Sl. No.	Subject Code	SEMESTER I	L	T	P	C
1	ECS 5101	Design and Analysis of Algorithms	3	0	2	4
2	ECS 5102	Foundations of Computer Systems	3	0	2	4
3	EMC 5103	Probability and Statistics	3	0	2	4
4	EHS 5104	Technical Writing and Soft Skill	1	2	2	4
5	EXX 61YY	DE-1	3	0	0	3
	Total		**13**	**2**	**8**	**19**

Sl. No.	Subject Code	SEMESTER II	L	T	P	C
1	ECS 5201	Artificial Intelligence	3	0	2	4
2	EMC 5202	Numerical Linear Algebra and Optimization Techniques	3	0	2	4
3	ECS 5203	Theoretical Computer Science	3	1	0	4
4	ERM 5204	Research Methodology	3	1	0	4
5	EXX 62YY	DE-2	3	0	0	3
	Total		**15**	**2**	**4**	**19**

Sl. No.	Subject Code	SEMESTER III	L	T	P	C
1	EXX 63YY	DE-3	3	0	0	3
2	EIK 6101	IKS	3	0	0	3
3	EXX 6199	Project I	0	0	36	18
	Total		**6**	**0**	**36**	**24**

Sl. No.	Subject Code	SEMESTER IV	L	T	P	C
1	EXX 64YY	DE-4	3	0	0	3
2	EXX 6299	Project II	0	0	38	19
	Total		**3**	**0**	**38**	**22**

Course number	**ECS 5101**
Course Credit	L-T-P-C: 3-0-2-4
Course Title	**Design and Analysis of Algorithms**
Learning Mode	Online
Learning Objectives	The objective of this course is to equip students with a solid understanding of data structures and algorithms, enabling them to design, analyze, and implement efficient algorithms to solve complex computational problems. The course covers fundamental topics such as data structures, complexity analysis, sorting and searching techniques, problem-solving strategies, graph algorithms, and advanced topics like string matching, FFT-DFT, and approximation algorithms. By the end of the course, students will have developed the skills to critically analyze algorithm efficiency and apply advanced algorithms in practical scenarios.
Course Description	This course will provide basic understanding of methods to solve problems on computers. It will also provide an overview to analyze those theoretically.
Course Outline	Data structures: linked list, stack, queue, tree, balanced tree, graph; Complexity analysis: Big O, omega, theta notation, solving recurrence relation, master theorem Sorting and searching: Quick sort, merge sort, heap sort; Sorting in linear time; Ordered statistics; Problem solving strategies: recursion, dynamic programming, branch and bound, backtracking, greedy, divide conquer, Graph algorithms: BFS, DFS, Shortest path, MST, Network flow; NP-completeness Advanced topics: string matching, FFT-DFT, basics of approximation and randomized algorithms. Lab Component: Implementation of above topics
Learning Outcome	By the end of this course, students will be able to: Use linked lists, stacks, queues, trees, balanced trees, and graphs. Analyze algorithm complexity and solve recurrence relations. Implement Quick sort, Merge sort, Heap sort, and linear time sorting methods. Apply recursion, dynamic programming, branch and bound, backtracking, greedy, and divide-and-conquer methods. Implement BFS, DFS, shortest path algorithms, MST, and network flow algorithms. Comprehend NP-completeness and its significance.
Assessment Method	Quiz / Assignment / ESE

Suggested Reading:

- Mark Allen Weiss, "Data Structures and Algorithms in C++", Addison Wesley, 2003.
- Adam Drozdek, "Data Structures and Algorithms in C++", Brooks and Cole, 2001.
- Aho, Hopcroft and Ullmann, "Data structures and Algorithm", Addison Welsey, 1984.

