AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY
School of Engineering
Amritanagar P.O., Ettimadai,
Coimbatore - 641 112 Tamil Nadu, India Ph: +91 422 2685000
Fax: +91 422 2686274 Email: ase@amrita.edu

M.TECH COMPUTER SCIENCE AND ENGINEERING

CURRICULUM & SYLLABUS 2021

M. TECH – COMPUTER SCIENCE AND ENGINEERING
Department of Computer Science and Engineering

The M.Tech. Computer Science and Engineering program is offered at Amrita Vishwa Vidyapeetham by the Department of Computer Science and Engineering in the Amrita School of Engineering.

The field of Computer Science and Engineering is a constantly evolving one and drives the technological trends in today's world. The foundations of these technological trends are rooted in the core concepts and principles of the field of Computer Science. This master's programme is designed to produce graduates who can apply fundamental knowledge of mathematics, programming, problems solving and computing systems to model and solve problems in the real-world and provide a strong foundation to be able to adapt to emerging technological trends that are increasingly end-to-end systems driven combining both software and hardware.

With a view towards developing skilled and adaptable graduate students in Computer Science and Engineering the program curriculum has been framed to incorporate and deliver on foundational aspects of mathematics and computer science, programming and problem solving, system and network design and upcoming key technologies and tools for solving real-world problems. The courses include core courses in Mathematics and Computer Science and a bouquet of soft core courses spanning the core foundations of programming and problem solving, software engineering, computing systems and key technological trends like Cloud Computing, IoT, AI, and Data Science. There are also a wide range of electives in areas spanning Internet of Things, Fog, Edge and Cloud Computing, Cyber Security, Web Science, Mobile Computing, Full Stack Engineering etc. At the end of the course the student would have developed strong foundational skills and strength in selected key technologies so that they can take up advanced research as part of the thesis component and become professionals in this area.

The degree is suitable for students with a bachelor's degree in a computing related field as well as students who want to demonstrate computer science expertise in addition to a degree in another field. The curriculum has been designed to prepare students for a broad range of rewarding careers like: Software Developer, Software Architect, Full Stack Developer, Technical Architect, AI Engineer, Big Data Engineer, Application analyst, Data Scientist, Computer Network Architect, Research Analyst, Senior Research Engineer etc. As a part of the programme during the period of study, students have the opportunity to intern at leading companies and R&D labs for a period of upto to one year. There are opportunities for the students to take up a semester or one year study at International Universities like Vrije University, Netherlands, UC Davis, UNM for an exchange programme or to pursue a dual degree programme.

Program Educational Objectives (PEO)

1. Demonstrate application and adaptation of core concepts in Computer Science and Engineering in industry or research and become prolific professionals and entrepreneurs.
2. Pursue lifelong learning to adapt to emerging computing trends and design computing solutions for real world inter-disciplinary problems.
3. Demonstrate high regard for professionalism, team-spirit, integrity and respect diversity, societal needs and sustainability when designing technological solutions.

Program Outcomes (PO)

At the end of the M.Tech CSE programme the students will be

1. Able to demonstrate a mastery over the foundations of Computer Science and Engineering specifically with respect to solving problems, designing algorithms and systems, and upcoming key technologies
2. Able to design and develop computing solutions using emerging computing paradigms to interdisciplinary problems following standard practices, tools and technologies
3. Able to demonstrate independent study and life-long learning inorder to adapt to the changing landscape of technology and computing trends
4. Able to independently carry out research investigation and development work to solve practical problems
5. Able to write and present a substantial technical report/document
6. Able to demonstrate commitment to professional ethics

Ideas behind the design of POs

1. Program Outcomes PO1, PO4 and PO5 have been adopted from NBA (Ref. https://www.nbaind.org/files/PG_Eng_Annexure/PG_Engineering_Manual.pdf **Page 16**). PO1 has been elaborated to suit our M.Tech CSE.
2. PO1 is about foundational knowledge and skill (and has relevance to all courses). PO2 is about state-of-the-art and current. PO3 is futuristic. PO2 and PO3 very well relates to case studies, lab implementations and dissertation.
3. PO4 and PO5 (adopted from NBA) relates to dissertation as well as to case studies.
4. PO6 explicitly mentions about professional ethics. Ethics and integrity have been part of our mission statements and PEO. PO6 aligns with those explicitly. There are various facets of ethics and integrity which we can map to our courses.

Proposed New Curriculum and New Courses

M.Tech Computer Science & Engineering

Amrita Vishwa Vidyapeetham, Coimbatore

CURRICULUM

I Semester

Course Code	Type	Subject	L T P	Credits
21CS601	FC	Advanced Data Structures and Algorithms	3-0-2	4
21MA609	FC	Mathematical Foundations for Computer Science & Engineering	3-1-0	4
	SC	Soft Core 1	3-0-2	4
	SC	Soft Core 2	3-0-2	4
	SC	Soft Core 3	3-0-2	4
21HU602	HU	Career Competency I		P/F
21HU601	HU	Amrita Values Program		P/F
			Credits	**20**

II Semester

Course Code	Type	Subject	L T P	Credits
	SC	Soft Core 4	3-0-2	4
	SC	Soft Core 5	3-0-2	4
	E	Elective 1	2-0-2	3
	E	Elective 2/Open Elective	3-0-0	3
	SC	Technology Elective	2-0-0	2
21RM605	SC	Research Methodology	2-0-0	2
21HU603	HU	Career Competency II		P/F
			Credits	**18**

III Semester

Course Code	Type	Subject	L T P	Credits
	E	Elective 3	2-0-2	3
	E	Elective 4/ Open Elective	3-0-0	3
21CS798	P	Minor Project/ Capstone Project		10
			Credits	**16**

IV Semester

Course Code	Type	Subject	L T P	Credits
21CS799	P	Major Project		16
			Credits	**16**
			Total Credits	**70**

List of Courses

FOUNDATION CORE

Course Code	Subject	L T P	Credits
21CS601	Advanced Data Structures and Algorithms	3-0-2	4
21MA609	Mathematical Foundations for Computer Science & Engineering	3-1-0	4

SOFT CORE

Course Code	Subject	L T P	Credits
21CS631	Digital Signal Processing	3-0-2	4
21MA603	Graph Analytics and Algorithms	3-0-2	4
21CS632	Programming Paradigms	3-0-2	4
21CS633	Software Engineering with Agile and DevOps	3-0-2	4
21CS634	Advanced Compiler Design	3-0-2	4
21CS635	Information Service Engineering	3-0-2	4
21CS636	Distributed Systems	3-0-2	4
21CS637	Advanced Networks	3-0-2	4
21CS638	Foundations of Cyber-Security	3-0-2	4
21CS639	Full Stack Development	3-0-2	4
21CS640	Modern Database Management Systems	3-0-2	4
21CS641	Modern Computer Architecture	3-0-2	4
21CS642	Advanced Operating Systems	3-0-2	4
21CS643	Foundations of Data Science	3-0-2	4
21CS644	Machine Learning	3-0-2	4

ELECTIVES

Course Code	Subject	L T P	Credits
21CS701	Concurrent Programming	2-0-2	3
21CS702	GPU Architecture and Programming	2-0-2	3
21CS703	Quantum Computing	2-0-2	3
21CS704	Design Patterns	3-0-0	3
21CS705	Mobile Application Development	2-0-2	3
21CS706	Software Quality Assurance and Testing	2-0-2	3
21CS707	Web Science	2-0-2	3
21CS708	Cyber-Physical Systems	3-0-0	3
21CS709	Fog and Edge Computing	3-0-0	3
21CS710	Embedded Programming / Firmware Engineering	2-0-2	3
21CS711	Mobile and Adaptive Networks	3-0-0	3
21CS712	Augmented Reality and Virtual Reality	2-0-2	3
21CS713	Blockchain and Crypto economics	2-0-2	3
21CS714	Cloud Computing and Development	2-0-2	3
21CS715	Cyber Forensics and Malware Analysis	2-0-2	3
21CS716	Deep Learning	2-0-2	3
21CS717	Image and Video Processing	2-0-2	3
21CS718	Internet of Things	2-0-2	3
21CS719	IoT protocols and Architecture	2-0-2	3
21CS720	Location Data Analysis and Modelling	2-0-2	3
21CS721	Offensive Cyber Security	2-0-2	3
21CS722	Privacy Engineering	2-0-2	3
21CS723	Remote Sensing and GIS	2-0-2	3
21CS724	Special Topics in Cyber-Security	3-0-0	3
21CS725	Large Scale Data Processing Using Full Stack (Technology Elective)	2-0-0	2

OPEN ELECTIVES

The students have the option of choosing electives from other M.Tech programmes like Artificial intelligence, Data Science, Cyber-Security etc.

TECHNOLOGY ELECTIVE

This is an industry centric elective aimed at providing insight into the latest technology trends of the day. This will be offered in collaboration with industry experts.

SYLLABUS

FOUNDATION CORE

21CS601 ADVANCED DATA STRUCTURES AND ALGORITHMS 3-0-2-4

PREREQUISITES: Basics of Programming, Data Structures and Algorithms

SYLLABUS

Algorithm Analysis - Methodologies for Analyzing Algorithms, Asymptotic growth rates, Amortized Analysis. Number Theory; Median and Order Statistics; Array based structures, lists and String algorithms. Advanced Data Structures - Dictionaries, hash tables, bloom filters, binary search trees, interval and range trees; skip lists, suffix trees.
Foundations and Applications of Divide-and-Conquer, Greedy techniques, Dynamic Programming, Backtracking and Branch and Bound. Applications of graph algorithms: Topological sort, Strongly Connected Components, Bi-connected Components, Bridges, Articulation points. All Pair Shortest Paths, Single Source Shortest Paths.
Flow Networks: Ford-Fulkerson, Edmonds Karp, Applications of maximum flows - Efficient algorithms for maximum bipartite matching, minimum cost matching. NP-Completeness: Important NP-Complete Problems, Polynomial time reductions, Approximation algorithms, Parallel Algorithms (overview): Tree Contraction - Divide and Conquer - Maximal Independent Set.

SKILLS ACQUIRED: Introduces students to advanced data structures and applications and techniques for the design and analysis of algorithms and explores a variety of applications.

TEXT BOOKS/REFERENCES:

1. Goodrich M T, Tamassia R and Michael H. Goldwasser, "Data Structures and Algorithms in Python++", Wiley publication, 2013.
2. Cormen T H, Leiserson C E, Rivest R L and Stein C. Introduction to Algorithms, Prentice Hall of India Private Limited, Third Edition; 2009.
3. Michael T Goodrich and Roberto Tamassia, "Algorithm Design and Applications", John Wiley and Sons, 2014
4. Motwani R, Raghavan P. Randomized algorithms. Cambridge university press; 1995.
5. Vijay V. Vazirani. Approximation Algorithm, Springer; 2003.

CO Code	Course outcome statement
CO1	Understand the theoretical foundations of analysis of algorithms and analyze complexity of data structures and algorithms
CO2	Understand and apply advanced data structures in problem solving
CO3	Understand, apply different algorithm design techniques for real-world problems

CO4	Analyze the complexity classes of different problems and map it to classical problems
CO5	Design approximation and parallel algorithms for complex problems

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	1				
CO2	3	2	1	1		1
CO3	3	2	1	1		1
CO4	3	2	1	1		
CO5	3	2	1	1		1

COURSE TYPE: Lab
EVALUATION: 70/30

21MA609 MATHEMATICAL FOUNDATIONS FOR COMPUTER SCIENCE 3-1-0-4

PREREQUISITES: Basics of Linear Algebra

SYLLABUS:

General Vector Spaces – Subspaces, Linear Combination, Span, Linear Independence, Bases and Dimension – Inner Product Spaces-Norm, Angle and Orthogonality, Gram Schmidt Process, Least Square Fitting to Data – Eigenvalues, Eigenvectors, and Diagonalization – LU, QR and SVD decompositions.

Review of Probability Theory , Conditional Probability, Baye's Rule, Random Variables – Expectation and Variance, Binomial, Poisson, Geometric and Uniform Distributions, Exponential, Uniform and Normal Distributions. Two Discrete and Continuous Random Variables – Joint Probability Distributions, Covariance – Central Limit Theorem–Point Estimation of Parameters, Maximum Likelihood Estimation, Confidence Intervals, Test of Hypothesis of single Mean, Variance

SKILLS ACQUIRED:

1) Understand how the given data can be handled and analysed systematically using computers based on the Linear Algebra concepts.
2) Statistically analyse the random data of huge size which is more common in real life.

TEXT BOOKS/ REFERENCES:

1. Howard Anton and Chris Rorrers," Elementary Linear Algebra", Tenth Edition, 2010 John Wiley & Sons, Inc.
2. Gilbert Strang, "Introduction to Linear Algebra", Fourth Edition, Wellelsley- Cambridge Press, 2009.
3. Douglas C. Montgomery and George C. Runger, "Applied Statistics and Probability for Engineers", Third Edition, John Wiley & Sons Inc., 2003.
4. David Forsyth, "Probability and Statistics for Computer Science", Springer international publishing, 2018
5. Ernest Davis, "Linear Algebra and Probability for Computer Science Applications", CRC Press, 2012.
6. Michael T. Heath, "Scientific Computing an Introductory Survey", Second Edition, Siam Publisher, 2018

CO Code	**Course Outcome Statement**
CO1	Understand and Apply the basic concepts of vector spaces, subspaces, linear independence, span, basis and dimension and analyze such properties on the given set.
CO2	Understand and Apply the concept of inner products and apply it to define the notion of length, distance, angle, orthogonality, orthogonal complement, orthogonal projection, orthonormalization and apply these ideas to obtain least square solution.
CO3	Understand the theory of random variable and distributions to analyse the data
CO4	Understand the theory of two random variables and analyse the relationship in data analytics
CO5	Understand the statistical procedure of hypothesis testing and use it to analyse the data

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	3		1		
CO2	2	2		2		
CO3	1	2		2		
CO4	1	2		2		
CO5	1	2		2		

COURSE TYPE: Theory
EVALUATION: 50/50

SOFT CORE

21CS631 DIGITAL SIGNAL PROCESSING 3-0-2-4

SYLLABUS

Two-Dimensional Signals and Systems:Two-Dimensional Signals-2-D Discrete-Space Fourier Transform-Sampling in Two Dimensions:Sampling Theorem—Rectangular Case,
Sampling Theorem—General Regular Case,Sample Rate Change-Two-Dimensional Systems and Z-Transforms:2D Systems,Z-Transforms,Regions of Convergence

2-D Discrete-Space Transforms: Discrete Fourier Series-Discrete Fourier Transform-2-D Discrete Cosine Transform-Subband/Wavelet Transform-Fast Transform Algorithms- Two-Dimensional Filter Design:FIR Filter Design,IIR Filter Design,Subband/Wavelet Filter Design

Image Perception and Sensing: Light and Luminance-Still Image Visual Properties-Time-Variant Human Visual System Properties-Color Spaces- Digital Image Compression: Transformation, Quantization, Entropy Coding, DCT Coder ,SWT Coder,JPEG 2000

SKILLS ACQUIRED: Understand about 2D signals-Type of Transforms-Importance of different filters -2D and 3D Signal Representation-Image compression formats

TEXT BOOKS/REFERENCES:

1. John W. Woods,"Multidimensional Signal, Image, and Video Processing and Coding ", Elsevier, 2nd edition,2011
2. Richard G. Lyons,"Understanding Digital Signal Processing" ,Pearson Education,2010

CO code	**Course Outcome**
CO1	Understand the representation of 2D Signals and its operations
CO2	Understand and apply the transforms for 2D signals
CO3	Analyze the working principles of different transforms and Filters
CO4	Apply the basics of image processing and video processing along with signal processing

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	2			
CO2	2	2	1	2		
CO3	3	2	1	2		
CO4	2	3	2	2		

COURSE TYPE: Lab
EVALUATION: 70/30

21MA603 GRAPH ANALYTICS AND ALGORITHMS 3-0-2-4

PREREQUISITES: Basic Matrix Theory-Matrices, operations on matrices, inverse and determinant of a matrix. Eigen values and Eigen vectors.

SYLLABUS

Review of Graphs: Graphs and Sub graphs, isomorphism, matrices associated with graphs, degrees, walks, connected graphs, shortest path algorithm.Trees: Trees, cut-edges and cut-vertices, spanning trees, minimum spanning trees, DFS, BFS algorithms. Eccentricity, Center of the graph.

Connectivity: Graph connectivity, k-connected graphs and blocks. Euler and Hamilton Graphs: Euler graphs, Euler's theorem. Fleury's algorithm for Eulerian trails. Hamilton cycles, Chinese-postman problem, approximate solutions of traveling salesman problem. Closest neighbour algorithm. Domination, minimal domination, independent and maximal independent set, Matchings, maximal matchings, Coverings and minimal coverings. Job assignment problem and matching algorithms. Applications to graph networks. Colorings: Vertex colorings, greedy algorithm and its consequences. Vertex coloring algorithm. Planar graphs. Euler theorem on planar graphs.

Graph Networks and Centralities: Graph Networks. Network topologies. Degree and distance centralities. Clustering centrality. Closeness centrality. Betweeness centrality and Eigen value centrality with application in page ranking algorithm.
CASE STUDIES: Apply the centralities to various date networks and analysis.

TEXTBOOKS/ REFERENCES

1. J.A. Bondy and U.S.R. Murty, Graph Theory and Applications, Springer, 2008.
2. Mohammed Zuhair Al-Taie, Seifedine Kadry, Python for Graph and Network Analysis, Springer, 2018.
3. Barabasi and Pasfai, Network Science, Cambride University press, 2016.
4. Meghanathan Natarajan, Centrality Metrics for Complext Networks Analysis, IGI publisher, 2018.
5. Frank Harary, Graph Theory, New York Academy of Sciences, 1979.

6. Amy E. Hodler and Mark Needham, "Graph Algorithms: Practical Examples in Apache Spark and Neo4j", O'reilly Publications, 2019

SKILLS ACQUIRED: Graphs, Trees, Eccentricity city and its applications. Dominations and Matchings. Graph Centralities and its applications.

CO Code	Course Outcome Statement
CO1	Understand the concepts of graph theory and the properties of graphs and apply them in graph analytics.
CO2	Understand and analyze the properties of special graphs in the context of graph analytics
CO3	Understand and apply the concept of various graph parameters such as dominations, independent set, matchings and coverings to any data networks.
CO4	Understand and apply the concepts graph colorings and planar graphs for solving real-world problems
CO5	Analyze the impact of graph centralities through case studies.

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1	1	1	1	1	1
CO2	1	2		1	2	2
CO3	1	2	2	1	1	1
CO4	1	1	1	1	1	1
CO5	1	3	1	2	2	2

COURSE TYPE: Lab
EVALUATION: 70/30

21CS632 PROGRAMMING PARADIGMS 3-0-2-4

Overview of different programming paradigms – Imperative, object-oriented, functional, logical and concurrent programming.

Functional Programming – Lambda Calculus, expressions and lists, evaluation, types, type systems, infinite lists, programming with rules, pattern-matching, type inference, higher-order and curried functions, lazy evaluation, polymorphic functions, continuations, abstract data types.

Concurrency and Parallelism – Identifying parallelism, Dataflow parallelism, nested data parallelism, semi-implicit parallelism, explicit concurrency, threads, concurrency abstractions, lazy and parallel evaluation, concurrency design patterns, designing parallel programs

SKILLS ACQUIRED: solid foundations on declarative (functional and parallel programming paradigm, exposure to different implementation strategies and conscious design decisions when solving problems, familiarity with the features and potential of declarative languages etc.

TEXTBOOKS/REFERENCES

1. Richard Bird and Jeremy Gibbons, Algorithm Design with Haskell, Cambridge University Press, 2020.
2. Richard Bird, Thinking Functionally with Haskell, Cambridge Universiy Press, 2014.
3. Max Bramer, Logic Programming with Prolog, Springer, 2013.
4. Ivan Bratko, Prolog Programming for Artificial Intelligence, Pearson Education, 4th Edition, 2011.
5. Simon Marlow, Parallel and Concurrent Programming in Haskell, O'Reilly, 2013.
6. Stefania Lordana Nita and Marias Mihailescu, Practical Concurrent Haskell – With Big Data Applications, Apress, 2017.

CO Code	Course Outcome Statement
CO1	Understand and apply the concepts that form the basis of declarative programming paradigms.
CO2	Formulate abstractions with procedures and data in declarative programming paradigm.
CO3	Identify parallelism, design and write concurrent programs in declarative paradigm
CO4	Formulate, implement and solve a given problem scenario using declarative programming paradigm.

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1	1	1	-	-	-
CO2	1	1	1	-	-	-
CO3	2	2	1	-	-	-
CO4	3	3	1	1	1	-

COURSE TYPE : Lab
EVALUATION : 70-30

21CS633 SOFTWARE ENGINEERING WITH AGILE AND DEVOPS 3-0-2-4

PREREQUISITES: Basic knowledge of Software Engineering

Introduction to Agile: Agile versus traditional method comparisons and process tailoring Software Process Models – overview, Various Agile methodologies - Scrum, XP, Lean, and Kanban, Agile Manifesto, Scrum and artifacts, Agile Requirements - User personas, story mapping, user stories, 3Cs, INVEST, acceptance criteria, sprints, requirements, product backlog and backlog grooming; Tools: Agile tracking tools;

Definition of Done, Definition of Ready; Estimation; Agile forecasting and project Management - Big visible information radiators, velocity, progress tracking, Track Done pattern, project forecasting, Ux Design, Control the Flow, Sprint Planning, Create product roadmap Sprints: Iterations/Sprints Overview. Velocity Determination, Iteration Planning Meeting, Iteration, Planning Guidelines, Development, Testing, Daily Stand-up Meetings, Progress Tracking, Velocity Tracking, Monitoring and Controlling

Scaled agile frameworks: SAFe, Scrum@Scale, Disciplined Agile Testing: Functionality Testing, UI Testing, Performance Testing, Security Testing, Tools - Selenium Agile Testing: Principles of agile testers; The agile testing quadrants, Agile automation, Test automation pyramid. Test automation using UI test tools such as Selenium, Writing unit tests

DevOps: Continuous Integration and Continuous Delivery CI/CD: Jenkins Creating pipelines, Setting up runners Containers and container orchestration (Dockers and Kubernetes) for application development and deployment; Checking build status; Kubernetes, Run a container image within a kubernetes cluster Fully Automated Deployment; Continuous monitoring with Nagios; Introduction to DevOps on Cloud

SKILLS ACQUIRED: Introduces students to industry standard agile practices and Devops and explores a variety of applications.

TEXT BOOKS/REFERENCES:

1. Agile Project Management: Creating Innovative Products, Second Edition By Jim Highsmith, Addison-Wesley Professional, 2009
2. Agile Project Management: Managing for Success, By James A. Crowder, Shelli Friess, Springer 2014
3. Learning Agile: Understanding Scrum, XP, Lean, and Kanban, By Andrew Stellman, Jennifer Greene, 2015, O Reilly
4. DevOps: Continuous Delivery, Integration, and Deployment with DevOps: Dive... By Sricharan Vadapalli, Packt, 2018
5. Agile Testing: A Practical Guide For Testers And Agile Teams, Lisa Crispin, Janet Gregory, Pearson, 2010
6. More Agile Testing: Learning Journeys for the Whole Team By Janet Gregory, Lisa Crispin, Addison Wesley, 2015
7. DevOps: Puppet, Docker, and Kubernetes By Thomas Uphill, John Arundel, Neependra Khare, Hideto Saito, Hui-Chuan Chloe Lee, Ke-Jou Carol Hsu, Packt, 2017

CO Code	**Course outcome statement**
CO1	Compare and contrast the differences between Agile and other project management methodologies
CO2	Interpret and apply various principles, phases and activities of the Scrum methodology
CO3	Apply Agile Testing principles for real life situations and understand the basics of SAFe for scaled agile
CO4	Identify and apply various tools for Agile development and CI/CD
CO5	Implement DevOps principles for CI/CD

CO Code	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**
CO1	1	1				
CO2	1	1			3	
CO3	3	3	1	1		2
CO4	3	3	1		1	2
CO5	3	3			1	2

COURSE TYPE: Project

EVALUATION: 70/30

21CS634 ADVANCED COMPILER DESIGN 3-0-2-4

PREREQUISITES: Nil

SYLLABUS: Overview: phases of the compiler, lexical analysis, Top down parser, Bottom up parser, semantic analysis and symbol table, intermediate representations – Procedure abstraction: procedure call, namespace structure, standard linkages, communicating between procedures – Scope optimization: introduction to common optimizations, local value and superlocal value numbering, Tree-height balancing, loop unrolling, global code placement, inline substitution, procedure placement – Data flow analysis: Iterative data-flow analysis, SSA, interprocedural analysis – Control flow analysis: graph traversal, dominators, Interval analysis, structural analysis – Scalar optimization: dead code elimination, code hoisting, tail-call and leaf-call optimization, redundancy elimination – Register allocation: webs, inference graphs, graph colouring, spilling, live range splitting, Briggs-style register coalescing – Instruction selection: tree-pattern matching.
CASE STUDY: classic Fortran-H and PL8 compiler/ Swift Java Compiler from HP Labs / Small-talk-80 compiler

SKILLS ACQUIRED: Develop a fundamental understanding of various approaches for analysis and optimization of programs, understanding the industry architecture of the compilers, construct an industry standard compiler using the Low-Level Virtual Machines (LLVM).

TEXT BOOKS/REFERENCES:

1. Keith Cooper and Linda Torczon, Engineering a Compiler, 2nd Edition, Morgan Kaufmann, 2011.
2. Steven Muchnick, Advanced Compiler Design & Implementation, Morgan Kaufmann, August 1997.
3. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools, 2nd Edition, Addison-Wesley, 2006.
4. Andrew W Appel, Modern Compiler Implementation in Java (or ML, or C), Cambridge University Press, 2002.

CO Code	Course outcome statement
CO1	Identify the various basic blocks and its control flow and apply algorithms to perform scope level optimizations
CO2	Be able to identify the regions in a program and apply various flow analysis techniques
CO3	Understand the single threaded process view of a program and apply machine dependent and independent optimization
CO4	Understand the classic and state-of-art compilers in the industry and develop an industry standard compiler using compiler generator tools

CO5	Identify the execution trace of a program and apply graph algorithms to perform register allocation for variables
CO6	Apply algorithms to generate machine/target code from an intermediate representation

CO Code	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO** 6
CO1	3	2	1	1		
CO2	3	3	1	1		
CO3	3	2	1	1		
CO4	1	3	3	2	2	2
CO5	3	2	1	1		
CO6	3	2	1	1		

COURSE TYPE: Project
EVALUATION: 70-30

21CS635 INFORMATION SERVICE ENGINEERING 3-0-2-4

PREREQUISITES: Nil

SYLLABUS:

Information and Knowledge, Language and Understanding, NLP Techniques and Challenges. Linguistic Basics - Morphology - Evaluation metrics - Regular Expressions. Finite State Automata - Finite State Transducers. Language Model and N-Grams - Part-of-Speech Tagging. Understanding Data on the Web- Towards a Universal Data Representation - How to Identify and Access Linked Data. Representing Facts with RDF - RDF Turtle Serialization - Model Building with RDFS - Logical Inference with RDF(S) - Linked Data Principles - Web Ontology Language OWL - How to Query RDF with SPARQL - Knowledge Mining and Information Extraction - Semantic Search.

SKILLS ACQUIRED: Knowledge Mining and Information Extraction, Semantic Search, Exploratory Search and Recommendation Systems, Linked data programming

TEXT BOOKS/REFERENCES:

1. Jurafsky, Daniel., Martin, James H. Speech and Language Processing. N.p.: Pearson Education, 2014.

2. S. Hitzler, M. Krötzsch, S. Rudolph, Foundations of Semantic Web Technologies, Chapman / Hall, 2009. - T. Heath, Ch. Bitzer, Linked Data - Evolving the Web into a Global Data Space, Morgan & Claypool, 2001.
3. Development of Linguistic Linked Open Data Resources for Collaborative Data-Intensive Research in the Language Sciences, MIT Press, 2020.
4. Rao, Delip., McMahan, Brian. Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning. China: O'Reilly Media, 2019.
5. R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, 2nd ed., Addison Wesley, 2010.

CO Code	Course outcome statement
CO1	Analyze the methods adopted in the explicit formal representation of information.
CO2	Understand the basics of natural language processing and apply the learnt principles in traditional language symbols and acquire intended information using available tools
CO3	Design and develop Tbox, Abox, Rbox axioms to represent the concepts in the structured form.
CO4	Apply linked data principles using open vocabularies to published structured data.

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2			2	
CO2	3	3		2	1	
CO3	3	3	3	3	3	
CO4	3	3	3	3	3	

COURSE TYPE: Lab
EVALUATION: 70/30

21CS636 DISTRIBUTED SYSTEMS 3-0-2-4

PREREQUISITES: None

Introduction and types of distributed systems – Introduction to P2P systems, Edge Networks, CPS, etc, architecture of DS- overview of processes - A Taxonomy of Distributed Systems, scalable performance, load balancing, and availability. Models of computation - shared memory and message passing system— synchronous and asynchronous systems. Communication in Distributed Systems - Remote Procedure Calls and Message Oriented Communications and implementation, High-level communication and publish-subscribe in Mapreduce

Logical time and event ordering, Global state and snapshot algorithms, distributed snapshots in VMs, clock synchronization, Distributed mutual exclusion, Group based Mutual Exclusion, leader election, deadlock detection, termination detection, Distributed Databases, implementations over a simple distributed system and case studies of distributed databases and systems -Distributed file systems: scalable performance, load balancing, and availability. Examples from Dropbox, Google FS (GFS)/ Hadoop Distributed FS (HDFS), Bigtable/HBase MapReduce, RDD

Consistency control: Data Centric Consistency, Client Centric Consistency, Replica Management, Consistency Protocols. Fault tolerance and recovery: basic concepts, fault models, agreement problems and its applications, commit protocols, voting protocols, check pointing and recovery. Case Studies from Apache Spark, Edge Networks, Cyber-Physical Systems, Google Spanner, Amazon Aurora, BlockChain Systems etc.

SKILLS ACQUIRED: Understand design of distributed systems and algorithms that support distributed computing. Gain a practical exposure into the design and functioning of existing distributed systems and algorithms.

TEXT BOOKS/REFERENCES:

1. Andrew S. Tannenbaum and Maarten van Steen, Distributed Systems: Principles and Paradigms, Third Edition, Prentice Hall, 2017.
2. Ajay D. Kshemkalyani and Mukesh Singhal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press, 2011.
3. Garg VK. Elements of distributed computing. John Wiley & Sons; 2002.
4. George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair, Distributed Systems: Concepts and Design, Fifth Edition, Pearson Education, 2017.
5. Fokkink W. Distributed algorithms: an intuitive approach. Second Edition, MIT Press; 2018.

CO Code	Course outcome statement
CO1	Understand the design principles in distributed systems and the architectures for distributed systems.
CO2	Apply various distributed algorithms related to clock synchronization, concurrency control, deadlock detection, load balancing, voting etc.
CO3	Analyze fault tolerance and recovery in distributed systems and algorithms for the same.

CO4	Analyze the design and functioning of existing distributed systems and file systems.
CO5	Implement different distributed algorithms over current distributed platforms

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	1			
CO2	3	3	1	1		
CO3	3	2	1	1	1	
CO4	3	2	1	1	1	
CO5	3	3	2	2		2

COURSE TYPE: Lab and Case Study
EVALUATION: 70/30

21CS637 ADVANCED NETWORKS 3-0-2-4

PREREQUISITES: Computer Networks

SYLLABUS

Internetworking: Architectural principle, Layering, Names and addresses. TCP/IP suite of protocols, TCP extensions for high-speed networks, SCTP, RTSP, RTCP, RTP - End-to-End data, multimedia networking applications, IPv6: API for IPv6, MAC protocols for high Speed LANS, MANs, WLANs. QoS in IP Network, traffic engineering and analysis, SNMP and access control,

Recent trends in Wireless communication networks, wireless sensor networks, multimedia sensor networks, mobile ad-hoc networks, mobility based protocols, next generation IP networks, WIN – Wireless intelligent networking, information aggregation, information storage and query, localization services.

SDN: Data and Control Plane, Open flow Control, Network Function Virtualization for IoT. CASE STUDY: Network management tools used at ICTS, AVVP, Coimbatore Campus, Network protocols along with network security for any one Industrial Use Case.

Skills Acquired: Deep understanding of TCP / IP and L2 protocols in high speed networks, Proper usage of various enabling technologies and protocols related with wireless and mobile networks for

practical applications, Application and usage of various network services in wireless intelligent networks as well as adoption of SDN and NFV for IoT applications.

TEXT BOOKS/REFERENCES:

1. James F. Kurose & Keith W. Ross, Computer Networking: A Top-Down Approach, 7/E. Pearson Education India, 2017.
2. Douglas E Comer, Computer networks and Internets, 6th Edition, Pearson Education, 2015
3. Goransson, P., Black, C., & Culver, T. Software defined networks: a comprehensive approach. Morgan Kaufmann,2016.
4. Online Resources: Technical papers in course related topics and IEEE Standards documents

CO Code	Course Outcome
CO1	Analyze roles of TCP/IP protocol as well as MAC protocols in high-speed networks
CO2	Apply the various enabling technologies and protocols related with wireless and mobile networks for practical applications.
CO3	Understand and apply various network services in Wireless intelligent networks
CO4	Understand the design principles in SDN and NFV for IoT and apply for practical use case

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2			
CO2	2	2	2			
CO3	2	2	2	2		
CO4	2	2	2	2		

COURSE TYPE: Lab
EVALUATION: 70 / 30

21CS638 FOUNDATIONS OF CYBER- SECURITY 3-0-2-4

SYLLABUS

Security Concepts: Confidentiality, Integrity, and Availability, Cryptography, Confusion vs Diffusion, Stream vs Block ciphers, Secret-Key vs Public Key Cryptography, Cryptanalysis. Feistal Networks and Non Feistal Networks, Key Exchange, Public Key Encryption systems. Message digest, Signing and verification, X509 certificates, SSL, HTTPS, Open SSL.
Authentication and Authorization: Factors, Multi-factor, Kerberos, Role based Access Control.
Security in Windows and Linux: Protection system, Authorization, Security Analysis and vulnerabilities, OS issues related to Internet, Intranet, Pervasive Computing, Embedded System Mobile System and Wireless Networks.
Database Security: Securing different forms of data, Database security, attacks on data, SQL Injection, Buffer-flow, Privacy of data, Security issues in big data. Implementing Cryptographic schemes on datasets. Tools: Snort, NPM, Open VAS, Nessus.

SKILLS ACQUIRED

Knowledge in encryption and decryption using private and public key cryptography, digital signatures, cryptographic hash functions. Understand and apply authentication and authorization mechanisms to protect OS and database. Understand the threats and vulnerabilities and need for securing resources and exercising privacy preservation.

TEXTBOOKS/REFERENCES

1. Padmanabhan TR, Shyamala C K, and Harini N, “Cryptography and Security”, First Edition, Wiley India Publications, 2011.
2. Matt Bishop, Computer security: Art and Science, Vol. 2, Addison-Wesley, 2012.
3. M. Gertz and S. Jajodia, Handbook of Database Security-Applications and Trends, Springer, 2008.
4. Jose Manuel Ortega, Mastering Python for Networking and Security, Packt Publishing, 2018.

CO Code	Course Outcome
CO1	Understand the fundamentals of cyber security domain
CO2	Analyze and apply authentication and authorization techniques
CO3	Acquire foundational skills for developing expertise in one or more sub-domains of cyber-security and perform security reviews and audits
CO4	Identify insights on how to apply Cyber Security to secure operating systems and data base design

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		2	2		2
CO2	3	2	2	2		
CO3	3	3	2	2		2
CO4	3	2	2	3		

COURSE TYPE: Project
EVALUATION: 70-30

21CS639 FULL STACK DEVELOPMENT 3-0-2-4

PREREQUISITES: Programming and database fundamentals

SYLLABUS

HTML basics – structuring, positioning, alignment, CSS and JS basics, Browser development tools, Bootstrap basics. Basic Backend App serving text/HTML and HTML from templates. Introduction to web development, Git and GitHub, Taxonomy of frameworks.

Introduction to JavaScript - Introduction to Document and Window Object - Objects and Arrays - Functions. MERN stack - Introduction to ReactJS - Templating using JSX - Components, State and Props - Lifecycle of Components - Rendering List and Portals - Error Handling – Routers - Redux and Redux Saga - Immutable.js - Service Side Rendering - Webpack. Node js Overview - Basics and Setup - Modules – Events - Server side javascript - Exploring package.json - Express js. Creating templates using PUG. Introduction to MVC, Flux, Redux.

Introduction to NoSQL databases - MongoDB A Database for the Modern We - CRUD Operations in MongoD - Indexing and Aggregation - Replication and Sharding - Creating Backup for database - Developing Node JS Application with MongoDB. Hosting web application using public web hosting services.

SKILLS ACQUIRED: Able to design, develop and deploy three-tier web applications using front end and backend frameworks.

TEXT BOOKS/ REFERENCES:

1. Laura Lemay, Rafe Colburn, Jennifer Kyrnin, "Mastering HTML, CSS & JavaScript Web Publishing", Paperback, 2016.
2. Jon Duckett, "Web Design with HTML, CSS, JavaScript and jQuery", Paperback, 2014.
3. Wilson, Eddy. MERN Quick Start Guide: Build Web Applications with MongoDB, Express.js, React, and Node. United Kingdom: Packt Publishing, 2018.
4. Mardan, Azat. Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB. United States: Apress, 2015.
5. Elrom, Elad. React and Libraries: Your Complete Guide to the React Ecosystem. United States: Apress, 2021.

CO Code	Course outcome statement
CO1	Use markup and scripting languages to design and validate dynamic webpages
CO2	Customize pages for users need based on responsive web design concepts
CO3	Learn to design appropriate database services based on the requirements
CO4	Design, develop and deploy an end-to-end web application as a term project

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1	1	2	1	1	
CO2	2	2	1	3	2	
CO3	3	2	1	2	1	
CO4	3	2	3	3	1	

COURSE TYPE: Project
EVALUATION: 70/30

21CS640 MODERN DATABASE MANAGEMENT SYSTEMS 3-0-2-4

PREREQUISITES: None

Overview of RDBMS – Storage and File Structures, Indexing and Hashing - Indexing Structures – Single and Multi-level indexes. Query Processing Optimization and Database Tuning: - Algorithms for Query Processing and Optimization- Physical Database Design and Tuning. Intermediate and Advanced SQL - Embedded SQL Dynamic SQL, Functions and Procedural Constructs, Recursive Queries, Advanced SQL Features. Transactions Processing and Concurrency Control - Transaction Concept, Transaction model, Storage Structure, Transaction Atomicity and Durability, Transaction Isolation, Serializability.
Object Relational Data Models – Complex Data Types, Inheritance, Nesting and Unnesting. NoSQL Databases – NoSQL Data Models, Comparisons of various NoSQL Databases. CAP Theorem, Storage Layout, Query models. Key-Value Stores. Document-databases – Apache CouchDB, MongoDB. Column Oriented Databases – Google's Big Table, Cassandra.
Advanced Application Development – Connecting to MongoDB with Python, MongoDB query Language, Updating/Deleting documents in collection, MongoDB query operators. MongoDB and Python patterns – Using Indexes with MongoDB, GeoSpatial Indexing, Upserts in MongoDB. Document database with Web frameworks

SKILLS ACQUIRED: Understand the design, querying, storage management and transaction processing in SQL, NoSQL and Object Relational databases and design databases given real world scenarios.

TEXT BOOKS/REFERENCES:

1. Ramesh Elmasri and Shamkant B Navathe, "Fundamentals of Database Systems", Fifth Edition, Pearson Educaton India, 2008.
2. Silberschatz A, Korth H F and Sudharshan S, "Database System Concepts", Sixth Edition, Tata McGraw-Hill Publishing Company Limited, 2010.
3. Niall O'Higgins, "MongoDB and Python", O'reilly, 2011.
4. Hector Garcia-Molina, Jeff Ullman and Jennifer Widom, "Database Systems: The Complete Book", Pearson, 2011.
5. Raghu Ramakrishnan and Johannas Gehrke, "Database Management Systems", Third Edition, McGraw-Hill, 2003.
6. Andreas Meier, Michael Kaufmann, "SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management", Springer Verlag 2019

7.

CO Code	Course outcome statement
CO1	Understand and analyze the RDBMS and its internal organization
CO2	Apply algorithms for query processing and optimization
CO3	Apply transaction processing and concurrency control techniques for real-world applications
CO4	Understand and apply the design of Object relational and NoSQL databases
CO5	Understand and implement solutions on big data and graph databases

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	1	2			
CO2	3	2	1	1		1
CO3	3	2	1	1		1
CO4	3	2	1	1		1
CO5	3	3	2	1		1

COURSE TYPE; Project

EVALUATION: 70/30

21CS641 MODERN COMPUTER ARCHITECTURE 3-0-2- 4

PREREQUISITES: Foundation of computer systems

SYLLABUS:

Introduction-Fundamentals of computer design- evaluating performance ,Caches and memory hierarchy design-Review of memory hierarchy-Advanced memory hierarchy design concepts.
Instruction level parallelism and its exploitation-Limits on instruction level parallelism. Multiprocessors and Thread-level parallelism-Models of parallel computation, network topologies, consistency models. Case study of pipelined systems:RISC-V Instruction Sets and Efficiency of Pipelining.
Simultaneous Multi-Threading (SMT), Chip Multi-Processors (CMP), General Purpose Graphics Processing -s (GPGPU). VLSI Scaling issues, data speculation, dynamic compilation, communication architectures and near data processing.Case studies of some contemporary advanced architecture for processors of families like Intel, AMD, IBM etc.

SKILLS ACQUIRED:

Understanding of basic architecture, achieving performance enhancement using various techniques, Knowledge of various Memory ,ILP and multicore processors

TEXTBOOKS/REFERENCES

1. Computer Architecture: A Quantitative Approach, 5th Edition, 2011, By John L. Hennessy & David A. Patterson, Morgan Kaufmann, ISBN: 978-0-12-383872-8
2. Computer Organization and Design, the Hardware/Software Interface, David A Patterson & John L. Hennessy, Morgan Kaufmann, 5th Edition.)

CO Code	Course outcome statement
CO1	Understand the different architectures, and apply performance measures and techniques to achieve better programming performance
CO2	Analyze the various parallel processing techniques such as instruction level parallelism, thread level parallelism and process level parallelism and apply techniques to achieve parallelism
CO3	Analyze the memory organization of modern processor and apply various techniques to achieve parallelism
CO4	Understand the parallel architecture like GPU and employ them for improving code performance

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	-	-	-
CO2	2	2	2	1	1	-
CO3	2	2	2	-	-	-
CO4	2	2	1	1	1	-

COURSE TYPE; Lab

EVALUATION: 70/30

21CS642 ADVANCED OPERATING SYSTEMS 3-0-2-4

PREREQUISITES: Nil

SYLLABUS

Concurrent Execution: Threads, event systems, asynchronous/synchronous I/O, Parallelism, Ordering, and Races, Dynamic Data Race Detector for Multi-Threaded Programs, Discussions of synchronization with an emphasis on monitors, On Optimistic Methods for Concurrency Control, Concurrency Control and Recovery, Communication using lightweight remote procedure call (RPC)

Memory Management: virtual memory, NUMA machines, memory allocators – Hoard Scalable Memory Allocator, Memory Resource Management in VMware, Global Memory Management in Cluster machines

Virtualization: Machine virtualization, binary instrumentation, VMware design etc.
File Systems and Disk: File system interfaces, Networked file systems, AFS, The Design and Implementation of a Log-Structured File System, File system extensibility, non-disk file systems, A Case for Redundant Arrays of Inexpensive Disks (RAID), Using Model Checking to Find Serious File System Errors Big Data System

SKILLS ACQUIRED

Gain knowledge in advanced topics of operating system design and implementation including operating system structuring, synchronization, communication and scheduling in parallel and concurrent systems, communication mechanisms in distributed systems, virtualization, file and disk structure management

TEXTBOOK/REFERENCES

1. Mukesh Singhal, Niranjan Shivaratri, Advanced Concepts in Operating Systems: Distributed, Database, and Multiprocessor Operating Systems, McGraw Hill, 2017.
2. Andrew S Tanenbaum , Modern Operating Systems, Pearson, 2021.
3. Mukesh Singhal , Advanced concepts in operating systems, McGraw Hill, 2017.

CO Code	Course Outcome
CO1	Analyze and apply synchronization principles in parallel processing and distributed systems
CO2	Describe and analyze the memory management and its allocation policies in cluster machines
CO3	Understand Virtual Machines and its interaction with a Hypervisor through practical implications
CO4	Evaluate the storage management policies with respect to different storage management technologies

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	2	2		3
CO2	3	2	2	3		
CO3	3	2	2	3		
CO4	3	2	2	3		

COURSE TYPE: Lab
EVALUATION: 70-30

21CS643 FOUNDATIONS OF DATA SCIENCE 3-0-2-4

PREREQUISITES: Basic Probability

SYLLABUS:

Introduction to Data Science, Causality and Experiments, Data Pre-processing - Data cleaning - Data reduction - Data transformation, Visualization and Graphing: Visualizing Categorical Distributions - Visualizing Numerical Distributions - Overlaid Graphs and plots - Summary statistics of exploratory data analysis, Randomness, Probability, Introduction to Statistics, Sampling, Sample Means and Sample Sizes.

Probability distributions and density functions (univariate and multivariate), Error Probabilities; Expectations and moments; Covariance and correlation; Sampling and Empirical distributions; Permutation Testing, Statistical Inference; Central Limit Theorem, Hypothesis testing of means, proportions, variances and correlations - Assessing Models - Decisions and Uncertainty, Comparing Samples - A/B Testing, P-Values, Causality.

Estimation - Resampling and Bootstrap - Confidence Intervals, Properties of Mean - - Variability of mean -Choosing Sample Size, Prediction - Regression - Method of Least Squares - Visual and Numerical Diagnostics,- Inference for true slope - Prediction intervals, Classification - Nearest neighbors - accuracy of a classifier, Updating Predictions - Making Decisions - Bayes Theorem, Graphical Models. Case Studies

SKILLS ACQUIRED:

Statistical foundations of data science; techniques to pre-process raw data; (data wrangling, munging) with Numpy, Pandas and other Python statistical packages; visualization with Matplotlb, Plotly and Bokeh; EDA; statistical inferences; predictions using statistical tests; estimation of statistical parameters, apply machine learning algorithms for prediction/classification and evaluate the degree of certainty.

TEXT BOOK / REFERENCES:

1. Ani Adhikari and John DeNero, ”Computational and Inferential Thinking: The Foundations of Data Science”, e-book.
2. Joel Grus, ”Data Science from Scratch: First Principles with Python”, 2/e, O’Reilly Media, 2019.
3. Peter Bruce, Andrew Bruce and Peter Gedeck, ”Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python”, 2/e, O’Reilly Media, 2020.
4. Allen B. Downey, Think Stats: Probability and Statistics for Programmers”, 2/e, by O’Reilly Media, 2014. 12
5. Cathy O’Neil and Rachel Schutt,”Doing Data Science”, O’Reilly Media, 2013.

CO Code	Course outcome statement
CO1	Understand the statistical foundations of data science.
CO2	Apply preprocessing techniques on raw data so as to enable further analysis.
CO3	Analyse data employing methods in EDA and prepare insightful visualizations to identify patterns.
CO4	Apply associated inferencing techniques based on statistical models/ tests and evaluate the degree of certainty of predictions.
CO5	Apply various machine learning algorithms for prediction/classification, and evaluation of performance.

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3					
CO2	3	1	2	1		2
CO3	3	2	2	1	1	2
CO4	3	1	3	1	2	2
CO5	3	2	2	1	2	2

COURSE TYPE: Lab and Case Study

EVALUATION: 70/30

21CS644 MACHINE LEARNING 3-0-2-4

PREREQUISITE: BASICS OF LINEAR ALGEBRA, PROBABILITY THEORY AND OPTIMIZATION: Vectors, Inner product, Outer product, Inverse of a matrix, Eigenanalysis, Probability distributions – Discrete distributions and Continuous distributions; Independence of events, Conditional probability distribution and Joint probability distribution, Bayes theorem, Unconstrained optimization, Constrained optimization.

SYLLABUS

Introduction to machine learning - different forms of learning- Linear regression - ridge regression, Lasso, Logistic regression, Discriminant Functions and models, Bayesian regression, regression with basic functions.

Classification - Perceptron –Multilayer Perceptron - Feed forward network - Back propagation – Support vector machine - Decision trees - evaluation of classifiers – bias and variance. Gaussian mixture models -- Expectation-Maximization - Naive Bayes classifier - Ensemble Methods - Bagging – Boosting – Introduction to deep learning - Convolutional neural networks - application of classification algorithm.

Clustering - k-means – Hierarchical clustering - DBScan – evaluation of clustering - dimensionality reduction - Principal Component Analysis - Introduction to Reinforcement Learning.

SKILLS ACQUIRED

Students will be able to articulate the concepts of machine learning and its applications to real world problems. Students will be able to choose the appropriate techniques and algorithms for data analysis. Course gives an exposure to design, build and evaluate models for classification and clustering.

TEXTBOOKS/REFERENCES

1. Bishop, Christopher M. Pattern recognition and machine learning. springer, 2006.
2. Alpaydin, Ethem. Introduction to machine learning. MIT press, 2020.
3. Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.
4. Tom Mitchell, "Machine Learning", McGraw-Hill, 1997
5. Duda, Richard O., and Peter E. Hart. Pattern classification. John Wiley & Sons, 2006.
6. Han, Jiawei, Micheline Kamber, and Jian Pei. "Data mining concepts and techniques third edition." The Morgan Kaufmann Series in Data Management Systems 5.4 (2011): 83-124.

CO Code	Course Outcome
CO1	Understand and apply the basic of ML, learning paradigms and concepts of regression
CO2	Design and develop classifier models and evaluate their performance
CO3	Acquire skills to build probabilistic model and deep network models for classification.
CO4	Develop and build clustering models for real world applications
CO5	Understand and apply the concepts of dimensionality reduction and Reinforcement Learning

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2		1	1	1
CO2	3	3	3	2	2	2
CO3	2	3	2	2	2	1
CO4	3	2	2	2	2	2
CO5	2	2		1	1	1

COURSE TYPE: Lab
EVALUATION: 70-30

ELECTIVES

21CS701 CONCURRENT PROGRAMMING 2-0-1-3

PREREQUISITES: Basic programming knowledge

Syllabus: Introduction: Threads, process, applications, properties of concurrent systems, I/O and CPU bottlenecks, CPU and memory architectures supporting concurrency. Threads in Java/Python: state, types, starting, forking, daemonizing – Handling threads in Python. Synchronization between threads in Java/Python: race conditions, critical section, shared resources and data races – Communication between threads in Java/Python: Message passing. Debugging and benchmarking: testing strategies, debugging, benchmarking, profiling - Executors and pools in Java/Python: concurrent futures, future objects, process pool executor, applications.

Skills Acquired: Able to apply the basic concepts and techniques behind concurrent programming, ability to understand the issues in concurrent programming and apply relevant solution.

TEXT BOOKS/REFERENCES:

1. Elliot Forbes, Learning Concurrency in Python, Packt publishing, 2017.
2. Brian Goetz, Java Concurrency in Practice, Addison-Wesley, 2006.

CO Code	Course outcome statement
CO1	Understand the fundamental concepts of concurrent programming
CO2	Understand the behaviour of threads, analyse the requirement of threads in a real-world problem and apply techniques to handle the synchronous working of threads and inter-thread communication.
CO3	Identify the bugs in concurrent programs using debugging tool, analyse the concurrent programs using various testing tools and optimise the performance of the concurrent programs.
CO4	Improve the performance of concurrent program by applying executors at thread and process level.

CO Code	PO1	PO2	PO3	PO4	PO5	PO 6
CO1	3	1	1			
CO2	3	3	1			
CO3	3	3	1			
CO4	3	3	1			

COURSE TYPE: Lab
EVALUATION: 70-30

21CS702 GPU ARCHITECTURE AND PROGRAMMING 2-0-2-3

GPU accelerated processors are being actively used nowadays in general purpose and scientific computing. These massively parallel, off-the shelf devices are used to run compute-intensive and time consuming part of applications. This course introduces the students to the Single Instruction Multiple Thread (SIMT) architecture of modern GPUs and architecture-aware programming frameworks like Compute Unified Device Architecture (CUDA) and OpenCL. While CUDA programming model is a proprietary framework for the students to learn to interface with GPUs, OpenCL allows them to be familiarized with an open, heterogeneous parallel computing model. Modern day applications of GPUs are also introduced to the students through case studies.

PREREQUISITES : Computer Architecture, Programming Fundamentals , Data Structures

SYLLABUS

Introduction to Parallel Programming – Types of Parallelism – SIMD and SIMT – GPU architecture Threads, Blocks and Grids- GPU Memory Organization- CUDA Programming Model- CUDA Memory Model- Multidimensional thread management with CUDA- Basic CUDA Programming Examples -CUDA Streams – Synchronization and Warp Scheduling, Optimization.

Introduction to OpenCL - OpenCL Device Architectures - Basic OpenCL Programming Model – OpenCL Memory Model - Concurrency and Execution Model - Dissecting a CPU/GPU - OpenCL for Heterogeneous Computing - OpenCL Implementation – examples.

Case study: Convolution, Video Processing, Histogram and Mixed Particle Simulation - OpenCL Extensions - OpenCL Profiling and Debugging – WebCL, Applications of GPU Architecture like Gaming, Computer Vision, etc.

SKILLS ACQUIRED

GPU hardware architecture components – grid, blocks and threads, memory components for parallel processing. Writing parallel programs in C, Python, Python parallel programming packages and using CUDA libraries. Write a massively parallel program for a standard application.

TEXT BOOK / REFERENCES

1. Benedict R. Gaster, Lee Howes, David, R. Kaeli, PerhaadMistry and Dana Schaa, ”Heterogeneous Computing with OpenCL”, Elsevier, 2013.
2. Jason Sanders, Edward Kandrot, ”CUDA by Example: An Introduction to General-Purpose GPU Programming”, Addison-Wesley Professional, 2010
3. Shane Cook, ”CUDA Programming: A Developer’s Guide to Parallel Computing with GPUs”, Newnes, 2012

4. AaftabMunshi, Benedict Gaster, Timothy G. Mattson, James Fung and Dan Ginsburg, ”OpenCL Programming Guide”, Addison-Wesley Professional, 2011.
5. Ryoji Tsuchiyama, Takashi Nakamura, TakuroIizuka and Akihiro Asahara, ”The OpenCL Programming Book”, Fixstars Corporation, 2010.
6. Matthew Scarpio, ”OpenCL in Action: How to Accelerate Graphics and Computations”, Manning Publications, 2011.

CO Code	Course Outcome
CO1	Understand the difference between different parallel programming architectures
CO2	Apply GPU aware programming using CUDA
CO3	Design and develop GPU accelerated real-world simulations and applications
CO4	Apply CUDA libraries to design parallel programs

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1	3	3	-	-	-
CO2	3	3	3	-	2	2
CO3	3	3	3	3	3	3
CO4	3	3	3	3	3	3

COURSE TYPE: Lab
EVALUATION: 70-30

21CS703 QUANTUM COMPUTING 2-0-2-3

This course deals with understanding quantum algorithms and computing. The course also covers Quantum physics-based information and probability theory and how computations are done in a quantum computer. Students will get an introduction to the principles of quantum computation and its mathematical framework.

PREREQUISITES: Probability

SYLLABUS

Introduction to quantum physics - -ary Evolution - Quantum Mechanics - Hilbert space - Quantum Time Evolution -Von Neumann Entropy - Measurement – Schrodinger Equation - Heisenbergs uncertainty principle - Randomness - Computation with Qubits -Matrix Representation of Serial and Parallel Operations - Quantum Boolean Circuits -Periodicity - Quantum Fourier Transform - -ary Transforms - Search and Quantum Oracle - Grovers Amplification - Circuit Representation - Speeding up the Traveling Salesman Problem -The Generate-and -Test Method - Quantum Problem - Solving -Heuristic Search - Quantum Tree Search -Tarratacas Quantum Production System.

Problem Solving-Rules-Logic-based operators-Frames-Categorial representation - Binary vector representation-Production System-Deduction systems - Reaction systems - Conflict resolution - Human problem – solving - Information and measurement - Reversible Computation-Reversible circuits - Toffoligate – Gate based Quantum Computer – standard gates and their operations.

A General Model of a Quantum Computer - Cognitive architecture - Representation -Quantum Cognition - Decision making - Unpacking Effects - Quantum walk on a graph -Quantum annealing - Optimization problems - Quantum Neural Computation - Applications on Quantum annealing Computer – Development libraries - Quantum Computer simulation toolkits.

SKILLS ACQUIRED

Components of computing in a Quantum world – mathematical representation of quantum physics and operations. Write computations in the real world (standard) in a Quantum computer and simulator.

TEXT BOOK/REFERENCES

1. Jack D. Hidary, Quantum Computing: An Applied Approach, Firstedition, Springer International Publishing, 2019
2. N. David Mermin, Quantum Computer Science: An Introduction, Firstedition, Cambridge University Press, 2007

CO Code	Course Outcome
CO1	Understand the computation with Qubits
CO2	Apply Quantum algorithms -Fourier Transform and Grovers amplification
CO3	Apply Quantum problem solving using tree search
CO4	Understand and explore the models of Quantum Computer and Quantum Simulation tools

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	2	-	3	2
CO2	2	2	2	-	3	3
CO3	3	3	3	2	3	3
CO4	3	3	2	2	3	3

COURSE TYPE: Lab
EVALUATION: 70-30

21CS704 DESIGN PATTERNS 3-0-0-3

PREREQUISITES: Object oriented programming, UML

SYLLABUS

Overview of Object-oriented concepts, UML, SOLID design principles. Introduction to Design Patterns, Taxonomy of design patterns. Template method pattern, Strategy pattern.

Observer pattern, State pattern, Composite pattern, Decorator pattern, Singleton pattern, Factory method and Abstract factory patterns, Visitor pattern.

Builder and Prototype patterns, Adapter, Bridge, Proxy and Façade patterns, Command pattern, Iterator pattern. Overview of other patterns. Anti-patterns.

SKILLS ACQUIRED: Understand the common software design problems, how to use design patterns to solve these problems, ability to use the right pattern for a given scenario.

TEXT BOOKS/REFERENCES:

1. Erich Freeman, Elisabeth Robson, Bert Bates and Kathy Sierra "Head First Design Patterns", O'Reilly Media Inc., October 2004.
2. Erich Gamma, Richard Helm, Ralph Johnson and John M. Vlissides, "Design Patterns: Elements of Reusable Object Oriented Software", Second Edition, Addison Wesley, 2000
3. James W. Cooper, "Java Design Patterns: A Tutorial", Second Edition, Pearson Education, 2003.
4. Mark Grand, "Patterns in Java – A Catalog of Reusable Patterns Illustrated with UML", Wiley – Dream tech India, 2002.5.

CO Code	Course outcome statement
CO1	Understand the common software design problems seen in the development process
CO2	Demonstrate the use of various design patterns to tackle these common problems
CO3	Identify the most suitable design pattern to address a given software design problem
CO4	Analyze existing code for anti-patterns and refactor the code
CO5	Apply best practices of design principles for software design and development

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	2	2	1	
CO2	3	2	1	3	1	
CO3	1	2	1	2	2	
CO4	3	1	1	1	2	
CO5	2	2	3	2	3	

COURSE TYPE: Theory
EVALUATION: 70/30

21CS705 MOBILE APPLICATION DEVELOPMENT 2-0-2-3

PREREQUISITES:

Object-oriented programming language (Java) , Full Stack Development

SYLLABUS:

Introduction to Mobile Application Development - Mobile Application Environments and Architecture - SDK, API Levels. Set up a mobile app development environment - Developing and debugging mobile app components - First application - understanding file structure - layout and resource files - deployment - emulators and devices.
Basic UI design - Button, EditText, TextView, basic event handlers - Selection components - Radio, checkbox, Date/Time Picker. ListView, Grid view, ScrollView, Image view, Image buttons, Spinner, Toggle, AutocompleteTextView. Advanced UI design - Intents, Managing Permissions - Menu, Notifications, Material Design, Navigation Drawer, WebView.
Data storage and sharing - SQLite, SharedPreferences, Internal/External Storage, Cloud database-Data Sharing. Background Processing - Services.
Sensors - Motion sensors, Environmental, Position sensors. Touch sensors and Gesture detector. Location Based Services - GPS and Google Maps.Apps with Connectivity to External APIs.

SKILLS ACQUIRED: Learn to create and customise rich user interfaces, manage data on databases, handle background processing using services. Use APIs such as Maps, Sensors and GPS to develop ready to use android applications for real-world use cases.

TEXTBOOKS / REFERENCES:

1. Bill Phillips, Chris Stewart, Kristin Marsicano. *Android Programming: The Big Nerd Ranch Guide*, 4th Edition, Big Nerd Ranch Guides, 2019.
2. Android Developer Fundamentals Version2, 2018. Accessible online: https://developer.android.com/courses/fundamentals-training/overview-v2
3. Barry Burd, *Android Application Development All in one for Dummies*, 4th edition, John Wiley & sons, 2015.
4. Payne, Rap. *Beginning App Development with Flutter: Create Cross-Platform Mobile Apps*. Apress, 2019.
5. Napoli, Marco L. *Beginning Flutter: A Hands on Guide to App Development*. John Wiley & Sons, 2019.

CO Code	Course outcome statement
CO1	Design and develop interactive mobile user interfaces
CO2	Store and handle data using local and cloud databases.
CO3	Perform background processing using services and handle sensors.

CO4	Use various APIs to create location based services.

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	3	3	2	1	2
CO2	2	3	2	2	1	1
CO3	2	3	2	2		1
CO4	3	3	3	2		1

COURSE TYPE: Lab
EVALUATION: 70/30

21CS706 SOFTWARE QUALITY ASSURANCE AND TESTING 2-0-2-3

PREREQUISITES: Software Engineering with Agile and Dev-Ops

Software quality - People's Quality Expectations, Frameworks and ISO-9126, McCall's Quality Factors and Criteria – Relationship. Quality Metrics. Quality Characteristics ISO 9000:2000 Software Quality Standard. Maturity models- Test Process Improvement ,Testing Maturity Model.

Quality Assurance - Root Cause Analysis, modeling, technologies, standards and methodologies for defect prevention. Fault Tolerance and Failure Containment - Safety Assurance and Damage Control, Hazard analysis using fault-trees and event-trees. Comparing Quality Assurance Techniques and Activities. QA Monitoring and Measurement, Risk Identification for Quantifiable Quality Improvement. Case Study: FSM-Based Testing of Web-Based Applications.

Quality Revolution, Verification and Validation, Failure, Error, Fault, and Defect, Objectives of Testing, Testing Activities, Test Case Selection White-Box and Black ,test Planning and design, Test Tools and Automation, . Power of Test. Test Team Organization and Management-Test Groups, Software Quality Assurance Group ,System Test Team Hierarchy, Team Building.

System Testing - System Integration Techniques-Incremental, Top Down Bottom Up Sandwich and Big Bang, Software and Hardware Integration, Hardware Design Verification Tests, Hardware and Software Compatibility Matrix Test Plan for System Integration. Built- in Testing. functional testing - Testing a Function in Context. Boundary Value Analysis, Decision Tables. acceptance testing - Selection of Acceptance Criteria, Acceptance Test Plan, Test Execution Test. software reliability - Fault and Failure, Factors Influencing Software, Reliability Models

SKILLS ACQUIRED: Students learn to identify defects in software by applying the concepts of testing. They will learn to assess the quality of a software.

TEXT BOOKS/REFERENCES:

1. Software Testing And Quality Assurance-Theory and Practice, Kshirasagar Nak Priyadarshi Tripathy, John Wiley & Sons Inc,2008
2. Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, Jeff Tian, John Wiley & Sons, Inc., Hoboken, New Jersey. 2005.
3. Software Quality Assurance - From Theory to Implementation, Daniel Galin, Pearson Education Ltd UK, 2004
4. Software Quality Assurance, Milind Limaye, TMH ,New Delhi, 2011

CO Code	Course outcome statement
CO1	Understand the basic concepts of Software Quality and standards
CO2	Apply appropriate defect prevention techniques and software quality assurance metrics.
CO3	Apply techniques of quality assurance for a given application.
CO4	Perform functional and non-functional tests in the life cycle of the software product.
CO5	To build design concepts for system testing and execution

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	3	1	1	1
CO2	3	3	1	1	1	1
CO3	3	3	3	3	3	3
CO4	3	3	1	3	3	3
CO5	3	3	1	2	3	2

COURSE TYPE: Lab

EVALUATION: 70/30

21CS707 **WEB SCIENCE** **2-0-2-3**

How big is the web, A simplistic Descriptive Model, Generative Modelling, Predictive model, Graph Theoretic Web Modelling, Herding Behaviour; An Overview on Social Networking: Design, Issues, Emerging Trends, and Security; Classification and Analysis of Facebook Metrics Dataset Using Supervised Classifiers, Emergence of Stable and Glassy States in Dynamics of Social Networks, De-Anonymization Techniques for Social Networks, An Analysis of Demographic and Behavior Trends Using Social Media: Facebook, Twitter, and Instagram; Social Network Influence on Mode Choice and Carpooling During Special Events; Sentiment Analysis on a popular social networking dataset such as Movie reviews, Arline services; The Interplay of Identity and Social Network: A Methodological and Empirical Study Social Networks and Their Uses in the Field of Education:Public Engagement in the Digital Ecosystem, Investigating Meme Spreading, Introduction to online advertising, User Modelling, Copyright, Net Neutrality.

SKILLS ACQUIRED: Data analytic techniques that can be applied to social networks, Cutting edge research on social network analysis and includes applications to a number of domains, Methods of visualizing, modeling and tracking network patterns, organization, growth and change.

TEXT BOOKS/REFERENCES:

1. Dey, Nilanjan., Borah, Samarjeet., Ashour, Amira., Babo, Rosalina. Social Network Analytics: Computational Research Methods and Techniques. Netherlands: Elsevier Science, 2019.
2. Anderson, Paul. Web 2.0 and Beyond: Principles and Technologies. -ed States: CRC Press, 2016.
3. Mixed Methods Social Network Analysis: Theories and Methodologies in Learning and Education. -ed Kingdom: Taylor & Francis, 2019.
4. Schenker, Adam., Last, Mark., Kandel, Abraham., Bunke, Horst. Graph-theoretic Techniques for Web Content Mining. Singapore: World Scientific, 2005.

CO Code	Course outcome statement
CO1	Analysis of modeling techniques and their usage related to web content
CO2	Design and analyze web applications as socio-technical systems
CO3	Compare techniques needed to evaluate web based social systems.
CO4	Analyse strategic benefits and risks of social networks and understand, how collaborative social networking works
CO5	Understand techniques employed in developing advertising models on the web

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2		1	1	
CO2	3	2	1	2	1	
CO3	2	3	2	3	3	
CO4	3	3	3	3	3	
CO5	1	2	2	3	2	

COURSE TYPE: Lab
EVALUATION: 70/30

21CS708 CYBER-PHYSICAL SYSTEMS 3-0-0-3

PREREQUISITES: Distributed Systems, Basic OS, Networks knowledge

SYLLABUS:

Introduction to Cyber-Physical Systems(CPS): Definition, features. CPS Application Domains: Introduction and Motivation, System Description, Operational Scenarios, Design Drivers and Attributes in Medical CPS, Energy CPS, CPS built on WSNs, Robotics and Autonomous Vehicles.

Symbolic Synthesis for CPS: Introduction and Techniques, Temporal Logic, Symbolic Models. Software and Platform issues in Feedback Control Systems: Basic Techniques for Controller design and timing, Event-Based Control, Controller Software Structures, Sharing of Computing Resources, Analysis and Simulation. Model Integration in CPS: Causality, Semantic Domains for Time, Interaction Models for Computational Processes, Semantics of CPS DSMLs, ForSpec, The Syntax of CyPhyML, Formalization of Semantics, Formalization of Language Integration.

Distributed CPS: Synchronization needs in CPS, Distributed Consensus Algorithms, 3 Synchronous Lockstep Executions, Time-Triggered Architecture, Physically Asynchronous, Logically Synchronous Systems. Realtime Scheduling: Scheduling with Fixed Timing Parameters, Memory Effects, Multiprocessor/Multicore Scheduling, Accommodating Variability and Uncertainty, Managing Other Resources, Rhythmic Tasks Scheduling. Logical Correctness for Hybrid Systems: Introduction and Motivation, Basic Techniques, Discrete Verification, Advanced Techniques, Real-Time Verification, Hybrid Verification. Security of CPS: Introduction and Motivation, Attack Model and Counter Measures, System Theoretic Approaches.

SKILLS ACQUIRED: Introduction to CPS, CPS foundations including the symbolic synthesis and modeling paradigms, engineering problems in CPS and applications from various domains.

TEXTBOOKS/REFERENCES

1. Mark Klein, Dionisio de Niz, Raj Rajkumar, "Cyber-Physical Systems", Addison-Wesley Professional, 2016
2. Rajeev Alur, "Principles of Cyber-Physical Systems", MIT Press, 2015
3. Edward Ashford Lee and Sanjit A. Seshia, "Introduction to Embedded Systems— A Cyber-Physical Systems Approach", Second Edition, MIT Press, 2017

CO Code	Course outcome statement
CO1	Understand the fundamentals of cyber-physical systems and analyze their design in different applications.
CO2	Understand the foundations of modeling in CPSs, software-based feedback control and apply them in the context of sample CPS systems.
CO3	Understand the design of distributed CPS systems with respect to synchronization, real-time scheduling and management and security issues.
CO4	Understand the techniques for formal verification and model integration in CPS and apply them in different domain applications.

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	1		
CO2	3	2	2	1		
CO3	3	2	2	1		
CO4	3	3	2	1		

COURSE TYPE: Theory
EVALUATION: 70/30

21CS709 FOG & EDGE COMPUTING 3-0-0-3

PREREQUISITES: Distributed Systems, Basic OS, Networks knowledge

SYLLABUS:

Introduction and Definition of Edge Networks : edge vs cloud, low power IoT platforms, hardware and software edge platforms, S-IoT resource estimation and challenges. Design of Multilayer Edge Networks: Architectures and methodologies, fog and cloud based architectures, platforms and technologies for configuring and orchestrating edge computing topologies like OpenFog Reference Architecture. Fog protocols. Case Study: Apache Edgent. Data Collection, Analytics, and Decision-making in Edge Computing, tools and technologies for data collection, transmission, and analytics for IoT, statistical and machine learning modeling techniques like regression, tools like IOTA distributed ledger, sensor fusion techniques. Self Learning on the Edge: Federated Learning, Reinforcement learning and online learning on the edge, optimisation for self learning on edge. Self monitoring and healing on edge: Quality of Service monitoring, fault management, and self-healing for edge networks. Edge Computing Applications and Case Studies.

SKILLS ACQUIRED: Understanding the concepts of edge networks and computing, designing edge networks, be able to do analytics, data collection and learning at the edge, improve performance at the edge and analyze the latest edge based systems and platforms and design edge networks for applications.

TEXTBOOKS/REFERENCES

1. Rajkumar Buyya, Satish Narayana Srirama, "Fog and Edge Computing: Principles and Paradigms", Wiley, 2019
2. Javid Taheri, Shuiguang Deng, "Edge Computing: Models, technologies and applications", IET, 2020

CO Code	Course outcome statement
CO1	Understand the foundations of edge networks and different architectures
CO2	Design edge networks for different applications using reference architectures
CO3	Understand and apply data collection, analysis, decision making and learning methodologies over the edge for different applications
CO4	Apply optimization techniques for effective learning, quality of service monitoring and fault management in edge networks for different application

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	1	2			
CO2	2	3	2	2		
CO3	2	2	2			
CO4	2	2	2			

COURSE TYPE: Theory

EVALUATION: 70/30

21CS710 EMBEDDED PROGRAMMING 2-0-2-3

PREREQUISITES: Modern Computer Architecture and Advanced Operating System.

