argument Values Keyword Arguments	Making your own module, namespaces	Introduction to Files – types – formatting- command line arguments	Inheritance	Introduction to GUI
	SLO-2	Special parameters Positional-or-Keyword Arguments Positional-Only Parameters Keyword-Only Arguments	Executing modules as scripts The Module Search PathStandard library modules	Operations on Files – opening and closing, reading and writing , file position	Multiple Inheritance	Tkinter Module
S-10	SLO-1	Recursion	Packages in python Importing * From a Package	Creating records and manipulating them using files	Operator Overloading	Basic Widgets
	SLO-2	Anonymous FunctionGlobal, Local and Nonlocal	Intra-package References Packages in Multiple Directories	Formatted String Literals The String format() Method Manual String Formatting	Examples using Overloading	managing images - Plotting
S-11-12	SLO-1	Lab-3: problems using functions and recursion concept	Lab-6: Problems using Modules and Packages	Lab-9: Problems on File and its operations	Lab-12: Problems using Inheritance and Overloading	Lab-15: Problems using Tkinter module
	SLO-2					

Learning Resources	1. Reema Thareja, “Python Programming “, Oxford University Press, 2017	5. Amit Saha, Doing Math with Python: Use Programming to Explore Algebra, Statistics, Calculus and More, Kindle Edition, 2015
	2. S.A. Kulkarni , “Problem Solving and Python Programming”, Yes Dee Publishing, 2017	
	3. Daniel Liang , “Introduction to Programming using Python” , Pearson India Education Services Pvt. Ltd., 2017.	
	4. Robert Sedgewick, Kevin Wayne, Robert Dondero, —Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016	

Learning Assessment					
Bloom’s Level of	Continuous Learning Assessment (50% weightage)				Final Examination (50% weightage)
	CLA – 1 (10%)	CLA – 2 (15%)	CLA – 3 (15%)	CLA – 4 (10%)#	

CLO-1:	Acquire the knowledge of mathematics and engineering principles for the basics of Formal Language	3	8	7	M	H	-	H	L	-	-	-	L	L	-	H	-	-	-
CLO-2:	Acquire the ability to identify specification of a Regular language's with Automata	3	8	7	M	H	L	M	L	-	-	-	M	L	-	H	-	-	-
CLO-3:	Acquire knowledge of Context free Grammar and simplify using normal forms	3	7	7	M	H	M	H	L	-	-	-	M	L	-	H	-	-	-
CLO-4:	Understand the concepts of push down automata and CFL .	3	8	8	M	H	M	H	L	-	-	-	M	L	-	H	-	-	-
CLO-5:	Apply the knowledge to turning machine and its methods	3	8	7	H	H	M	H	L	-	-	-	M	L	-	H	-	-	-
CLO-6:	Design the computational and acceptor machines using FA, PDA and Turing machines	3	8	7	L	H	-	H	L	-	-	-	L	L	-	H	-	-	-

Duration (hour)	9	9	9	9	9	
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 ϵ - 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 ϵ - moves	Simplification of CFG : Null productions	Problems related to Equivalence of PDA to CFG		Time and Space complexity of TMs
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 N DFA's with and without ϵ -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 N DFA's with and without ϵ -moves	Problems related to CNF	Problems related to Equivalence of CFG to PDA	Checking off symbols	
S-8	SLO -1	Minimization of DFA	Greiback Normal form	Pumping lemma for CFL	Modifications of Turing Machine	NP Completeness
	SLO -2	Problems related to Minimization of DFA			Multi-tape Turing Machine	

S-9	SLO -1	<i>Regular Languages : Equivalence of Finite Automata and Regular Languages</i>	<i>Problems related to GNF</i>	<i>Problems based on pumping Lemma</i>	<i>Non-Deterministic Turing Machine</i>	<i>Problem discussion</i>
	SLO -2	<i>Equivalence of Finite Automata and Regular Grammars</i>			<i>Semi-Infinite Tape Turing Machine</i>	