Introduction to Algorithms Book by Charles E. Leiserson, Clifford Stein, Ronald Rivest, and Thomas H. Cormen

Course Number	**ECS 5102**
Course Credit	L-T-P-C: 3-0-2-4
Course Title	**Foundations of Computer Systems**
Learning Mode	Online
Learning Objective	The objective of the course is to provide a conceptual and theoretical understanding of computer architecture and operating systems.
Course Description	Foundations of computer systems is a review of two fundamental subjects of computer science viz., computer architecture and operating systems.
Course Outline	**Computer architecture:** Performance measures, Memory Location and Operations, Addressing Modes, Instruction Set, A Simple Machine, Instruction Mnemonics and Syntax, Machine Language Program, Assembly Language Program with examples. Processing Unit Design: Registers, Datapath, CPU instruction cycle, Instructions and Micro-operations in different bus architectures, Interrupt handling, Control Unit Design: Control signals, Hardwired Control unit design, Microprogram Control unit design. Pipelining and parallel processing, Pipeline performance measure, pipeline architecture, pipeline stall (due to instruction dependancy and data dependancy), Methods to reduce pipeline stall. RISC and CISC paradigms, I/O Transfer techniques, Memory organization: hierarchical memory systems, cache memories, virtual memory. **Operating systems:** Process states, PCB, Fork, exec system call, Threads, Process scheduling, Concurrent processes, Monitors, Process Synchronization, Producer Consumer Problem, Critical section, semaphore, Various process synchronization problems. Deadlock, Resource Allocation Graph, Deadlock prevention, Deadlock Avoidance: Banker's Algorithm and Safety Algorithm. Memory management techniques, Allocation techniques, Paging, Page Replacement Algorithms, Numericals. Lab Component: Implementation of above topics
Learning Outcome	This course will revisit two fundamental subjects of computer science viz., computer architecture and operating systems, thereby enabling the students to pursue more advanced problems in computer science based on these topics.
Assessment Method	Quiz / Assignment / ESE

Suggested readings:

1. A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, 7^{th} Ed, John Wiley and Sons, 2004.
2. M. Singhal and N. Shivratri, Advanced Concepts in Operating Systems, McGraw Hill, 1994.
3. David A Patterson and John L Hennessy, Computer Organisation and Design: The Hardware/Software Interface, Morgan Kaufmann, 1994. ISBN 1-55860-281-X.

Course Number	**EMC 5103**
Course Credit (L-T-P-C)	L-T-P-C: 3-0-2-4
Course Title	**Probability and Statistics**
Learning Mode	Online
Learning Objective	To understand the basic concepts in Probability Theory and Statistics through practical examples.
Course Description	The course is divided into two parts: In first part, basic concepts of probability theory are introduced. In the second part, different problems in classical statistics are discussed.
Course Outline	Conditional probability, Bayes' rule, Total probability law, Independence of events. Random variables (discrete and continuous), probability mass functions, probability density functions, Expectation, variance, moments, cumulative distribution functions, Function of random variables, Multiple random variables, joint and marginal, conditioning and independence, Markov and Chebyshev inequalities, Different notions of convergence. Weak law of large number, Central limit theorem. Estimation: Properties, Unbiased Estimator, Minimum Variance Unbiased Estimator, Rao-Cramer Inequality and its attainment, Maximum Likelihood Estimator and its invariance property, Efficiency, Mean Square Error. Confidence Interval: Coverage Probability, Confidence level, Sample size determination. Testing of Hypotheses: Null and Alternative Hypotheses, Test Statistic, Error Probabilities, Power Function, Level of Significance, Neyman-Pearson Lemma.
Learning Outcome	Students will become familiar with principal concepts probability theory and statistics. This helps them to handle, mathematically, various practical problems arising in uncertain situations.
Assessment Method	Quiz / Assignment / ESE

Text Books:

1. Ross, S.M.(2008) Introduction to Probability Models, Ninth edition, Academic Press.
2. Statistical Inference (2007), G. Casella and R.L. Berger, Duxbury Advanced Series.