SYLLABUS:

Hardware and Software architecture of Embedded Systems. Review of general C programming and data types, arrays, functions, pointers, structure, enum, files. Introduction to Embedded C, Interfacing C with Assembly. Embedded programming issues - Reentrancy, Portability, Optimizing and testing embedded C programs. Embedded Applications using Data structures, Linear data structures– Stacks and Queues, Linked List. Embedded C++ and Scripting Languages for Embedded Systems. Character Device Driver Development, Driver Concepts, Block v/s character distinction, Writing character drivers, device enumeration and configuration- Software to hardware mapping for specific architecture.
Introduction to real-time systems, RTOS basic architecture, RTOS Kernel, Kernel services: Task Management -tasks, process and threads, task attributes and types - task states and transition, task control block, Introduction to real-time task scheduling. RTOS for multi-core processors. OS for end and edge devices in cyber physical systems. Development, debugging and version control tools for Embedded systems.

Case study: Embedded Linux / VxWorks / Free RTOS / RTLinux / uCOS in critical real-time embedded systems.

SKILLS ACQUIRED: Hands-on experience with various popular Application Processors and Micro Controllers in both hardware and associated software frame work. Practical Exposure to development of device drivers in both bare metal embedded systems as well as real Time Embedded Systems. Good hands –on experience with version control, debugging and testing tools for Embedded systems.

TEXT BOOKS / REFERENCES:

1. Michael Barr, Programming Embedded Systems in C and C++ Oreilly Publications 2012
2. Jim Cooling, Real-time Operating Systems: - The Practice (The engineering of real-time embedded systems) Kindle Edition, 2017
3. David E Simon, "An Embedded Software Primer", Pearson Education Asia, 2005.
4. Cheng, A. M. K., "Real-Time Systems: Scheduling, Analysis, and Verification", John Wiley, 2003.
5. Jean J Labrosse, "MicroC/OS-II-The Real-Time Kernel", 2nd edition, CM

CO Code	**Course outcomes**
CO1	Understand the hardware and software architecture of Embedded Systems
CO2	Develop an embedded application using different hardware and software platforms
CO3	Understand the essentials of real time operating systems and device driver development
CO4	Debug and test embedded hardware and software products

CO Code	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**
CO1	2	2	2			
CO2	2	3	2	2	2	2
CO3	2	2	2			
CO4	2	3	2	2	2	2

COURSE TYPE: Lab
EVALUATION: 70/30

21CS711 MOBILE AND ADAPTIVE NETWORKS 3-0-0-3

PREREQUISITES: Advanced Network

Overview of wireless and mobile systems (Wireless LAN's, Cellular systems and Sensor Networks). Wireless Physical layer – Channel capacity, Modulation technique –OFDM, Chanel coding, MIMO. Wireless Link layer – MAC protocols, CSMA, TDMA, CDMA, link Adaptation, bit rate Adaptation Protocols, error control. Energy consideration in wireless PHY and Link layers.
Network layer: Mobility Management, Mobile IP, Cellular handoffs, Multihop routing protocols, Opportunistic routing. Node clustering ln Adhoc Networks: Mobility-Based Clustering Algorithm, Transport layer mobility, Application layer design for mobility and disconnected operations.
Mobile Computing Platforms, Android Architecture, Sensor Localization Algorithms and crowd sourcing system for Mobile access.

TEXT BOOK

1. Jochen Schiller, Mobile Communications, 2nd Edition.
2. Jun Zheng, and Abbas Jamalipour," Wireless sensor Networks", A Networking Perspective, John Wiley ad sons, 2009

SKILLS ACQUIRED: At the end of the course the students would be able to design algorithms and applications over real time mobile adhoc systems, and study their performance

CO Code	Course outcome statement
CO1	Understand layout and principles of wireless and mobile networks
CO2	Analyze the design challenges involved in physical and data link layer, mobility in transport layer.
CO3	Analyze routing performance and QoS over heterogeneous platforms
CO4	Design mobile and wireless networks for different applications

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	2	2	2	
CO2	3	3	3	3	3	
CO3	3	3	3	3	3	
CO4	3	3	3	3	3	

COURSE TYPE: Theory

EVALUATION: 70:30

21CS712 AUGMENTED REALITY AND VIRTUAL REALITY 2-0-2-3

PRE-REQUISITES: C Programming and Data Structure

SYLLABUS:

Introduction of Virtual Reality: Fundamental Concept and Components of Virtual Reality. Primary Features and Present Development on Virtual Reality. Multiple Models of Input and Output Interface in Virtual Reality: Input -- Tracker, Sensor, Digital Glove, Movement Capture, Video-based Input, 3D Menus & 3DScanner etc. Output -- Visual /Audit ory / Haptic Devices. Visual Computation in Virtual Reality: Fundamentals of Computer Graphics. Software and Hardware Technology on Stereoscopic Display. Advanced Techniques in CG: Management of Large-Scale Environments & Real Time Rendering. Interactive Techniques in Virtual Reality: Body Track, Hand Gesture, 3D Manus, Object Grasp. Development Tools and Frameworks in Virtual Reality: Frameworks of Software Development Tools in VR. X3D Standard; Vega, MultiGen, Virtools etc. Application of VR in Digital Entertainment: VR Technology in Film & TV Production. VR Technology in Physical Exercises and Games. Demonstration of Digital Entertainment by VR. Augmented and Mixed Reality, Taxonomy, technology and features of augmented reality, difference between AR and VR, Challenges with AR, AR systems and functionality, Augmented reality methods, visualization techniques for augmented reality, wireless displays in educational augmented reality applications, mobile projection interfaces, marker-less tracking for augmented reality, enhancing interactivity in AR environments, evaluating AR systems.

TEXTBOOK/ REFERENCES

1. Burdea, G. C. and P. Coffet. Virtual Reality Technology, Second Edition. Wiley-IEEE Press, 2003/2006.
2. Alan B. Craig, Understanding Augmented Reality, Concepts and Applications, Morgan Kaufmann, 2013.
3. Alan Craig, William Sherman and Jeffrey Will, Developing Virtual Reality Applications, Foundations of Effective Design, Morgan Kaufmann, 2009.

SKILLS ACQUIRED: The student acquires knowledge in VR and AR technologies in terms of used devices, building of the virtual environment and modalities of interaction and modelling. Acquire knowledge in the main application of VR and AR technologies in medicine and surgery, cultural heritage and games. This course provides students with an opportunity to explore the research issues in Augmented Reality and Virtual Reality (AR&VR).

CO Code	Course outcome statement
CO1	Understand the fundamentals of Augmented Reality and Virtual Reality, their design in different applications.
CO2	Understand the foundations of modelling in VR and various sensing Gloves, apply them in the context of sample AR systems.
CO3	Analyze the design of VR- specific input & output devices , their principles, capacities and design tradeoffs of the current commercial VR output interfaces.
CO4	Analyze human factor issues, user performance, sensorial conflict aspects of VR

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	1	1	
CO2	3	2	2	1	1	
CO3	3	2	2	1	3	
CO4	3	3	2	1	3	3

COURSE TYPE: Pr**oject**

EVALUATION: 70-30

21CS713 BLOCKCHAIN AND CRYPTO ECONOMICS 2-0-2-3

SYLLABUS

Distributed Computing: P2P systems, Atomic Broadcast, Consensus, Byzantine Models of fault tolerance. Crypto primitives: Hash functions, Puzzle friendly Hash, Collision resistant hash, digital signatures, public key crypto, verifiable random functions, Zero-knowledge systems. Blockchain technology: Blockchain architecture, the challenges, and solutions, proof of work, Proof of stake, alternatives to Bitcoin consensus, Bitcoin scripting language and their use. Smart contracts: Smart contract programming architecture. Programming, deployment and execution. Solidity and Remix. Interacting with smart contracts via the web.

SKILLS ACQUIRED

Knowledge in cryptographic building blocks used in blockchain and crypto currency and reason behind their security. Gain insights on different consensus mechanisms of bitcoins and be able to combine technical methods of cryptography and incentive engineering.

TEXTBOOK/REFERENCES

1. A.Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016
2. Roger Wattenhofer, CreateSpace, The Science of the Blockchain, Independent Publishing Platform,2016
3. Imran Bashir, Mastering Blockchain, 2017.
4. Andreas M. Antonopoulos, Mastering Bitcoin - Programming the Open Blockchain, O'Reilly Media, Inc., 2017
5. Alex Leverington, Ethereum Programming, Packt Publishing Limited, 2017.

CO Code	Course Outcome
CO1	Understand and apply cryptography for securing blockchain based crypto currencies
CO2	Implement the foundations of blockchain that preserves transparency, privacy, anonymity, security, and history
CO3	Explore platforms such as Ethereum to build applications on blockchain technology
CO4	Analyze and identify new use cases for blockchain and/or cryptocurrency

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	2		
CO2	2	2	2	2		
CO3	2	3	2	3		2
CO4	3	3	2	3		

COURSE TYPE: Lab
EVALUATION: 70-30

21CS714 CLOUD COMPUTING AND DEVELOPMENT 2-0-2-3

PREREQUISITES: Distributed Systems, Foundations of Data

SYLLABUS:
Cloud Computing fundamentals - Principles of Cloud Computing Systems, Elastic Cloud Systems for Scalable Computing, Cloud Architectures Compared with Distributed Systems, Service Models, Ecosystems, and Scalability Analysis. Availability, Mobility, and Cluster Optimization; Cloud machine learning engine - cloud MLE train/deploy process, running single instance training and distributed training, hyper parameter tuning, Making predictions on cloud MLE, Batch prediction. Data Collection, Mining, and Analytics on Clouds - Data quality control and representations, Data mining and data analytics on cloud, cloud resources for supporting Big data analytics; Cloud AI services - overview, Natural language Processing - Document Classification, summarisation, sentiment analysis, topic modelling and theme extraction, chatbots. Understanding cloud language translation services, Analysing images with computer vision - Detecting objects and themes in images, image moderation, Facial analysis, text in images. Video Intelligence - Label detection, Operation status. Cloud Speech - synchronous and asynchronous Speech recognition, streaming speech recognition. Cloud dataflow – dataflow templates, data transformation with cloud dataflow. cloud publisher subscriber - architecture, message flow, implementation.

SKILLS ACQUIRED: Cloud store, manage, analyze, and skills required to build intelligent applications; Cloud computing tools and techniques to quickly build prototypes and eventually build applications.

TEXT BOOKS/REFERENCES:

1. Kai Hwang, "Cloud Computing for Machine Learning and Cognitive Applications", The MIT Press, 2017.
2. Ekaba Bisong, "Building Machine Learning and Deep Learning Models on Google Cloud Platform", Apress, 2019.

3. Anand Deshpande, Manish Kumar, Vikram Chaudhari, "Hands-On Artificial Intelligence on Google Cloud Platform", Packt Publishing, 2020
4. Jeffrey Jackovich, Ruze Richards, "Machine Learning with AWS", Packt Publishing, 2017.

CO Code	Course outcome statement
CO1	Understand the basic principles of cloud computing
CO2	Apply cloud machine learning platform to train machine learning models at scale, host trained model in the cloud, and use model to make predictions about new data.
CO3	Apply the cloud big data analysis framework to capture, manage, and process real-time data.
CO4	Apply cloud Artificial Intelligence platform and cloud cognitive services to build, deploy, and manage machine learning models.
CO5	Understand and apply Cloud dataflow models

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2			1		
CO2	3	3	3	2	1	
CO3	3	3			2	
CO4	3		3		1	
CO5	3	3	3	1	3	

COURSE TYPE: Lab
EVALUATION: 70/30

21CS715 CYBER FORENSICS AND MALWARE ANALYSIS 2-0-2-3

PREREQUISITES: Foundations of Cyber Security, Advanced Operating Systems

SYLLABUS:

Operating Systems Fundamentals- Kernel and device driver architecture, Registry, Auditing and security architecture. Introduction to Cyber Forensics - Computer/Network/ Internet Forensics, Anti-forensics; Evidence Collection, Processing. Fundamentals of host forensics for MS-Windows. File system handling - Reconstruction of files and directory structures on the FAT and NTFS, File system and pseudo file systems using UFS and EXT2/3/4 file systems as exemplars. Forensic analysis of database systems, Database tampering, Forensic analysis of database components, Table storage, Transaction logs, indexes, Forensic recovery for table storage

Introduction to malware, Basic Static and Dynamic Analysis, Malware Behavior - malicious activities and techniques, Analyzing Windows programs, disassembler and debugger aided debugging, reverse engineering high level languages.

SKILLS ACQUIRED: Introduction to Cyber Forensics and Malware Analysis, Cyber Forensics foundations including security audit mechanism Reverse engineering of benign and malicious programs.

TEXTBOOKS/REFERENCES:

1. David Cowen, Computer Forensics: A Beginners Guide, Mc Graw Hill Education, 2013.
2. E. Casey, Handbook of Digital Forensics and Investigation, Academic Press, 2010
3. Sikorski M, Honig A. Practical malware analysis: the hands-on guide to dissecting malicious software. no starch press; 2012.
4. Kleymenov A, Thabet A. Mastering Malware Analysis: The complete malware analyst's guide to combating malicious software, APT, cybercrime, and IoT attacks. Packt Publishing Ltd; 2019

CO Code	Course outcome statement
CO1	Understand fundamental forensics techniques, processes and basic malware analysis
CO2	Apply security and audit mechanisms for forensic analysis
CO3	Understand fundamental concepts of reverse engineering and data tampering.
CO4	Apply the static and dynamic malware analysis techniques to dissect malicious codes.

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1	1	1	1	1	1
CO2	2	3	2	2	1	3
CO3	2	1	2	2	1	2
CO4	2	3	3	3	1	2

COURSE TYPE: Project/Case Study
EVALUATION: 70/30

21CS716 DEEP LEARNING 2-0-2-3

PREREQUISITES:

History of Deep Learning – Multi Layer Perceptron - Feedforward Neural Networks - Back propagation –Training - Testing - Parameter Vs Hyperparameter – Optimization algorithms and Regularization – Dropout – Batch Normalization -Vanishing/exploding Gradient – Tensor flow - Convolutional Neural Networks- Autoencoders- Recurrent Neural Networks- Long Short-Term Memory Parameter- Deep Unsupervised Learning – Generative Adversarial Networks -Transfer Learning – Deep Reinforcement Learning - Deep Learning architectures for Computer Vision - Image Segmentation/Object Detection/Image Captioning/Image Generation/Attention Models - Deep Learning architectures for NLP – Named Entity Recognition/Opinion Mining/Sentiment Analysis/Question Answering/Neural Summarization

SKILLS ACQUIRED: Hands-on practice to develop deep neural network models for various applications.