Learning Resources	<p>1. Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory, Languages and Computations", Second Edition, Pearson Education, 2008.</p> <p>2. Michael Sipser, "Introduction to the Theory of Computation" Cengage Learning, 2012.</p> <p>4. John.C.Martin, "Introduction to Languages and the Theory of Computation" McGraw-Hill Education, 01- May-2010.</p> <p>5. Kamala Krithivasan, Rama.R," Introduction to Formal Languages, Automata Theory and Computation", Pearson Education India, 01-Sep-2009.</p> <p>6. Peter Linz , "An introduction to formal languages and automata", Jones & Bartlett Learning, 2001.</p>
--------------------	--

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	40 %	-	30 %	-	30 %	-	30 %	-	30%	-
Level 2	Understand	40 %	-	40 %	-	40 %	-	40 %	-	40%	-
	Apply	40 %	-	40 %	-	40 %	-	40 %	-	40%	-
Level 3	Analyze	20 %	-	30 %	-	30 %	-	30 %	-	30%	-
	Evaluate	20 %	-	30 %	-	30 %	-	30 %	-	30%	-
	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
		Dr.R.AnnieUthra
		Dr.Jeyasudha

Course Code	18CSE005J	Course Name	COMPUTER NETWORKS	Course Category	E	Professional Elective	L	T	P	C
							2	0	2	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:	Understand the evolution of computer networks using the layered network architecture		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CLR-2:	Understand the addressing concepts and learn networks devices																			
CLR-3:	Design computer networks using subnetting and routing concepts																			
CLR-4:	Understand the error types, framing, flow control																			
CLR-5:	Understand the various Medium Access Control techniques and also the characteristics of physical layer functionalities																			
CLR-6:	Understand basic network administration																			
Course Learning Outcomes (CLO):		At the end of this course, learners will be able to:	Learning			Program Learning Outcomes (PLO)														
CLO-1:	Acquire the basics of computer network and its architecture		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CLO-2:	Acquire the knowledge of various networks devices and addressing methods																			
CLO-3:	Ability to design the network routing methods																			
CLO-4:	Acquire the various error codes and framing concepts																			
CLO-5:	Ability to understand the physical layer functions and components																			
CLO-6:	Ability to design a computer network using a switch and router																			

Duration (hour)	12	12	12	12	12
-----------------	----	----	----	----	----

Level 3	Evaluate	1	1	1	1	15	1	1	1	15	1
	Create	0	0	5	5	%	5	5	5	%	5
	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. Sricharan, Wipro Technologies, Chennai	1. Dr.Noor Mahammad, IIITDM, Kancheepuram, noor@iiitdm.ac.in	1. Mr. K. Venkatesh, SRMIST
2.	2.	2. Ms.D. Anitha, SRMIST
	3.	3. Ms. Ferni Ukrit, SRMIST

CLO-6 :	Appreciate the fundamental concepts of transaction processing- concurrency control techniques and recovery procedures.	3	8	7		H	L	L	L	L		-	-	L	L	L		
---------	--	---	---	---	--	---	---	---	---	---	--	---	---	---	---	---	--	--

Duration (hour)	12		12		12		12		12	
S-1	SL O-1	What is Database Management System	Database Design	Basics of SQL-DDL,DML,DCL,TCL , Structure Creation, alternation	Relational Algebra – Fundamental Operators and syntax, relational algebra queries	Transaction concepts, properties of transactions,				
	SL O-2	Advantage of DBMS over File Processing System	Design process	Defining Constraints-Primary Key, Foreign Key, Unique, not null, check, IN operator	Tuple relational calculus	Serializability of transactions,				
S-2	SL O-1	Introduction and applications of DBMS	Entity Relation Model	Functions-aggregation functions	Pitfalls in Relational database, Decomposing bad schema	testing for serial inability, System recovery,				
	SL O-2	Purpose of database system, Views of data	ER diagram	Built-in Functions-numeric, date, string functions, string functions, Set operations,	Functional Dependency – definition, trivial and non-trivial FD	Concurrency Control				
S-3-4	SL O-1	Lab 1: SQL Data Definition Language Commands on sample exercise* The abstract of the project to construct database must be framed	Lab 7 : Join Queries on sample exercise. * Frame and execute the appropriate DDL,DML,DCL,TCL for the project	Lab 10: 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					
	SL O-2									
S-5	SL O-1	Database system Architecture	Keys , Attributes and Constraints , Mapping Cardinality	Sub Queries, correlated sub queries ,Nested Queries	closure of FD set , closure of attributes	Two- Phase Commit protocol, Recovery and Atomicity				
	SL O-2						irreducible set of FD			
S-6	SL O-1	Data Independence	Extended ER - Generalization	Views and its Types, Transaction Control Commands	Normalization – 1NF, 2NF, 3NF,	Log-based recovery				
	SL O-2	The evolution of Data Models	Specialization and Aggregation	Commit, Rollback, Save point	Decomposition using FD-dependency	concurrent executions of transactions				
S-7-8	SL O-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				
	SL O-2									
S-9	SL O-1	Degrees of Data Abstraction	ER Diagram Issues	PL/SQL Concepts- Cursors	BCNF, examples	Locking mechanism, solution to concurrency related problems				
	SL O-2	Database Users and DBA	Weak Entity	Stored Procedure						
S-10	SL O-1	Database Languages	Relational Model	Functions Triggers and Exceptional Handling	Multi- valued dependency, 4NF	Deadlock				

	SL O- 2		Conversion of ER to Relational Table	Query Processing	Join dependency and 5NF	two-phase locking protocol, Isolation, Intent locking
S 11-12	SL O- 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
	SL O- 2					

Learning Resources	<ol style="list-style-type: none"> 1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, Database System Concepts, Sixth Edition, Tata McGraw Hill, 2011. 2. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 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. 	<ol style="list-style-type: none"> 4. Martin Gruber, Understanding SQL, Sybex, 1990 5. Shrad Maheshwari, Introduction to SQL and PL/SQL, 2nd ed., Laxmi Publications, 2016. 6. Raghurama Krishnan, Johannes Gehrke, Database Management Systems, 3rd Edition, McGraw Hill 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	2 0	2 0	1 5	1 5	15 %	1 5	1 5	1 5	15 %	1 5
	Understand	%	%	%	%	%	%	%	%	%	%
Level 2	Apply	2 0	2 0	2 0	2 0	20 %	2 0	2 0	2 0	20 %	2 0
	Analyze	%	%	%	%	%	%	%	%	%	%
Level 3	Evaluate	1 0	1 0	1 5	1 5	15 %	1 5	1 5	1 5	15 %	1 5
	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	18CSE007 J	Course Name	COMPILER DESIGN	Course Category	E	Professional Elective	L	T	P	C
							2	0	2	3

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

Course Learning Rationale (CLR):		Learning			Program Learning Outcomes (PLO)															
The purpose of learning this course is to:		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
CLR-1:	Utilize the mathematics and engineering principles for the Design of Compilers	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Society & Culture	Environment & Sustainability	Individual & Team Work									
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																			
Course Learning Outcomes (CLO):		At the end of this course, learners will be able to:																		
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	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	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	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	H

CLR-5:	Analyze the methods of implementing a Code Generator for compilers	3	85	7 5	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	7 0	H	H	H	H	M	L	L	L	M	M	L	H	H	H	H