Reference Book:

1. An Introduction to Probability and Statistics, V.K. Rohatgi and A.K.Md. Ehsanes Saleh, John Wiley, 2nd Ed, 2009.

Course Number	ECS 5201
Course Credit	L-T-P-C: 3-0-2-4
Course Title	**Artificial Intelligence**
Learning Mode	Online
Learning Objectives	● To understand the foundational concepts and motivations behinrtificial Intelligence and intelligent agents. ● To learn and apply uninformed and informed search strategies for problem-solving. ● To explore local search techniques and optimization methods beyond classical search. ● To implement adversarial search techniques and problem reduction strategies. ● To formulate and solve Constraint Satisfaction Problems (CSPs) using advanced techniques.
Course Description	This course aims to provide students with a comprehensive understanding of the fundamental principles and techniques of Artificial Intelligence (AI). It covers the basics of intelligent agents and their environments, various problem-solving methods through search strategies, and techniques beyond classical search. Students will learn about adversarial search, constraint satisfaction problems, knowledge representation, reasoning, planning, and various learning techniques. The course prepares students to design and implement AI solutions for complex real-world problems.
Course Outline	● Introduction and motivation Artificial Intelligence, intelligent agents, nature of environments ● Problem-solving by searching: Example problems, uninformed, informed search strategies ● Uninformed/ blind search techniques: Breadth-first search (BFS), Depth-first search (DFS), Uniform-cost search (UCS) ● Informed search: Heuristic function design and evaluation, A* search ● Beyond classical search: local search techniques and optimization, hill climbing, simulated annealing, beam search ● Adversarial search: Games, optimal decision in games, min-max, alpha-beta pruning, partially observable games ● Problem reduction techniques: And-OR (AO) and AO* ● Constraint Satisfaction Problem (CSP): definition and examples of CSPs, basic techniques: backtracking search, forward checking, arc consistency ● Knowledge Representation, Reasoning, and Planning: Propositional logic, first-order logic, inference, planning ● Learning Techniques: meta-heuristic (genetic algorithm), Bayesian, decision tree, etc.

	● Some advanced techniques of AI and its applications ● Lab component: Implementation of above architectures.
Learning Outcome	By the end of this course, students will be able to: ● Understand the foundational concepts and motivations behind Artificial Intelligence and intelligent agents. ● Apply uninformed and informed search strategies to solve example problems. ● Utilize local search techniques and optimization methods such as hill climbing, simulated annealing, and beam search. ● Implement adversarial search techniques including min-max, alpha-beta pruning, and strategies for partially observable games. Apply problem reduction techniques. ● Formulate and solve Constraint Satisfaction Problems (CSPs) using techniques like backtracking search, forward checking, and arc consistency. ● Represent knowledge using propositional and first-order logic, and perform inference and planning. ● Explore and apply various learning techniques such as genetic algorithms, Bayesian methods, and decision trees.
Assessment Method	Quiz / Assignment / ESE

Suggested Reading

1. Russell, S. J., & Norvig, P. (2016). Artificial intelligence: A modern approach. Pearson.
2. Poole, D. L., & Mackworth, A. K. (2010). Artificial Intelligence: foundations of computational agents. Cambridge University Press.
3. Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: Springer.

Course Number	EMC 5202
Course Credit	L-T-P-C: 3-0-2-4
Course Title	**Numerical Linear Algebra and Optimization Techniques**
Learning Mode	Online
Learning Objectives	The objective of the course is to train students about the different numerical techniques to solve linear equations, linear least square problems and find eigen values of matrices as well as check the stability of numerical methods. Moreover, students would be able to perform modeling of convex programming problems and employ various classical and numerical optimization techniques and algorithms to solve these problems
Course Description	Numerical Linear Algebra and Optimization Techniques, as a basic subject for postgraduate students, provides the knowledge of various numerical techniques to solve linear equations as well as check the stability of numerical methods. Moreover, this course would help the students to models convex optimization problems and learn different algorithms to solve such problems with its applications in various problems arising in economics, science and engineering.
Course Content	Review of matrix Algebra, Norms and condition numbers of Matrix, Systems of Equations, Gaussian Elimination, LU, PLU and Cholesky Factorization, Iterative Solvers: Jacobi, Gauss Seidel, SOR and their convergence, Gram-Schmidt orthogonalization QR Factorization and Least Squares, Eigenvalues, Singular Value Decomposition and PCA. Introduction to nonlinear programming, Convex Sets, Convex Functions and their properties. Unconstrained optimization of functions of several variables: Classical techniques. Numerical methods for unconstrained optimization: One Dimensional Search Methods, Golden Section Search and Fibonacci search, Basic descent methods, Conjugate direction, Newton's and Quasi-Newton methods
Learning Outcome	On successful completion of the course, students should be able to: 1. Understand different Matrix factorization method and employ them to solve linear equations and linear least square problems 2. To comprehend the basic computer arithmetic and the concepts of conditioning and stability of a numerical method. 3. Understand the terminology and basic concepts of various kinds of convex optimization problems and solve different solution methods to solve convex Programing problem.
Assessment Method	Quiz / Assignment /ESE