TEXT BOOKS/REFERENCES:

1. Ian Goodfellow, Yoshua Bengio and Aeron Courville, Deep Learning, MIT Press, First Edition, 2016.
2. Gibson and Josh Patterson, Deep Learning A practitioner's approach, Adam O'Reilly, First Edition, 2017.
3. Aurelien Geron, Hands-On Learning with Scikit-Learn and Tensor flow, O'Reilly, First Edition, 2017.
4. Francois Chollet, Deep Learning with Python, Manning Publications Co, First Edition, 2018.
5. Yuxi (Hayden) Liu, Python Machine Learning by Example, First Edition, 2017.
6. Geoffrey Hinton, 2010, https://www.cs.toronto.edu/~hinton/absps/guideTR.pdf

CO Code	Course outcome statement
CO1	Understand the basics of deep neural network architecture
CO2	Understand and apply deep learning hyperparameter tuning for model fitting
CO3	Understand and apply different frameworks to develop deep neural network architecture
CO4	Analyze the performance of different deep neural network models
CO5	Design appropriate deep learning models for real-world problems

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	1				
CO2	3	2	1	1		1
CO3	3	2	1	1		1
CO4	3	2	1	1		
CO5	3	2	1	1		1

COURSE TYPE: Lab
EVALUATION: 70/30

21CS717 IMAGE AND VIDEO PROCESSING 2-0-2-3

PREREQUISITES:

Vector & Matrix Algebra: Vector spaces, Linear independence, Basics of Matrix Algebra, Eigenvalues & Eigenvectors

SYLLABUS

Introduction-Digital Image Fundamentals: Elements of Computer Vision-Light and the Electromagnetic Spectrum-Light and the Electromagnetic Spectrum-Image Sensing and Acquisition-Image Sampling and Quantization-Some Basic Relationships Between Pixels-Introduction to the Basic Mathematical Tools Used in Digital Image Processing- Intensity Transformations and Spatial Filtering: Basic Intensity Transformation Functions-Histogram Processing-Fundamentals of Spatial Filtering -Smoothing (Lowpass) Spatial Filters-Sharpening (High pass) Spatial Filters

Filtering in the Frequency Domain: Sampling and the Fourier Transform of Sampled Functions-Discrete Fourier Transform of One Variable-Extensions to Functions of Two Variables-Properties of the 2-D DFT and IDFT-Filtering in the Frequency Domain-Image Smoothing Using Lowpass Frequency Domain Filters

Digital Images and Video: Human Visual System and Color-Analog Video-Digital Video-3D Video-Motion Estimation: Image Formation-Motion Models-2D Apparent Motion Estimation-Differential Methods-Matching Methods-Image Segmentation

SKILLS ACQUIRED: Understand about image representation-Type of Operations on Images in Spatial and Frequency Domain-Video representation-Algorithms for matching content using Block Matching

TEXT BOOKS/REFERENCES:

1. Rafael C. Gonzalez,Richard E. Woods,"Digital Image Processing" , pearson Education, 4th Edition , 2018
2. A. Murat Tekalp,"Digital Video Processing",O'Reilly, Second Edition,2015

CO code	**Course Outcome**
CO1	Formulate mathematical operations for image processing
CO2	Understand and apply different filtering operations in spatial domain
CO3	Understand and apply different filtering operations in frequency domain
CO4	Understand the principles in video representation and motion analysis
CO5	Understand and apply different block matching algorithms for segmentation

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2			
CO2	3	2	1	2	2	1
CO3	3	2	1	2	2	2
CO4	3	3	2	2	2	
CO5	3	3	2	2	3	2

COURSE TYPE: Lab
EVALUATION: 70/30

21CS718 INTERNET OF THINGS 2-0-2-3

PREREQUISITES: Advanced Networks

SYLLABUS:

Introduction to IoT – IoT definition – Characteristics – IoT Complete Architectural Stack – IoT enabling Technologies – IoT Challenges. Sensors and Hardware for IoT – Hardware Platforms – Arduino, Raspberry Pi, Node MCU. A Case study with any one of the boards and data acquisition from sensors. Protocols for IoT – Application level protocols: MQTT-CoAP,Web Sockets, Infrastructure protocol (IPV4/V6/RPL), Identification (URIs), Transport (Wi-Fi, Li-Fi, BLE), Discovery, Data Protocols, Device Management, Service Discovery and Management Protocols. IoT privacy, security and vulnerability solutions. Integration of IoT end devices with Edge and Cloud Environment.

Case studies with architectural analysis: IoT applications – Smart City – Smart Water – Smart Agriculture – Smart Energy – Smart Healthcare – Smart Transportation – Smart Retail – Smart waste management.

Skills Acquired: Good hands-on exposure in various IoT enabling technologies. Good practical knowledge in various application layer protocols for IoT systems. Practical exposure in design and deployment of IoT systems for a specific use case.

TEXT BOOKS / REFERENCES:

1. "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", by Pethuru Raj and Anupama C. Raman, CRC Press.
2. Sudip Misra, Anandarup Mukherjee, Arijit Roy, Introduction to IoT, Cambridge University Press, 2020
3. Adrian McEwen, Designing the Internet of Things, Wiley,2013.

CO Code	Course outcomes
CO1	Understand various concept of IoT and related technologies.
CO2	Develop an IoT application using different hardware and software platforms
CO3	Implement various IoT Protocols in different layers
CO4	Design and deploy IoT applications for a given use case

CO Code	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**
CO1	2	2	2			
CO2	2	3	2	2	2	2
CO3	2	2	2			
CO4	2	3	2	2	2	2

COURSE TYPE: Lab
EVALUATION: 70/30

21CS719 IOT PROTOCOLS AND ARCHITECTURE 2-0-2-3

PREREQUISITES: Internet of Things, Advanced Networks

SYLLABUS:

Introduction to IOT, Applications of IOT, Use cases of IOT, The IoT Architectural Reference Model as Enabler, IoT Reference Model: Domain, information, functional and communication models; IoT Reference Architecture: Architecture, Functional, information, deployment and operation views; SOA based Architecture, API-based Architecture, OPENIoT Architecture for IoT/Cloud Convergence Application Protocols for IoT: UPnP, CoAP, MQTT, XMPP. SCADA, WebSocket; IP-based protocols: 6LoWPAN, RPL; Authentication Protocols; IoT in Practice-Examples: IoT in Logistics and Health care. Case study: Cloud-Based Smart-Facilities Management, Healthcare, Environment Monitoring System.

SKILLS ACQUIRED:

Practical exposure to choose and design any IoT application based on Industrail standards reference architecture**.** Good Expertise in selection and design of suitable protocols in various layers of IoT / Cyber Physical System. Hands-on experience in design of IoT systems for the given scenario and applying necessary measures to secure the entire system.

TEXT BOOKS / REFERENCES:

1. Bassi, Alessandro, et al, "Enabling things to talk", Springer-Verlag Berlin An, 2016.
2. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", CISCO Press, 2017
3. Hersent, Olivier, David Boswarthick, and Omar Elloumi. The internet of things: Key applications and protocols. John Wiley & Sons, 2011.
4. Buyya, Rajkumar, and Amir Vahid Dastjerdi, eds. Internet of Things: Principles and paradigms. Elsevier, 2016.

CO Code	Course Outcome
CO1	Comprehend the essentials of IoT and its applications
CO2	Understand the concepts of IoT Architecture Reference model
CO3	Analyze various IoT Application layer Protocols
CO4	Apply IP based protocols and Authentication Protocols for IoT
CO5	Design IoT-based systems for real-world problems

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	2			
CO2	2	2	2			
CO3	2	2	2			
CO4	2	2	2			
CO5	2	2	2	1	2	2

COURSE TYPE: Lab
EVALUATION: 70 / 30

21CS721 OFFENSIVE CYBER SECURITY 2-0-2-3

PREREQUISITES

Basic knowledge of Networking protocols, Windows & Linux commands, Tools such as Bash Shell Scripting and Wireshark, SQL Commands, Oracle or MySQL Databases

SYLLABUS

Introduction to Offensive networking security, Penetration testing, and Ethical hacking – Types of Penetration Testing: Network, Web Application, Wireless network, and Physical, Social Engineering, Testing Methodologies: Black box/White box – Intelligence Gathering: Passive Open-Source Information Gathering: OSINT Framework, Email Harvesting, Password Dumps, Maltego Active Information Gathering: Port Scanning with Nmap, TCP/UDP Scanning – Threat Modeling – Common Vulnerability Analysis: using Nessus, Banner Grabbing – Web Application Assessment: Exploitation, Burp Suite, SQL Injection, Cross-Site Scripting (XSS) – CVEs, National Vulnerability Database, CWE/SANS TOP 25 Most Dangerous Software Errors, CIS: 20 Critical Security Controls; Buffer Overflows: Windows, Linux – Client-Side Attacks – Antivirus Circumvention – Privilege Escalation – Password Attacks – Active Directory Attacks – MetaSploit Framework – Red Team vs Blue Team

SKILLS ACQUIRED: Hands-on practice with Hypervisor such as VirtualBox or VMWare; Basic skills on SQL, Scripting languages, and Anti-malware tools

TEXT BOOKS/REFERENCES:

1. Linux Basics for Hackers: Getting Started with Networking, Scripting, and Security in Kali by OccupyTheWeb, Kindle Edition, No Starch Press, 2018.
2. Georgia Weidman, Penetration Testing: A Hands-On Introduction to Hacking, 1st Edition, Kindle Edition. No Starch Press, 2014.
3. Dafydd Stuttard, Marcus Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, 2nd Edition, Wiley.com, 2011.
4. James Forshaw, Attacking Network Protocols: A Hacker's Guide to Capture, Analysis, and Exploitation, Kindle Edition, No Starch Press, 2017.
5. Jon Erickson, Hacking: The Art of Exploitation, 2nd Edition, No Starch Press, 2008.

CO Code	Course outcome statement
CO1	Understand the basic principles and features of Offensive Security and Penetration Testing
CO2	Apply penetration testing strategies on various applications to evaluate and enhance the security of software systems
CO3	Identify and exploit common buffer overflow vulnerabilities, application vulnerability analysis using Nessus, Banner Grabbing, and MetaSploit
CO4	Comprehend Common Vulnerability Exposures, Common Weakness Enumeration of SANS Institute, and Critical Security Controls
CO5	Identify and Exploit XSS, SQL Injection, Antivirus Evasion, Privilege Escalation, Password Attacks as a Blue Team and Red Team under the Test Bed Environment

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	2	1	2	3
CO2	2	3	3	3	2	3
CO3	3	3	2	3	2	2
CO4	3	3	2	3	2	2
CO5	3	3	2	3	2	2

COURSE TYPE: Lab
EVALUATION: 70-30

21CS720 LOCATION DATA ANALYSIS AND MODELLING 2-0-2-3

SYLLABUS

Basic Concepts of Spatial Analysis and Space Conceptualization – introduction - Spatial Data - Spatial Relationships - Distance Measure- Contiguity- Interaction - Neighbourhood and Neighbours - Spatial Weights and Row Standardization. Tools and Descriptive Statistics for Visualizing Spatial Data - Bivariate Analysis - Inferential Statistics - Normal Distribution.

Analyzing Geographic Distributions and Point Patterns - Analyzing Spatial Patterns - Point Pattern Analysis Methods - Spatial Autocorrelation - Global Spatial Autocorrelation - Incremental Spatial Autocorrelation - Local Spatial Autocorrelation - Space-Time Correlation Analysis - Multivariate Data analysis - Data Reduction and Clustering - Multivariate Data Analysis - Principal Component Analysis - Factor Analysis - Multidimensional Scaling – Regionalization - Similarity Analysis.

Modeling Relationships: Regression and Geographically Weighted Regression - Simple Linear Regression - Multiple Linear Regression - Evaluating Linear Regression Results - Metrics, Tests and Plots – Multicollinearity - Geographically Weighted Regression -Spatial Econometrics - Spatial Econometrics - Spatial Dependence - Spatial Lag Model - Spatial Error Model - Spatial Filtering - Spatial Heterogeneity: Spatial Regression Models.

SKILLS ACQUIRED

Spatial data analysis and modeling course is interdisciplinary and trains students to analyze and build spatial models using GIS data in areas such as local-based service, social science and environment. Experience in understanding and processing diverse spatial data open new frontiers for scientific research and provide employment opportunities in GIS and analytics domain.

TEXTBOOK/REFERENCES

1. Grekousis, George. Spatial analysis methods and practice: describe–explore–explain through GIS. Cambridge University Press, 2020.
2. Yamagata, Yoshiki, and Hajime Seya, eds. Spatial analysis using big data: Methods and urban applications. Academic Press, 2019
3. Haining, Robert P., and Robert Haining. Spatial data analysis: theory and practice. Cambridge university press, 2003.
4. Bivand, Roger S., et al. Applied spatial data analysis with R. Vol. 2. New York: Springer, 2013.
5. Lansley, G., and J. Cheshire. "An Introduction to Spatial Data Analysis and Visualisation in R." CDRC Learning Resources (2016).

CO Code	Course Outcome
CO1	Understand the fundamentals concepts of spatial data for real world applications and model them.
CO2	Apply mapping techniques to explore spatial data
CO3	Analyze geographic distributions, point patterns and spatial autocorrelation in data
CO4	Apply cluster analysis and regression and spatial econometrics techniques on data

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	2	1	
CO2	2	3	2	2	1	2
CO3	2	3	2	3	1	2
CO4	3	2	1	3	1	

COURSE TYPE: Lab
EVALUATION: 70-30

21CS722 PRIVACY ENGINEERING 2-0-2-3

SYLLABUS

Privacy Concepts, Privacy Principles and Policies, Authentication and Privacy, Privacy preserving Data Mining, Privacy on the Web, Email Security, Privacy Impacts of Emerging Technologies.Data Privacy: Privacy, need for Data Sharing, Anonymization design principles, Data Anonymization in multidimensional data, Data Anonymization in time series data ,Threats to anonymized data, Privacy preserving data mining, Dynamic data Protection.Case study in Security Design : Kerberos, Default Hadoop Model without security, Hadoop Kerberos Security, Open source authentication in Hadoop, Log monitoring, Encryption for Hadoop.

SKILLS ACQUIRED

Learners would be able to have a strategic view on how to build an information security framework that preserves privacy and align with business objectives. He/She would be able to limit the ability of an attacker to corrupt or modify data in the event of a security breach.

TEXTBOOKS/REFERENCES

1. Mark Van Rijmenam, "Think Bigger: Developing a Successful Big Data Strategy for Your Business", Amazon, 1 edition, 2014.
2. Ben Spivey, Joey Echeverria, "Hadoop Security Protecting Your Big Data Problem", O'Reilly Media, 2015.
3. Nataraj Venkataramanan, Ashwin Shriram," Data Privacy: Principles and Practice", Chapman and Hall/CRC; 1 edition , 2016
4. Michael E. Whitman and Herbert J Mattord, "Principles of Information Security", 6th edition, Vikas Publishing House, 2017.

CO Code	Course Outcomes
CO1	Understand the privacy lifecycle principles and identify appropriate techniques for risk mitigation
CO2	Apply and critique strategies for personal privacy protection.
CO3	Understand and build security systems for big data.
CO4	Build security in Hadoop environment.

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	2	2	
CO2	2	2	2	2	2	
CO3	2	3	2	3	3	3
CO4	3	3	2	3	3	3

COURSE TYPE: Project

EVALUATION: 70-30

21CS723 REMOTE SENSING AND GIS 2-0-2-3

PREREQUISITES: Digital Signal Processing.

SYLLABUS

Foundations of Remote Sensing - Introduction - Electromagnetic spectrum – wavelength regions important to remote sensing – Wave theory, Particle theory, Stefan-Boltzman and Wein's Displacement Law – Atmospheric scattering, absorption – Atmospheric windows –Global Positioning System - Global Navigation Satellite Systems - Spectral signature – Spectral signature curves - reflective characteristics– EMR interaction with water, soil and earth surface. Earth Resource Satellites - Landsat – SPOT- Hyper spectral - Meteorological satellites - Ocean monitoring satellites – Applications

Basics of photographic systems- Basics of Photogrammetry, Multi spectral imaging, Thermal imaging, Hyper spectral sensing - Image Pre-processing: Radiometric and Geometric Correction - Image Enhancement - contrast manipulation, spatial filtering and edge enhancement, multi spectral and hyper spectral image manipulations and analysis.

Fundamentals of GIS, vector, raster and attribute data models, vector and raster data structure, spatial data input and editing, visualization and query of spatial data, spatial data transformations, spatial analysis.

SKILLS ACQUIRED

Spatial data analysis and modeling course is interdisciplinary and trains students to analyze and build spatial models using GIS data in areas such as local-based service, social science and environment. Experience in understanding and processing diverse spatial data open new frontiers for scientific research and provide employment opportunities in GIS and analytics domain.

TEXTBOOKS/REFERENCES

1. Lillesand, Thomas, Ralph W. Kiefer, and Jonathan Chipman. Remote sensing and image interpretation. John Wiley & Sons, 2015.
2. Unger Holtz, Tanya S. "Introductory digital image processing: A remote sensing perspective." (2007): 89-90.
3. Jensen, John R. Remote sensing of the environment: An earth resource perspective 2/e. Pearson Education India, 2009.
4. DeMers, Michael N. Fundamentals of geographic information systems. John Wiley & Sons, 2008.
5. Lo, Chor Pang, and Albert KW Yeung. Concepts and techniques of geographic information systems. Pearson Prentice Hall, 2007.
6. Chang, Kang-Tsung. Introduction to geographic information systems. Vol. 4. Boston: McGraw-Hill, 2008

CO Code	Course Outcome
CO1	Understand and apply the concepts and laws related to remote sensing
CO2	Understand the features and working of various remote sensing platforms
CO3	Understand the characteristics of different types of remote sensors and apply for different applications
CO4	Apply the concept of image processing and interpretation techniques for satellite data
CO5	Create applications using GIS tools, analyse and interpret data

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	1		1	
CO2	2	1	2			
CO3	2	1	1	1	1	
CO4	2	2	3	2	2	2
CO5	2	3	3	2	2	2

COURSE TYPE: Lab
EVALUATION: 70-30

21CS724 SPECIAL TOPICS IN CYBER-SECURITY 3-0-0-3

PREREQUISITES: Foundations of Cybersecurity, Cyber-physical systems, Cloud Computing and IoT
Note - Special Topics in Cybersecurity may change from year-to-year based on current trends in the field.

SYLLABUS:

Insider Threats in Cyber Security - Defining Insider Threats - Detecting and Identifying Insider Threats - Assessing and Managing Insider Threats.
Securing the Cloud Infrastructure - Securing the Platform. Restricting Network Access through Security Groups. Configuring Platform-Specific User Access Control. Integrating Cloud Authentication/Authorization Systems. Compartmentalizing Access to Protect Data Confidentiality and Availability. Securing Data in Motion and Data at Rest. Identifying Your Security Perimeter. Cloud Access Control and Key Management. Cloud Computing Architecture and Security Concepts. Secure Cloud Architecture. Designing Resilient Cloud Architectures.

IoT Security – Today's IoT attacks - Security Engineering for IoT Development - IoT Security Lifecycle - Identity and Access Management Solutions for the IoT - Mitigating IoT Privacy Concerns -Cloud Security for the IoT – Case study: Smart home IoT security, Industrial IoT security and Cybersecurity for SCADA and Industrial Control Systems

Cybersecurity and Privacy in Cyber-physical Systems: Towards secure software-defined networking integrated cyber-physical systems: Attacks and countermeasures - Detecting pilot contamination attacks in wireless- Security and privacy in Big Data- Case study: Secure cyber-physical systems for smart cities.

SKILLS ACQUIRED: –Students will be able to analyze the insider threats that an insider intentionally or unintentionally misuses access to organization's critical information. Students will be able to assess the security of cloud computing platforms. Students will be able to acquire the concepts of IoT security and strategic measures for secure usage. Students will be able to acquire the knowledge on the security of cyber-physical systems for the evolving skills.

TEXT BOOKS/REFERENCES:

1. Insider Threat Mitigation Guide, November 2020, Cybersecurity and Infrastructure Security Agency.
2. John R. Vacca (editor), Cloud Computing Security: Foundations and Challenges, by CRC Press, 2017.
3. Brian Russell and Drew Van Duren, Practical Internet of Things Security, Packt publishing, 2016.
4. Maleh Yassine, Mohammad Shojafar, Cybersecurity and Privacy in Cyber Physical Systems, May 2019, Publisher: CRC Press Taylor & Francis, ISBN: 9781138346673

CO Code	Course outcome statement
CO1	Identify the purposes and methods of insider threats.
CO2	Assess the potential threats in cloud computing and the roadblocks in building secure cloud computing platforms
CO3	Analyze the IoT security attacks, exposures, threats, and frame strategic measures for their secure usage and operation.
CO4	Analyze the security of cyber-physical systems

CO Code	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	3	2	2	3
CO2	3	3	3	3	3	3
CO3	3	3	3	3	3	3

CO4	3	3	3	3	2	3

COURSE TYPE: Theory/Case study
EVALUATION: 70/30

21CS725 Large Scale Data Processing Using Full Stack – (Technology Elective) 2 – 0 – 0 – 2

Course Outcomes

COs	Description
CO1	Understand the technology trend and its relevance for industries
CO2	Understand and apply the tools related to the technology trend for solving problems pertaining to IT industry
CO3	Be able to relate fundamental concepts in Computer Science to technology trends
CO4	Demonstrate self-exploration and assimilation of new tools and technique

CO-PO Mapping

COs /POs	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	2		
CO2	2	3	3	3	3	2
CO3	3	3	3	2	1	
CO4	2	3	3	2	1	2

Introduction to Full-Stack development

The following sessions are 75 minutes each of duration

1. Introduction to React
2. Communicating with server
3. Programming a server with NodeJS and Express
4. Graph QL
5. React Native
6. CI/CD
7. Containers
8. Using relational databases

Assessments:

One project-based assessment where the students will be required to implement a web app involving the above topics. The evaluation will be done by our faculty.

Large Scale Data Processing

Introduction to Data Processing: The need for large-scale data processing will be covered in this session. There will be a briefly study on some important characteristics of data that influence data processing. The scope of this course, i.e., what is covered and what is not covered in the course, will be clarified.

No. of sessions: 1

Hours: 2

Understand Your Data: Various characteristics of data that are important to consider when processing data will be explored in detail. How these data characteristics influence the data processing will also be discussed.
Class Exercise: Signup for Microsoft Azure account using your university email id and setup Azure Synapse workspace. Azure Synapse service will be used for the upcoming assignment.
No. of sessions: 1
Hours: 2

Batch Data Processing: Various use-cases for batch data processing and how it is typically done for large data sets will be explored. Some of the popular batch data processing systems that are used in industry will also be covered.
Class Exercise: Implementing and executing a batch data processing job using Azure Synapse service.
No. of sessions: 2
Hours: 4

Data Stream: What a data stream is and how it is used in industry will be discussed. Various real-world use-cases where data streams are used will also be discussed. Some of the important aspects to consider when processing data streams will also be discussed.
Class Exercise: A data stream using Azure EventHub service will be created. We will implement and execute a script that will ingest sample events/data to the stream that we created.
No. of sessions: 1
Hours: 2

Stream Data Processing: Various use-cases for stream data processing and how it is typically done at scale will be explored. We will also briefly explore some of the popular data stream processing systems used in industry.
Class Exercise: A sample stream using Azure EventHub service will be created. A stream processing job using Azure Stream Analytics service will be created and executed.
No. of sessions: 1
Hours: 2

Data Pipeline: What a data pipeline means and the need for the same will be discussed. Various real-world use-cases for data pipeline will be discussed. Some of the important aspects to consider when building a data pipeline will also be discussed.
Class Exercise: A sample data pipeline using Azure Data Factory service will be created and executed.
No. of sessions: 1
Hours: 2

Q&A Session: This session is dedicated to the Q&A based on the previous sessions
No. of sessions: 1
Hours: 2

Assessments :

Assignment 1: Parse a large dataset using any programming language/script of your choice and perform some aggregation on the dataset. The goal of this task is to understand the constraints that we typically encounter when processing large dataset on your laptop (or) single server.
Evaluation: Student will upload their executable binary, their source code along with the instruction to run it. Student will get full credit if their code is implemented as expected and produces correct result.

Assessment 2: Implement a query to perform some data aggregation on a large dataset. Execute the query using Azure Synapse SQL service.
Evaluation: Student will upload a video in which they demonstrate how their batch processing job works. Student will get full credit if there were able to demonstrate the batch processing functionality successfully. Along with the video, student will upload the code/query for the batch processing job.

Assessment 3: Write a stream processing job using Azure Stream Analytics service.
Evaluation: Student will upload a video in which they demonstrate how their stream processing job works. Student will get full credit if there were able to demonstrate the stream processing functionality successfully. Along with the video, student will upload the code for the stream processing job.

Assessment 4: Implement a data pipeline (using Azure Data Factory service) that connects the stream processing job & batch processing job that are already implemented in assessment 2 & 3.
Evaluation: Student will upload a video in which they demonstrate how the pipeline works. Student will get full credit if there were able to demonstrate the data pipeline functionality successfully.

21RM605 Research Methodology 2-0-0-2

Course Syllabus

Meaning of Research, Types of Research, Research Process, Problem definition, Objectives of Research, Research Questions, Research design, Approaches to Research - Quantitative vs. Qualitative Approach, Understanding Theory, Building and Validating Theoretical Models, Exploratory vs. Confirmatory Research, Experimental vs Theoretical Research, Importance of reasoning in research.

Problem Formulation, Understanding Modeling & Simulation, Conducting Literature Review, Referencing, Information Sources, Information Retrieval, Role of libraries in Information Retrieval, Tools for identifying literatures, Indexing and abstracting services, Citation indexes

Experimental Research: Cause effect relationship, Development of Hypothesis, Measurement Systems Analysis, Error Propagation, Validity of experiments, Statistical Design of Experiments, Field Experiments, Data/Variable Types & Classification, Data collection, Numerical and Graphical Data Analysis: Sampling, Observation, Surveys, Inferential Statistics, and Interpretation of Results

Preparation of Dissertation and Research Papers, Guidelines for writing the abstract, introduction, methodology, results and discussion, conclusion sections of a manuscript. References, Tables and illustrations and Citation.

Intellectual property rights (IPR) – patents – copyrights - Trademarks - Ethics of Research- Scientific Misconduct - Forms of Scientific Misconduct – Plagiarism - Unscientific practices in thesis work.

References:

1. Bordens, K. S. and Abbott, B. B., "Research Design and Methods – A Process Approach", 11thEdition, McGraw-Hill, 2022.
2. Roy Sabo and Edward Boone, "Statistical Research Methods: A Guide for Non-Statisticians", Springer, 2013.
3. Davis, M., Davis K., and Dunagan M., "Scientific Papers and Presentations", 3rd Edition, Elsevier Inc., 2013.
4. Ron Iphofen (Ed), "Handbook of Research Ethics and Scientific Integrity", Springer, 2020.
5. Elsevier, "Ethics in Research & Publication", https://www.elsevier.com/__data/assets/pdf_file/0008/653885/Ethics-in-research-and-publication-brochure.pdf

Course Outcomes

COs	Description
CO1	Understand the basic concepts of research and its methodologies
CO2	Understand and apply the process of searching for, selecting and critically analysing research articles and papers
CO3	Formulate and evaluate research questions and apply the process of designing a research study and interpreting the outcomes of the study
CO4	Write and present a research report and thesis

CO-PO Mapping

COs	PO1	PO2	PO3	PO 4	PO 5	PO6
CO1	1		1	1		
CO2		1	2	2		1
CO3	2	2		2	3	2
CO4				2		2

COURSE TYPE: Theory/Case study
EVALUATION: 70/30

21CS798 Dissertation Phase I 10

Course Objectives

The student is expected to carry out supervised research resulting in comprehensive analytical study of the area of interest. The student should be able to conduct an intensive review of literature, and analysis of existing work through comprehensive experimental analysis resulting in identification of a novel and well-defined problem .

Course Outcomes

COs	Description
CO1	Demonstrate sound fundamentals in a chosen area of computing.
CO2	Identify and formulate the research problem through conduct of an intensive review of literature
CO3	Analyze existing work solving the defined problem through comprehensive experimental analysis
CO4	Effectively communicate the work at all stages of the project adhering to ethical practices

References

1. Relevant literature for the computing problem.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	2	2	2
CO2	2	3	3	3	2	3
CO3	3	3	3	2	2	3
CO4	2	3	3	2	2	3

COURSE TYPE: Project
EVALUATION: 80/20

21CS799 Dissertation Phase II 16

Course Objectives

The course shall be pursued for a minimum of 16 weeks during the final semester, following the preliminary work carried out in Phase-1 during the previous semester. The student is expected to enhance the knowledge base in the chosen field of research in computing. The student demonstrates the ability to design effective and novel solutions to the defined problem in Phase 1 and through rigorous experimental and theoretical analysis demonstrate the effectiveness of the proposed solution. The student is also expected to effectively communicate the scholarly outcomes as presentations and report and publish the same in reputed conference or journal.

Course Outcomes

COs	Description
CO1	Identify gaps and needs in the chosen areas to refine the problem defined.
CO2	Design and develop novel and efficient solutions to the problem and analyze results.
CO3	Prepare the thesis report and defend the thesis based on the work done.
CO4	Augment the knowledge base in the chosen area of computing by publishing scholarly articles, and adhere to ethical practices at every stage.

Prerequisites

- Dissertation Phase I

References

1. Relevant literature for the computing problem.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	3	3	3	2
CO2	3	3	3	3	2	3
CO3	3	3	3	2	2	3
CO4	3	3	3	2	3	3

COURSE TYPE: Project
EVALUATION: 80/20