Duration (hour)	12	12	12	12	12	
S-1	SLO-1	Compilers – Analysis of the source program	Syntax Analysis Definition - Role of parser	Bottom Up Parsing , Reductions	Intermediate Code Generation	Code optimization -Principal Sources of Optimization
	SLO-2	Phases of a compiler – Cousins of the Compiler	Lexical versus Syntactic Analysis	Handle Pruning	Intermediate Languages - prefix - postfix -Quadruple - triple - indirect triples Representation	Function Preserving Transformation
S-2	SLO-1	Grouping of Phases – Compiler construction tools	Representative Grammars,	Shift Reduce Parsing, problems and conflicts	Syntax tree- Evaluation of expression - three-address code	Loop Optimization
	SLO-2	Lexical Analysis – Role of Lexical Analyzer, Input Buffering	Elimination of Ambiguity, Left Recursion ,Left Factoring		Synthesized attributes – Inherited attributes, Declarations	Optimization of basic Blocks- Building Expression of DAG
S-3-4	SLO-1	Lab 1 - Implementation of Lexical Analyzer	Lab 4 Elimination of Ambiguity, Left Recursion and Left Factoring	Lab 7 - Shift Reduce Parsing	Lab 10-Intermediate code generation – Postfix, Prefix	Lab 13 Implementation of DAG
	SLO-2					
S-5	SLO-1	Finite automation - deterministic, non deterministic	Top down parsing	LR Parsers- Why LR Parsers	Assignment Statements , Case Statements	Peephole Optimization , Basic Blocks,
	SLO-2	Transition Tables	Recursive Descent Parsing, back tracking	Items and LR(0) Automaton, Closure of Item Sets, LR Parsing Algorithm	Boolean Expressions, Back patching – Procedure calls	Flow Graphs , Next -Use Information
S-6	SLO-1	Acceptance of Input Strings by Automata - State Diagrams and Regular Expressions	Computation of FIRST, FOLLOW	Operator Precedence Parser	Code Generation- Issues in the design of code generator	Introduction to Global Data Flow Analysis
	SLO-2	Conversion of regular expression to NFA – Thompson's	Problems related to FIRST, FOLLOW	Computation of LEADING ,TRAILING	The target machine – Runtime Storage management	Computation of gen and kill, in and out
S-7-8	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-9	SLO-1	Conversion of NFA to DFA - Simulation	Construction of a predictive parsing table	SLR Grammars- SLR Parsing Tables	A simple Code generator - Code Generation Algorithm	Parameter Passing - Runtime Environments
	SLO-2	Converting Regular expression directly to DFA	Predictive Parsers LL(1) Grammars	Problems related to SLR	Register and Address Descriptors	Source Language issues
S-10	SLO-1	Minimization of DFA	Predictive Parsing Algorithm	Construction of Canonical LR(1) and LALR	Generating Code of Assignment Statements	Storage Organization - Activation Records
	SLO-2	Design of lexical analysis (LEX)	Error Recovery in Predictive Parsing- Non Recursive Predictive Parser	Problems related to Canonical LR(1) and LALR Parsing Table	Cross Compiler – T diagrams -Issues in Cross compilers	Storage Allocation strategies
S-11-12	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	<ol style="list-style-type: none"> 1. AlfredV Aho, Jeffery D Ullman, Ravi Sethi, "Compilers, Principle techniques and tools", Pearson Education 2011 2. S. Godfrey Winster, S. Aruna Devi, R. Sujatha, "Compiler Design", Yesdee Publishing Pvt. Ltd, 2016 3. William M. Waite and Gerhard Goos. Compiler Construction. Springer-Verlag, New York, 2013. 	<ol style="list-style-type: none"> 4. K. Muneeswaran, "Compiler Design", Oxford Higher Education, Fourth edition 2015 5. David Galles, "Modern Compiler Design", Pearson Education, Reprint 2012. 6. Raghavan V., "Principles of Compiler Design", Tata Mc Graw Hill Education Pvt. 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	2	2	1	1	15	1	1	1	15	1
	Understand	0	0	5	5	%	5	5	5	%	5
Level 2	Apply	2	2	2	2	20	2	2	2	20	2
	Analyze	0	0	0	0	%	0	0	0	%	0
Level 3	Evaluate	1	1	1	1	15	1	1	1	15	1
	Create	0	0	5	5	%	5	5	5	%	5
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

Duration (hour)		12	12	12	12	12
S - 1	SL O- 1	<i>Introduction to AI-AI techniques</i>	<i>Searching techniques- Uniformed search- General search Algorithm</i>	<i>Knowledge and reasoning- Approaches and issues of knowledge reasoning</i>	<i>Planning- Planning problems, Simple planning agent</i>	<i>Expert system-Architecture , Pros and Cons of expert system</i>
	SL O- 2	<i>Problem solving with AI ; Problem solving process, Formulating problems</i>	<i>Uniformed search Methods- Breadth first search</i>	<i>Knowledge base agents-Logic Basics Logic-Propositional logic- syntax ,semantics and inferences</i>	<i>Planning languages , Blocks world</i>	<i>Rule based systems,</i>
S - 2	SL O- 1	<i>AI Models, Data acquisition and learning aspects in AI</i>	<i>Uniformed search Methods- Depth first search , Depth limited search</i>	<i>Reasoning patterns, Predicate logic – Syntax and semantics, instance and is relationship</i>	<i>Goal stack planning , Mean Ends Analysis</i>	<i>Frame based expert system</i>
	SL O- 2	<i>Problem types and characteristics ,Problem space and search</i>	<i>Uniformed search Methods- Iterative Deepening search , Bi-directional search</i>	<i>Unification and Resolution</i>	<i>Non-linear Planning , Conditional planning, Reactive planning</i>	<i>Case study</i>
S 3 - 4	SL O- 1	<i>Lab 1: Implementation of toy problems</i>	<i>Lab4: Implementation and Analysis of DFS and BFS for an application</i>	<i>Lab 7: Implementation of unification and resolution for real world problems.</i>	<i>Lab 10 :Implementation of block world problem</i>	<i>Natural language processing- Levels of NLP</i>
	SL O- 2					
S - 5	SL O- 1	<i>Intelligent agent , Types of agents</i>	<i>Informed search- Generate and test, Best First search</i>	<i>Knowledge representation using rules and semantic nets</i>	<i>Learning- Machine learning- Goals and Challenges</i>	<i>Syntactic and Semantic Analysis</i>
	SL O- 2	<i>Rationality and Rational agent with performance measures</i>	<i>Informed search-A* Algorithm</i>	<i>Knowledge representation using frames</i>	<i>Learning concepts, models</i>	<i>Information retrieval ,Information Extraction</i>
S - 6	SL O- 1	<i>Flexibility and Intelligent agents,</i>	<i>AO* Algorithm</i>	<i>Inferences, Uncertain Knowledge and reasoning- Methods</i>	<i>Artificial neural network based learning- Back propagation</i>	<i>Machine translation</i>

	SL O- 2	Task environment and its properties	Local search Algorithms-Hill Climbing, Simulated Annealing	Bayesian probability and belief network	Support vector machines , Reinforcement learning	NLP Applications
S 7 - 8	SL O- 1	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
	SL O- 2					
S - 9	SL O- 1	Constraint satisfaction problems(CSP)	Adversarial search Methods-Game playing- Important concepts ,knowledge structure	Probabilistic reasoning	Adaptive learning	Advance topics in Artificial Intelligence- Cloud Computing and intelligent agent
	SL O- 2	CSP as a search problem- constrains and representation, Backtracking, Role of heuristic	Game as a search problem-Mini max approach - Mini max Algorithm	Forward and backward reasoning	Multi agent based learning, Ensemble learning	Business intelligence and analytics
S - 1 0	SL O- 1	CSP-Forward checking and constraint propagation	Alpha beta pruning	Other uncertain techniques-Data mining	Learning for decision making	Sentiment Analysis , Planning and logic in intelligent agents
	SL O- 2	CSP-Intelligent backtracking ,Crypto arithmetic puzzles	Game theory problems	Fuzzy logic , Dempster -shafer theory	Distributed learning, Speedup learning	Deep learning Algorithms
S 1 1 - 1 2	SL O- 1	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	Lab 12: Development of ensemble model for an application	Lab 15: Applying deep learning methods to solve an application.
	SL O- 2					

Learning Resources	<ol style="list-style-type: none"> 1. Parag Kulkarni, Prachi Joshi, Artificial Intelligence –Building Intelligent Systems, 1st ed., PHI learning, 2015 2. Deepak Kemhani, First course in Artificial Intelligence, McGraw Hill Pvt Ltd, 2013 3. Stuart J. Russell, Peter Norvig , Artificial Intelligence –A Modern approach, 3rd Pearson Education, 2016 	<ol style="list-style-type: none"> 4. Prateek Joshi, Artificial Intelligence with Python, 1st ed., Packt Publishing, 2017 5. Denis Rothman, Artificial Intelligence by Example, 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		2	1	1	1	1	1	1	1	1
	Understand	20%	0%	0%	0%	5%	5%	5%	5%	15%	5%
Level 2	Apply		2	2	2	2	2	2	2	2	2
	Analyze	20%	0%	0%	0%	0%	0%	0%	0%	20%	0%
Level 3	Evaluate		1	2	2	1	1	1	1	1	1
	Create	10%	0%	0%	0%	5%	5%	5%	5%	15%	5%
	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.ietf@auxolabs.in	1. Dr. Chitrakala, Anna University, au.chitras@gmail.com	1. Dr. M. Pushpalatha, SRMIST
2.	2.	2. Dr. G. Vadivu, SRMIST
	3.	3. Dr. C. Lakshmi, SRMIST

CLO-4 :	<i>Acquire the knowledge of web security attack and prevention mechanism</i>	2	80	75
CLO-5 :	<i>Acquire the knowledge of wireless network security and prevention mechanism</i>	2	75	85

H								H							
H								H							

Duration (hour)	9		9		9		9		9	
S-1	SLO-1	Networking Devices(Layer1,2)	Overview of IPSEC- Security Associations, Security Association Database	Security Services for E-mail	SSL/TLS Basic Protocol	Wireless Security:IEEE 802.11 Wireless LAN				
	SLO-2	Networking Devices(Layer 3)	Security Policy databases , AH and ESP	Security Services for E-mail	SSL/TLS Basic Protocol	Wireless Security:IEEE 802.11 Wireless LAN				
S-2	SLO-1	Different types of network layer attacks	Tunnel and Transport mode	Establishing keys	computing the keys	Authentication				
	SLO-2	Different types of network layer attacks	IP header Protection	Establishing Public and secret keys	computing the keys	Authentication and confidentiality				
S-3	SLO-1	Firewall- ACL	IP and IPv6	Privacy	client authentication	Cellphone Security				
	SLO-2	Packet Filtering	IPV4 and IPV6 header	End-to end Privacy, Privacy with distribution List Exploders	client authentication	GSM (2G) Security				
S-4	SLO-1	DMZ, Alerts	Authentication Header	Authentication of the source	PKI as deployed by SSL	Security in UMTS (3G)				
	SLO-2	Audit Trails	Mutable, Immutable and Mutable but predictable	Based on public key technology and secret keys and with distribution list	PKI as deployed by SSL	Security in UMTS (3G)				
S-5	SLO-1	IDS	Encapsulation Security Payload(ESP)	Message Integrity	SSLAttacks fixed in v3	Wireless LAN Vulnerabilities				
	SLO-2	Advantages and Disadvantages of IDS(Need of IPS)	Internet Key Exchange	Non-repudiation	SSLAttacks fixed in v3	Phishing				
S-6	SLO-1	Advantages of IPS ove IDS	Phases of IKE	Introduction and Overview of PGP	Exportability	Buffer Overflow				
	SLO-2	IPS	Phase I IKE- Modes and key types	Efficient Encoding	Exportability	Buffer Overflow				
S-7	SLO-1	IPS Types- Signature based	Phase I IKE Protocols	Certificate and key revocation	Encoding	Format String Attacks				
	SLO-2	Anomaly based, Policy based	Phase I IKE Protocols	Singature types, Private key, Fing types	Encrypted Record	Cross-site Scripting (XSS)				
S-8	SLO-1	IPS Types - Honeygot based	Phase II IKE	Anomalies	Handshake messages	SQL Injection				
	SLO-2	Applications	Phase II IKE	Object Format	Changechipherspec and Alerts	SQL Injection				
S-9	SLO-1	Malicious Software	ISAKMP/IKE Encoding	S/MIME	SET	Case Studies: Secure Inter-branch Payment Transactions				
	SLO-2	Malicious Software	ISAKMP/IKE Encoding	S/MIME	SET	Virtual Elections				

Learning Resources	1. Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security, Prentice Hall of India, 2002.	3. William Stallings, Cryptography and Network Security - Principles and Practice, 7th edition, Pearson Publication, 2017
	2. Bernard Menezes - Network Security and Cryptography- Cengage Learning. 2010.	4. Cryptography and network security , AtulkahateTata McGraw-Hill 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	40 %	-	30 %	-	30 %	-	30 %	-	30%	-	
	Understand											
Level 2	Apply	40 %	-	40 %	-	40 %	-	40 %	-	40%	-	
	Analyze											
Level 3	Evaluate	20 %	-	30 %	-	30 %	-	30 %	-	30%	-	
	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
Mr. M. Sudhakar, M.Tech, (Ph.D)-IIT, IT Infrastructure Service, Tata Consultancy Services.	Dr. P. Yogesh, Associate Professor, Dept of Information Science and Technology, College of Engineering, Guindy,	Dr. A. Jeyasekar, Associate Professor Dr. J. Femilda, Associate Professor Mrs. G. Sujatha, Assistant Professor

--	--	--

Course Code	18CSE010T	Course Name	DATA MINING AND ANALYTICS	Course Category	E	Professional Elective	L	T	P	C
							3	0	0	3

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

Course Learning Rationale (CLR):	<i>The purpose of learning this course is to:</i>
CLR-1 :	<i>Understand the concepts of Data Mining</i>
CLR-2 :	<i>Familiarize with Association rule mining</i>
CLR-3 :	<i>Familiarize with various Classification algorithms</i>
CLR-4 :	<i>Understand the concepts of Cluster Analysis</i>
CLR-5 :	<i>Familiarize with Outlier analysis techniques</i>
CLR-6 :	<i>Familiarize with applications of Data mining in different domains</i>

Course Learning Outcomes (CLO):	<i>At the end of this course, learners will be able to:</i>
--	---

Learning		
1	2	3
Level of Thinking (Bloom)	Expected Proficiency (%)	Expected attainment (%)
2	80	85
2	75	80
2	85	80
2	80	75
2	75	85
2	80	85

Program Learning Outcomes (PLO)														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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
H	H													
		H	H	H										
			H	H										
	H		H	H										
H	H			H				H			H			

Duration (hour)	9	9	9	9	9	
S-1	SLO-1	Why Data mining? What is Data mining ?	Mining frequent patterns: Basic concepts	Classification: Basic concepts	Cluster Analysis: Introduction	Outliers: Introduction
	SLO-2	Kinds of data meant for mining	Market Basket Analysis	General approach to Classification	Requirements and overview of different categories	Challenges of outlier detection
S-2	SLO-1	Kinds of patterns that can be mined	Frequent itemsets, Closed itemsets	Decision tree induction	Partitioning method: Introduction	Outlier detection methods: Introduction
	SLO-2	Applications suitable for data mining	Association rules-Introduction	Algorithm for Decision tree induction	k-means	Supervised and Semi-supervised methods
S-3	SLO-1	Issues in Data mining	Apriori algorithm-theoretical approach	Numerical example for Decision tree induction	k-medoids	Unsupervised methods
	SLO-2	Data objects and Attribute types	Apply Apriori algorithm on dataset-1	Attribute selection measure	Hierarchical method: Introduction	
S-4	SLO-1	Statistical descriptions of data	Apply Apriori algorithm on dataset-2	Tree pruning	Agglomerative vs. Divisive method	Statistical and Proximity based methods
	SLO-2		Generating Association rules from frequent itemsets	Scalability and Decision tree induction	Distance measures in algorithmic methods	
S-5	SLO-1	Need for data preprocessing and data quality	Improving efficiency of Apriori	Bayes' Theorem	BIRCH technique	Statistical approaches
	SLO-2			Naïve Bayesian Classification		

S-6	SLO-1	Data cleaning	Pattern growth approach	IF-THEN rules for classification	DBSCAN technique	Statistical data mining
	SLO-2	Data integration		Rule extraction from a decision tree		
S-7	SLO-1	Data reduction	Mining frequent itemsets using Vertical data format	Metrics for evaluating classifier performance	STING technique	Data mining and recommender systems
	SLO-2		Strong rules vs. weak rules	Cross validation		
S-8	SLO-1	Data transformation	Association analysis to Correlation analysis	Bootstrap	CLIQUE technique	Data mining for financial data analysis
	SLO-2			Ensemble methods-Introduction		
S-9	SLO-1	Data cube and its usage	Comparison of pattern evaluation measures	Bagging and Boosting	Evaluation of clustering techniques	Data mining for Intrusion detection
	SLO-2			Random Forests: Introduction		

Learning Resources	1. Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques", 3 rd Edition, Morgan Kauffman Publishers, 2011.	
---------------------------	--	--

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	40 %	-	30 %	-	30 %	-	30 %	-	30%	-
	Understand										
Level 2	Apply	40 %	-	40 %	-	40 %	-	40 %	-	40%	-
	Analyze										
Level 3	Evaluate	20 %	-	30 %	-	30 %	-	30 %	-	30%	-
	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.V.Selvakumar, Hexaware Technologies, selvakumarv@hexaware.com	1. Dr.Latha Parthiba, Pondicherry University, lathaparthiban@yahoo.com	1. Mr.L.N.B.Srinivas, SRMIST
		2. Mr.S.Karthick, SRMIST
		3. Dr.V.V.Ramalingam, SRMIST