Text Books:

1. Lloyd N. Trefethen, David Bau III: Numerical Linear Algebra, 1st Edition, SIAM, Philadelphia (1997)
2. Edwin K. P. Chong, Stanislaw H. Zak: An Introduction to Optimization, 4th Edition, Wiley India (2017)
3. Gilbert Strang: Lecture Notes for Linear Algebra, Wellesley Cambridge Press, SIAM (2021)

Reference Books:

1. Stephan Boyd and Lieven. Vandenberghe: Convex Optimization, Cambridge University Press (2004)

Course Number	ECS 5203
Course Credit	L-T-P-C: 3-1-0-4
Course Title	**Theoretical Computer Science**
Learning Mode	Online
Learning Objectives	1. Explore advanced topics in theoretical computer science, including computational complexity, automata theory, and algorithms. 2. Develop rigorous mathematical reasoning and problem-solving skills applicable to theoretical computer science. 3. Understand foundational concepts such as Turing machines, formal languages, and computational models. 4. Engage in independent research and scholarly exploration of theoretical computer science topics. 5. Apply theoretical insights to practical problems in computer science and related fields.
Course Description	This course offers an advanced study of theoretical computer science, focusing on formal models of computation, computational complexity theory, and algorithmic analysis. Students will delve into abstract concepts and mathematical techniques essential for understanding the limits and capabilities of computing systems. Topics covered include formal languages, automata theory, computability theory, complexity classes, and advanced algorithms. Through lectures, seminars, and research projects, students will develop a deep understanding of theoretical frameworks and their implications for solving real-world computational problems.

Course Outline	1. Introduction to Theoretical Computer Science , Historical overview and scope of theoretical computer science , Mathematical foundations in computer science , Overview of formal models of computation 2. Automata Theory , Finite automata and regular languages , Context-free grammars and pushdown automata , Turing machines and computability 3. Formal Languages and Computability , Formal language definitions and properties , Decidability and undecidability , Church-Turing thesis and implications 4. Computational Complexity , Time and space complexity classes , NP-completeness and beyond , Complexity hierarchies and reductions 5. Advanced Topics in Algorithms , Design and analysis of algorithms , Approximation algorithms and randomized algorithms , Advanced data structures 6. Logic in Computer Science , Propositional and first-order logic , Model theory and logical reasoning , Applications of logic in computing 7. Theory of Computation , Formal models beyond Turing machines , Quantum computing and computational models , Complexity aspects of quantum computation 8. Cryptography and Computational Complexity , Basics of cryptography and cryptanalysis , Complexity-theoretic foundations of cryptography , Applications of cryptography in secure computation
Learning Outcome	1. Demonstrate advanced knowledge of theoretical computer science principles and methodologies. 2. Apply mathematical reasoning and formal methods to analyze computational problems. 3. Evaluate the computational complexity of algorithms and problems using theoretical frameworks. 4. Conduct independent research in theoretical computer science and contribute to scholarly discourse. 5. Translate theoretical insights into practical solutions for complex computational challenges.
Assessment Method	Quiz / Assignment / ESE

Suggested Reading

1. "Introduction to the Theory of Computation" (3rd Edition) by Michael Sipser
2. "Computational Complexity: A Modern Approach" (1st Edition) by Sanjeev Arora and Boaz Barak
3. "Automata Theory, Languages, and Computation" (3rd Edition) by John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman
4. "Algorithm Design" (1st Edition) by Jon Kleinberg and Éva Tardos
5. "Introduction to Algorithms" (3rd Edition) by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein