CURRICULUM & SYLLABI
B.Tech. COMPUTER SCIENCE & ENGINEERING

Effective from AY: 2024-25

NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

WARANGAL, TELANGANA

Table of Contents

Vision and Mission of the Institute
National Institute of Technology Warangal

VISION
Towards a Global Knowledge Hub, striving continuously in pursuit of excellence in Education, Research, Entrepreneurship and Technological services to the society

MISSION
- Imparting total quality education to develop innovative, entrepreneurial and ethical future professionals fit for globally competitive environment.
- Allowing stakeholders to share our reservoir of experience in education and knowledge for mutual enrichment in the field of technical education.
- Fostering product-oriented research for establishing a self-sustaining and wealth creating centre to serve the societal needs.

Vision and Mission of the Department
COMPUTER SCIENCE & ENGINEERING

VISION
Attaining global recognition in Computer Science & Engineering education, research and training to meet the growing needs of the industry and society

MISSION
- Imparting quality education through well-designed curriculum in tune with the challenging software needs of the industry.
- Providing state-of-art research facilities to generate knowledge and develop technologies in the thrust areas of Computer Science and Engineering.
- Developing linkages with world class organizations to strengthen industry-academia relationships for mutual benefit

Program: B.Tech. Computer Science & Engineering

Program Educational Objectives

PEO-1	Apply computer science theory blended with mathematics and engineering to model computing systems.
PEO-2	Design, implement, test and maintain software systems based on requirement specifications
PEO-3	Communicate effectively with team members, engage in applying technologies and lead teams in industry.
PEO-4	Assess the computing systems from the view point of quality, security, privacy, cost, utility, etiquette and ethics.
PEO-5	Engage in lifelong learning, career enhancement and adapt to changing professional and societal needs

Program Articulation Matrix

PEO / Mission Statements	PEO-1	PEO-2	PEO-3	PEO-4	PEO-5
Imparting quality education through well-designed curriculum in tune with the challenging software needs of the industry.	3	3	2	3	2
Providing state-of-art research facilities to generate knowledge and develop technologies in the thrust areas of Computer Science and Engineering.	2	3	3	3	2
Developing linkages with world class organizations to strengthen industry-academia relationships for mutual benefit	2	3	2	3	2

1 - Slightly; 2 - Moderately; 3 - Substantially

Program: B.Tech. Computer Science & Engineering

Program Outcomes

PO-1	**Engineering knowledge**: Apply the knowledge of mathematics, science, Engineering fundamentals, and computer science and engineering to the solution of complex engineering problems.
PO-2	**Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
PO-3	**Design/Development of solutions:** Design solutions for complex computer science & engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO-4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO-5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex civil engineering activities with an understanding of the limitations.
PO-6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO-7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO-8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO-9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO-10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO-11	**Project management and Finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO-12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

PSO-1	Design algorithms for real world computational problems and analyze their complexities.
PSO-2	Design and develop interfaces among subsystems of computing.
PSO-3	Analyze large data samples and discover knowledge to provide solutions to engineering problems.
PSO-4	Assess security, privacy, quality and cost parameters in developing software systems.

CURRICULUM
B.Tech. Computer Science & Engineering

1st Semester

S.No.	Code	Course Title	L-T-P	Credits
1	MA1161	Linear Algebra, Calculus and Ordinary Differential Equations	3-0-0	3
2	PH1161	Engineering Physics	3-0-2	4
3	HS1161	English for Technical Communication	2-0-2	3
4	CS1101	Programming and Data Structures	3-0-0	3
5	BT1161	Biology for Engineers	2-0-0	2
6	CS1103	Programming and Data Structures Lab	0-1-2	2
7	IC1101	EAA–I (Games & Sports / Yoga & Wellness)	0-0-0	0
			Total Credits	**17**

2nd Semester

S.No.	Code	Course Title	L-T-P	Credits
1	MA1162	Integral and Vector Calculus, Laplace and Fourier Transforms	3-0-0	3
2	EE1161	Basic Electrical & Electronics Engineering	3-0-0	3
3	CS1102	Design Thinking	0-1-4	3
4	CS1104	Optimization Techniques	3-0-0	3
5	CS1106	Data Structures and Algorithms	3-0-2	4
6	CS1108	Web Programming Lab	0-1-2	2
7	CS1110	Mathematical Foundations of CS	3-0-0	3
8	IC1102	EAA–II (Games & Sports / Yoga & Wellness)	0-0-0	0
			Total Credits	**21**

3rd Semester

S.No.	Code	Course Title	L-T-P	Credits
1	MA1264	Probability and Statistics	3-0-0	3
2	EC1263	Computer Organization and Architecture	3-0-2	4
3	CS1201	Theory of Computation	3-1-0	4
4	CS1203	Design and Analysis of Algorithms	3-0-0	3
5	CS1205	Object Oriented Programming	3-0-0	3
6	CS1207	Object Oriented Programming Laboratory	0-1-2	2
7	CS1209	Statistical Tools Practice	0-1-2	2
		Total Credits		**21**

4th Semester

S.No.	Code	Course Title	L-T-P	Credits
1	MS1262	Business Essentials for Engineers	3-0-0	3
2	CS1202	Compiler Design	3-0-2	4
3	CS1204	Software Engineering	3-0-2	4
4	CS1206	Database Management Systems	3-0-0	3
5	CS1208	Operating Systems	3-0-0	3
6	CS1210	Database Systems Laboratory	0-1-2	2
7	CS1212	Operating Systems Laboratory	0-1-2	2
		Total Credits		**21**

5th Semester

S.No.	Code	Course Title	L-T-P	Credits
1	CS1301	Data Warehousing and Data Mining	4-0-0	4
2	CS1303	Full Stack Development	3-0-2	4
3	CS1305	Computer Networks	3-0-0	3
4	CS1307	Distributed Computing	3-0-0	3
5	CS13XX	Professional Elective - I	3-0-0	3
6	CS1381	Fractal Course - I	0-0-1	0.5
7	CS1309	Computer Networks Laboratory	0-1-2	2
8	CS1311	Knowledge Engineering Laboratory	0-1-2	2
			Total Credits	**21.5**

6th Semester

S.No.	Code	Course Title	L-T-P	Credits
1	CS1302	Cryptography and Network Security	4-0-0	4
2	CS1304	Machine Learning	3-0-0	3
3	CS13XX	Professional Elective - II	3-0-0	3
4	CS13XX	Professional Elective - III	3-0-0	3
5	CS1306	Product Development	0-1-4	3
6	CS1382	Fractal Course - II	0-0-1	0.5
7	CS1308	Cryptography and Network Security Laboratory	0-1-2	2
8	CS1310	Machine Learning Laboratory	0-1-2	2
			Total Credits	**20.5**

7th Semester

S.No.	Code	Course Title	L-T-P	Credits
1	CS1401	High Performance Computing	3-0-0	3
2	CS14XX	Professional Elective-IV	3-0-0	3
3	CS14XX	Professional Elective-V	3-0-0	3
4		Open Elective - I	2-0-0	2
5	CS1403	High Performance Computing Laboratory	0-1-2	2
6	CS1489	Seminar and Technical Writing	0-0-0	2
7	CS1495	Minor Project	0-0-0	2
8	CS1491	Short Term Industrial/EPICS/Research Experience	0-0-0	2
			Total Credits	**19**

8th Semester

S.No.	Code	Course Title	L-T-P	Credits
1	CS14XX	Professional Elective - VI	3-0-0	3
2	CS14XX	Professional Elective - VII	3-0-0	3
3	CS14XX	Professional Elective - VIII	3-0-0	3
4	CS1498	Major Project	0-0-0	6
			Total Credits	**15**

Professional Elective Courses:

Professional Elective-I		
S.No.	**Code**	**Course Title**
1	CS1321	Advanced Algorithms
2	CS1323	Advanced Computer Architecture
3	CS1325	Artificial Intelligence
4	CS1327	Computational Number Theory
5	CS1329	Programming Language Concepts
6	CS1331	Software Testing

Professional Elective-II, III		
S.No.	**Code**	**Course Title**
1	CS1322	Advanced Data Mining
2	CS1324	Advanced Databases
3	CS1326	Bio-Informatics
4	CS1328	Computer Vision and Image Processing
5	CS1330	Data Networks
6	CS1332	Design Patterns
7	CS1334	Foundations of Blockchain Technology
8	CS1336	Information Retrieval
9	CS1338	Mobile Computing
10	CS1340	Natural Language Processing
11	CS1342	Program Analysis and Verification
12	CS1344	Software Metrics and Software Project Management

Professional Elective-IV, V		
S.No.	**Code**	**Course Title**
1	CS1421	Cloud Computing
2	CS1423	Data Science
3	CS1425	Deep Learning
4	CS1427	Design of Secure Protocols
5	CS1429	Internet of Things
6	CS1431	Intrusion Detection Systems
7	CS1433	Quantum Computing
8	CS1435	Real Time Systems
9	CS1437	Secure Software Engineering
10	CS1439	Security and Privacy
11	CS1441	Service Oriented Architecture
12	CS1443	Software Reliability Techniques
Professional Elective-VI, VII, VIII		
S.No.	**Code**	**Course Title**
1	CS1422	Advanced Theoretical Computer Science
2	CS1424	Agri Bioinformatics
3	CS1426	Algorithmic Game Theory
4	CS1428	Algorithmic Techniques for Big Data
5	CS1430	Computational Neuro Science
6	CS1432	Cyber Laws and IPR
7	CS1434	Cyber Security
8	CS1436	Fog and Edge Computing
9	CS1438	Formal Methods in Software Engineering
10	CS1440	Game Theory
11	CS1442	GPU Architecture and Computing
12	CS1444	Heterogeneous Computing
13	CS1446	Human Computer Interaction
14	CS1448	IoT Security
15	CS1450	Medical Image Processing
16	CS1452	Semantic Web
17	CS1454	Social Networks
18	CS1456	Software Defined Networks
19	CS1458	Virtual Reality and Augmented Reality

Basic Science Courses

S.No.	Code	Course Title	L-T-P	Credits
1	MA1161	Linear Algebra, Calculus and Ordinary Differential Equations	3-0-0	3
2	PH1161	Engineering Physics	3-0-2	4
3	BT1161	Biology for Engineers	2-0-0	2
4	MA1162	Integral and Vector Calculus, Laplace and Fourier Transforms	3-0-0	3
5	MA1264	Probability & Statistics	3-0-0	3

Engineering Science Courses

S.No.	Code	Course Title	L-T-P	Credits
1	CS1101	Programming and Data Structures	3-0-0	3
2	CS1103	Programming and Data Structures Lab	0-1-2	2
3	EE1161	Basic Electrical & Electronics Engineering	3-0-0	3
4	CS1102	Design Thinking	0-1-4	3
5	CS1104	Optimization Techniques	3-0-0	3
6	CS1106	Data Structures and Algorithms	3-0-2	4
7	CS1209	Statistical Tools Practice	0-1-2	2
8	EC 1263	Computer Organization and Architecture	3-0-2	4
9	CS1306	Product Development	0-1-4	3

Humanities and Social Science Courses

S.No.	Code	Course Title	L-T-P	Credits
1	HS1161	English for Technical Communication	2-0-2	3
2	MS1262	Business Essentials for Engineers	3-0-0	3

The Overall Credit Structure

Course Category	Credits
Basic Science	15
Engineering Science	27
Humanities and Social Sciences	6
Program Core	82
Professional Elective	24
Open Elective	2
Total Graded Credit Requirement	156

Minor Degree in Computer Science & Engineering:

Code	Course Title	L-T-P	Credits	Sem.
CS1M01	Data Structures	2-0-2	3	III
CS1M02	Principles of Database Systems	2-0-2	3	IV
CS1M03	Operating System Principles	3-0-0	3	V
CS1M04	Computer and Communication Networks	3-0-0	3	VI
CS1M05	Software Engineering Principles	3-0-0	3	VII
CS1M06	Pattern Recognition & Machine Learning	3-0-0	3	VII or VIII

A student should complete a minimum of five courses (15 Credits) to get Minors degree.

Honors Degree in Computer Science & Engineering:

Code	Course Title	L-T-P	Credits	Sem.
CS1H01	Advanced Compiler Design	3-0-0	3	V
CS1H02	Advanced Software Engineering	3-0-0	3	VI
CS1H04	Advanced Computer Networks	3-0-0	3	VI
CS1H05	Digital Video Processing	3-0-0	3	VII
CS1H07	Data Privacy	3-0-0	3	VII
CS1H09	Responsible and Explainable AI	3-0-0	3	VII
CS1H10	Algorithmic Coding Theory	3-0-0	3	VIII
CS1H12	Soft Computing Techniques	3-0-0	3	VIII
CS1H14	Reinforcement Learning	3-0-0	3	VIII

A student should complete a minimum of five courses (15 Credits) to get Honors degree.

SYLLABI

B.Tech. Computer Science and Engineering

1st Semester

MA1161 **2-0-0 (3)**

Linear Algebra, Calculus and Ordinary Differential Equations

Pre-requisites: None

Course Outcomes:

CO-1	Understand to solve the consistent system of linear equations.
CO-2	Apply orthogonal transformations to a quadratic form.
CO-3	Determine the series expansion of a given function.
CO-4	Explore the properties of functions of several variables.
CO-5	Solve arbitrary order linear differential equations.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	1	2	1	–	–	–	–	–	–	–	1	1	1	1
CO-2	3	3	1	2	1	–	–	–	–	–	–	–	–	–	1	–
CO-3	3	3	1	2	1	–	–	–	–	–	–	–	1	–	1	1
CO-4	3	3	1	2	1	–	–	–	–	–	–	–	–	–	–	1
CO-5	3	3	1	2	1	–	–	–	–	–	–	–	2	1	1	2

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Linear Algebra: Vector space, Subspace, Examples, Linear span, Linear independence and dependence, Basis, Dimension, Extension of a basis of a subspace, Intersection and sum of two subspace, Examples. Linear transformation, Kernel and Range of a linear map, Rank-Nullity Theorem (without proof). Rank of a matrix, Row, and column spaces, Solvability of the system of linear equations, Inner product spaces, Orthogonal basis, Gram-Schmidt orthogonalization process. Eigenvalues, Eigenvectors, and properties Caley-Hamilton Theorem (without proof) and applications, diagonalization of a matrix, diagonalization by similarity, and orthogonal transformations.

Differential Calculus: Taylor's theorem with remainders; Taylor's and Maclaurin's expansions;Functions of several variables – continuity, differentiability, partial derivatives, Euler's theorem, change of variables, Jacobians, Functional dependence; Maxima andminima of functions of several variables (2 and 3 variables) - Lagrange's method of multipliers.

Ordinary Differential Equations: First order differential equations - Basic definitions, Geometric interpretation of solutions of first-order ODE $y' = f(x, y)$, Exact differential equations, Integrating factors, Linear equation, Reducible to linear form: Bernoulli's equations; Orthogonal trajectories; Higher order linear differential equations with constant coefficients - Cauchy-Euler and Legendre's differential equations, Method of variation of parameters - System of linear differential equations; Applications to physical problems.

Learning Resources:

Text Books:

1. Howard Anton and Chris Rorres, Elementary Linear Algebra with Supplementary Applications, John Wiley & Sons, 2014, Eleventh Edition.
2. George B. Thomas and Ross L. Finney, Calculus and Analytic Geometry, Pearson, 2020, Ninth Edition.
3. Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley and Sons, 2015, Eighth Edition.

Reference Books:

1. Dennis G. Zill, Advanced Engineering Mathematics, Jones & Bartlett Learning, 2018, Sixth Edition.
2. R. K. Jain and S. R. K. Iyengar, Advanced Engineering Mathematics, Narosa Publishing House,

2016, Fifth Edition.

PH1161 **3-0-2 (4)**

Engineering Physics

Pre-Requisites: **None**

Course Outcomes:

CO1	Examine the concepts of Interference, diffraction, polarisation to solve engineering problems.
CO2	Assess the technological applications of lasers and optical fibers.
CO3	Apply the quantum mechanical principles for solving engineering problems.
CO4	Understand the basics of nanomaterials and their engineering applications.
CO5	Demonstrate the production, detection and applications of ultrasonics.

Course Articulation Matrix:

	PO1	PO	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO22	PSO3	PSO4
CO1	3	3	1	2	–	–	–	–	–	–	–	–	3	2	–	–
CO2	3	3	2	2	–	–	–	–	–	–	–	–	3	2	–	–
CO3	3	3	2	2	–	–	–	–	–	–	–	–	3	2	–	–
CO4	3	3	2	2	–	–	–	–	–	–	–	–	3	2	–	–
CO4	3	3	2	2	–	–	–	–	–	–	–	–	3	2	–	–

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Interference: Principle of Superposition, Coherence and Coherent Sources, Production of Coherent Light, Young's Double Slit Experiment, Concept of interference, Newton's Rings, working of Michelson Interferometer, Fabry-Perot Interferometer, and its application as wavelength filter.

Diffraction: Definition and types of Diffractions, Huygen's Principle and types of wave fronts, types of Diffraction, Single Slit Diffraction, Double Slit Diffraction, Diffraction Grating, Derivation of Resolving Power and Dispersive Power, Rayleigh's Criterion and applications.

Polarization: Introduction to Polarization, Production of Polarized Light by Reflection and Refraction, Phenomenon of Double Refraction, Construction and Working of Nicol's Prism, Half-Wave and Quarter Waveplates, Representation of Different Polarized Lights, Optical Activity, Practical Applications of Polarized Light, Construction and Working of Laurent's Half Shade Polarimeter and Engineering Applications.

Lasers & Optical Fibers: Basic theory of Laser, Fundamentals of lasers, Einstein Coefficients, Characteristics of Laser Pumping Mechanisms; Basic Components of Laser System, 2-Level, 3-Level and 4-Level Systems, Construction and working of He-Ne, Nd-YAG, and semiconductor diode Lasers and Engineering Applications of Lasers. Basic Principle of Optical Fiber, Derivation-Numerical Aperture and Acceptance Angle, Types of Optical Fibers (Step and Graded Index, Single Mode and Multimode), Applications in Communications and Sensors.

Quantum Mechanics: Concepts and experiments that led to the discovery of Quantum Nature, de Broglie hypothesis of matter waves, Heisenberg uncertainty principle, Schrodinger time independent and time dependent wave equations, the free particle problem, Particle in an infinite and finite potential well, Quantum mechanical tunnelling and applications.
Nanomaterials: Introduction and importance of Nanomaterials, classification (0D, 1D, 2D and 3D) of nanomaterials, properties of nanomaterials, carbon-based nanomaterials, synthesis of nanomaterials, top-down and bottom-up approaches, characterization of nanomaterials, Engineering Applications of Nanomaterials:

Ultrasonics: Production, detection, and applications of ultrasonics

List of Experiments:

1. Determination of Wavelength of Sodium light using Newton's Rings.
2. Determination of Wavelength of He-Ne laser - Metal Scale.
3. Measurement of Width of a narrow slit using He- Ne Laser.
4. Determination of Specific rotation of Cane sugar by Laurent Half-shade Polarimeter.
5. Determination of Numerical aperture, loss, Acceptance angle of optical fiber.
6. Determination of plank constant by photo electric effect.
7. Determination of I – V characteristics of photo diode.
8. Diffraction grating by normal incidence method.
9. Determination of capacitance by using R-C circuit.
10. Determination of resonating frequency and bandwidth by LCR circuit
11. Strain Gauge
12. Dielectric constant measurements
13. Determination of carrier concentration, charge by using Hall effect experiment
14. Study of I-V characteristics of Solar Cell
15. Determination of velocity of ultrasonic waves and adiabatic compressibility of liquids using ultrasonic interferometer.

Learning Resources:

Text Books:

1. Fundamentals of Physics by Halliday, Resnic and Walker, John Wiley, Ninth Edition, 2011.
2. Concepts of Modern Physics by Arthur Beiser, Shobhit Mahajan, S. Rai Choudhury, McGraw Hill Publications, Sixth Edition, 2009.
3. Engineering Physics by Shatendra Sharma, Jyotnsa Sharma, Pearson Education, 2018.
4. Nanotechnology: principles and practices by Sulabha K. Kulkarnl. Springer, Third Edition, 2018.
5. Practical Physics by G.L. Squire, Cambridge University Press, Fourth Edition, 2001.

Reference Books:

1. Optics by Ajoy K. Ghatak, Tata McGraw Hill, Sixth Edition, 2017.
2. Understanding Lasers An Entry-Level Guide, by Jeff Hecht, Wiley Publications, Fourth Edition, 2018.
3. A Textbook of Engineering Physics by M.N. Avadhanulu, P.G. Khirsagar, Ninth Edition, 2011.
4. University Physics with modern physics, Hugh D. Young,Roger A. Freedman Pearson Education, 2014.
5. Nanotechnology the whole story, B. Rogers, J Adams and S. Pennathur, CRC Press, 2013.
6. Engineering Physics Practical, Dr.S.K. Gupta Krishna Prakashan Publications, Ninth Edition, 2010.

Online Resources:

1. https://nptel.ac.in/courses/122/107/122107035/

HS1161 **2-0-2 (3)**

English for Technical Communication

Pre-Requisites: English proficiency above B1 level as per the CEFR (Common European Framework of Reference) for languages.

Course Outcomes:

CO-1	Understand and apply principles of technical communication to interact effectively in diverse environments.
CO-2	Analyze complex technical documents to extract and synthesize key information.
CO-3	Employ reported speech, active and passive voice in engineering and scientific contexts to compile technical reports.
CO-4	Demonstrate use of English speech sounds, stress, and intonation in day-to-day situations, conversations, and interactions.
CO-5	Interpret technical data presented in the form of graphs, pie charts, and diagrams.
CO-6	Critique and provide constructive feedback on peer communication performances and written works.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	–	–	3	–	2	–	–	–	–	–	–	–	–	–	–	–
CO-2	–	–	2	–	2	–	–	–	–	–	–	–	–	–	–	–
CO-3	–	–	2	–	1	–	–	–	–	–	–	–	–	–	–	–
CO-4	–	–	3	–	1	–	–	–	–	–	–	–	–	–	–	–
CO-5	–	–	2	–	1	–	–	–	–	–	–	–	–	–	–	–
CO-6	–	–	2	–	2	–	–	–	–	–	–	–	–	–	–	–

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Grammar Principles & Effective Sentence Construction: Correction of Sentences and Concord, - Vocabulary Building, - Synonyms and Antonyms, - Idioms and Phrasal Verbs: Patterns of Use and Suggestions for Effective Employment in Varied Contexts, - Technical Vocabulary (Jargons and Registers),- Strategies for Bringing Variety and Clarity in Sentences, - Removing Ambiguity, - Editing Long Sentences for Brevity and Clarity, - Reported Speech, - Contexts for Use of Reported Speech, - Impact on Audiences and Readers, - Active and Passive Voice, - Reasons for Preference for Passive Voice in Scientific English

Writing- Paragraph & Letter: Definition of Paragraph and Types, - Features of a Good Paragraph, - Unity of Theme, - Coherence, - Linking Devices, - Direction, - Patterns of Development. - Importance in the Context of Other Channels of Communication, - Qualities of Effective Letters, - Types of Letters, - Official Letters, - Letters for Various Purposes, - Letters of Application for Jobs, - Cover Letter and Resume Types, - Letters for Internship/Fellowship, - Writing Statements of Purpose (SOPs), - Examples and Exercises

Reading Techniques: Definition and Importance, - Skills and Sub-Skills of Reading, - Skimming and Scanning: Uses and Purposes, Examples and Exercises, - Reading Comprehension, - Reading Silently and with Understanding, - Process of Comprehension, - Types of Comprehension Questions

Technical Writing: Principles of a Technical Report, - Know Your Audience, Purpose, and Length of Report, - Understand the Cornerstones of a Presentation, - Define Various Purposes of Presentations and Plan the Correct Structure, - Writing Clear Sentences and Paragraphs, - Removing Jargon, Redundancy, and Wordiness, - Kinds of Graphics and Their Messages, - Suitability for Placement in Graphic Representation, - Introduction to Basic Concepts in Research, - Abstract, Keywords, Methodology, Hypothesis, Plagiarism, Critical Reading, - Abstract Writing, - How to Read Scientific Articles, - Basics of Writing a Research Project Proposal, - Preparation and Presentation of Project Reports

Reviews: Oral and Written Review of a Chosen Novel/Play/Movie, - Review of Scientific Articles and Science Fiction, Focus on Appropriate Vocabulary and Structure, - Use of Special Vocabulary and Idioms

Language laboratory:

English Sound System -vowels, consonants, Diphthongs, phonetic symbols- using dictionary to decode phonetic transcription-- Received Pronunciation, its value and relevance- transcription.

Stress and Intonation –word and sentence stress - their role and importance in spoken English- Intonation in spoken English -definition, -use of intonation in daily life-exercises

Introducing oneself in formal and social contexts- Role plays. - their uses in developing fluency and communication in general.

Oral presentation - definition- occasions- structure- qualities of a good presentation with emphasis on body language and use of visual aids.

Listening Comprehension- Challenges in listening, good listening traits, some standard listening tests- practice and exercises.

Debate/ Group Discussions-concepts, types, Do's and don'ts- intensive practice, Guided writing practice with examples, Drafting – the mindset to avoid writer's block, Checking your own reports and presentations, Giving and receiving constructive feedback.

Learning Resources:

Text Books:

1. English for Engineers and Technologists (Combined edition, Vol. 1 and 2) Orient Blackswan 2010.
2. Ashraf, M Rizvi. Effective Technical Communication. Tata McGraw-Hill, 2006
3. Meenakshi Raman and Sangeetha Sharma. Technical Communication: Principles and Practice Oxford University Press, 2nd Edition, 2011.
4. Tan, Zhongchao. Academic Writing for Engineering Publications: A Guide for Non-native English Speakers. Springer, 2022.

CS1101 **3-0-0 (3)**

Programming and Data Structures

Pre-Requisites: **None**

Course Outcomes:

CO-1	Design algorithms for solving simple mathematical problems including computing, searching and sorting.
CO-2	Compare and contrast algorithms in terms of space and time complexity to solve simple mathematical problems.
CO-3	Explore the internals of computing systems to suitably develop efficient algorithms.
CO-4	Examine the suitability of data types and structures to solve specific problems.
CO-5	Apply control structures to develop modular programs to solve mathematical problems.
CO-6	Understand the concept of abstract data types and apply them in real-world applications.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	1	1	1	2	1	-	-	-	-	-	3	3	3	3	3
CO-2	2	1	2	1	2	3	-	-	-	-	-	3	3	2	2	2
CO-3	1	2	2	2	2	1	-	-	-	-	-	3	3	2	2	1
CO-4	2	2	2	2	2	2	-	-	-	-	-	2	3	2	1	2
CO-5	2	2	3	1	2	2	-	-	-	-	-	2	3	2	2	2
CO-6	2	2	3	2	2	2	-	-	-	-	-	2	3	2	2	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Fundamentals of Computers - Components of a computers, Problems, Flowcharts, Memory, Variables, Values, Instructions, Programs.

Problem solving techniques – Algorithmic approach, characteristics of algorithm, Problem solving strategies: Top-down approach, Bottom-up approach, Number systems and data representation.

Elements of C++ programming language - Data types, constants, and variables, expressions and assignment statements, input and output statements, conditional and branch statements: If-else, Switch-case constructs, iteration statements: while, do-while, for, Arrays – Single and Multi-Dimensional Arrays, strings. Bit-wise operations.

Functions and Recursion – Modular approach for solving real time problems, user defined functions, library functions, parameter passing - call by value, call by reference, return values, passing arrays as parameters to functions Recursion.

Structures and Classes - Declaration, member variables, member functions, access modifiers, function overloading, Problems on Complex numbers, Date, Time, Large Numbers.

Pointers and Files - Introduction to pointers and dynamic allocation, String processing, File operations-create, read and write.

Searching and sorting - Linear and binary search, selection sort, bubble sort, insertion sort, merge sort, quick sort.

Data structures - Abstract Data Types (ADTs) – Stack ADT – Array-Based Implementation of Stack – Applications, Queue ADT – Array-Based Implementation – Applications

Learning Resources:

Text Books:

1. Walter Savitch, Problem Solving with C++, Pearson, 2014, Ninth Edition.
2. Cay Horstmann, Timothy Budd, Big C++, Wiley, 2009, Second Edition.

Reference Books:

1. R.G. Dromey, How to solve it by Computer, Pearson, 2008.
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Pearson Education, 2006, Third Edition.

BT1161 **2-0-0 (2)**

Biology for Engineers

Pre-Requisites: **None**

Course Outcomes:

CO-1	Realize the significance of biomolecules for sustaining life.
CO-2	Identify the difference between unicellular to multi-cellular organisms.
CO-3	Understand heredity, variation and central dogma of life.
CO-4	Apply the concepts of biology for engineering the cell.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3
CO-1	3	2	2	–	–	2	2	–	–	2	–	3	3	3	3
CO-2	3	2	2	–	–	2	–	–	–	–	–	3	3	3	3
CO-3	3	2	2	–	–	2	–	–	–	–	–	3	3	3	3
CO-4	3	3	2	2	1	2	2	1	–	2	–	3	3	3	3

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Molecules of Life: Chemical basis of life, Proteins, Nucleic acids, Carbohydrates, Lipids, Membranes and First cell, Inside the cell, Cell cycle and Division.

Information processing in living system: Central dogma, Concept of Gene, Genetic code, Transcription, Translation, Biological signal transduction, Quorum sensing and Biofilm formation.

Biomolecular machines and motors: Cytoskeletal motor proteins, ATP synthase, Cell motility.

Applied Biotechnology: Biocomputing, Synthetic biology, Biosensors, Biomedical instrumentation in disease diagnosis, Biomimicry, Biomechanics, Biomaterials, Nanobiotechnology, Industrial and Environmental Biotechnology, Biosafety and Bioethics.

Learning Resources:

Text Books:

1. Quillin, Allison Scott Freeman, Kim Quillin and Lizabeth Allison, Biological Science, Pearson Education India, 2016.
2. Reinhard Renneberg, Viola Berkling and Vanya Loroch, Biotechnology for Beginners, Academic Press, 2017.

1103 **0-1-2 (2)**

Programming and Data Structures Laboratory

Pre-Requisites: **None**

Course Outcomes:

CO-1	Design and test programs to solve mathematical and scientific problems.
CO-2	Develop and test programs using control structures.
CO-3	Implement modular programs using functions.
CO-4	Develop programs using classes.
CO-5	Develop ADT for stack and queue applications.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	2	2	1	2	1	-	-	-	-	-	-	3	3	3	3
CO-2	1	1	2	1	2	2	-	-	-	-	-	-	3	3	2	2
CO-3	1	2	3	2	2	1	-	-	-	-	-	-	3	3	2	2
CO-4	2	2	2	2	2	3	-	-	-	-	-	-	2	3	2	1
CO-5	2	2	2	2	2	3	-	-	-	-	-	-	2	3	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

1. Programs on conditional control constructs.
2. Programs on loops (while, do-while, for).
3. Programs using user defined functions and library functions.
4. Programs on arrays, matrices (single and multi-dimensional arrays).
5. Programs using pointers (int pointers, char pointers).
6. Programs on structures.
7. Programs on classes and objects.
8. Programs of stack and queue.

Learning Resources:

Text Books:

1. Walter Savitch, Problem Solving with C++, Ninth Edition, Pearson, 2014.
2. Cay Horstmann, Timothy Budd, Big C++, Wiley, Second Edition, 2009.

Reference Books:

1. R.G. Dromey, How to solve it by Computer, Pearson, 2008.
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Pearson Education, Third Edition, 2006.

2nd Semester

MA1162 **3-0-0 (2)**

Integral and Vector Calculus, Laplace and Fourier Transforms

Pre-requisites: MA1161

Course Outcomes:

CO-1	Analyze improper integrals for extracting certain properties of beta and gamma integrals.
CO-2	Evaluate multiple integrals in different coordinate systems.
CO-3	Apply the concepts of gradient, divergence and curl of scalar and vector point functions to formulate engineering problems.
CO-4	Find Laplace transforms of functions.
CO-5	Find Fourier Series and Fourier Transforms of functions.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	1	2	1	–	–	–	–	–	–	–	–	1	2	–
CO-2	3	3	1	2	1	–	–	–	–	–	–	–	–	1	2	–
CO-3	3	3	1	2	1	–	–	–	–	–	–	–	–	1	2	–
CO-4	3	3	1	2	1	–	–	–	–	–	–	–	–	1	2	–
CO-5	3	3	1	2	1	–	–	–	–	–	–	–	–	1	2	–

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Integral Calculus: Improper integrals; Beta and Gamma functions, and their properties; Differentiation under integral sign, Evaluation of double and triple integrals; Areas and Volumes, Change of order of integration; Change of variables in double and triple integrals.

Vector Calculus: Scalar and vector fields; Vector differentiation; Level surfaces; Directional derivative; Gradient of a scalar field; Divergence and curl of a vector field; Laplacian operator; Parametrization of curves and surfaces; Line, surface and volume integrals; Green's theorem in a plane; Stoke's theorem; Gauss divergence theorem.

Laplace Transforms: Laplace transforms; Inverse Laplace transforms; Properties of Laplace transforms; Laplace transforms of unit step, impulse and periodic functions; Convolution theorem;. **Fourier Series:** Expansion of a function in Fourier series for a given range - Half range sine and cosine expansions.

Fourier Transforms : Fourier transformation and inverse transforms - sine, cosine transformations and inverse transforms.

Learning Resources:

Text Books:

1. George B. Thomas and Ross L. Finney, Calculus and Analytic Geometry, Pearson, 2020, Ninth Edition
2. Dennis G. Zill, Advanced Engineering Mathematics, Jones & Bartlett Learning, 2018, Sixth Edition

Reference Books:

1. Maurice D. Weir, Joel Hass and Christopher Heil, Thomas' Calculus: Early Transcendentals, Pearson, 2014, Thirteenth Edition.
2. Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley and Sons, 2015, Eighth Edition.
3. R. K. Jain and S. R. K. Iyengar, Advanced Engineering Mathematics, Narosa Publishing House,

2016, Fifth Edition.

EE1161 **3-0-0 (3)**

Basic Electrical & Electronics Engineering

Pre-Requisites: **None**

Course Outcomes:

CO-1	Able to Analyze DC & AC circuits and determine power & power factor.
CO-2	Able to Understand the operation and characteristics of various electrical machines.
CO-3	Understand the operation of basic electronic circuits and characteristics of semiconductor devices.
CO-4	Able to select appropriate meters/transducers for measurement of various electrical /non-electrical quantities.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	2	2	2	2	1	1	1	1	2	1	--	--	--	--
CO-2	3	3	2	2	2	2	1	1	1	1	2	1	--	--	--	--
CO-3	3	3	2	2	2	2	1	1	1	1	2	1	--	--	--	--
CO-4	3	3	2	2	2	2	1	1	1	1	2	1	--	--	--	--

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

DC Circuits: Kirchhoff's Voltage and Current Laws, Superposition Theorem, Star-Delta Transformations.

AC Circuits: Complex representation of Impedance, Phasor diagrams, Power & Power Factor, Solution of 1-Phase Series & Parallel Circuits.

Single Phase Transformers: Principle of Operation of a Single-Phase Transformer, EMF Equation, Phasor Diagram, Equivalent Circuit of a 1-Phase Transformer, Determination of Equivalent circuit parameters, calculation of Regulation & Efficiency of a Transformer.

DC Machines: Principle of Operation, Classification, EMF and Torque Equations, Characteristics of Generators and Motors. Speed Control Methods.

AC Machines: 3-Phase Induction Motor- Principle of Operation, Torque – Speed Characteristics of 3-Phase Induction Motor & Applications, Principle of Operation of Alternator- EMF equation.

Electronic Devices & Circuits: P-type and N-Type semiconductors, P-N junction diode and its I-V characteristics, Single-phase Half-wave and Full wave rectifiers. Bipolar Junction Transistor-operation and CE, CC & CB configurations, Static Characteristics of SCR-MOSFET- IGBT.

Sensors & Transducers: Thermocouple, Thermistor, Resistance Temperature Detector, Hall effect and Piezoelectric Transducers (Qualitative Treatment only)

Electrical Measuring Instruments: Moving Coil & Moving iron ammeters & voltmeters. Wattmeter's (Qualitative).

Electronics Measurements : Principle of Operation of Digital Multi Meter & Cathode Ray Oscilloscope

Learning Resources:

Text Books:

1. Edward Hughes, Electrical & Electronic Technology, Pearson Education, 2016, 12th Edition.
2. Vincent Del Toro, Electrical Engineering Fundamentals, Pearson Education, 2015, 2nd Edition.
3. V. K Mehtha, Principals of Electrical & Electronics Engineering, S. Chand Publications, New Delhi, 2010, 3rd Edition.
4. V N Mittle and Arvind Mittal, Basic Electrical Engineering, Tata McGraw Hill, 2005, 2nd Edition.

Reference Books:

1. Millman&Halkias, Integrated Electronics - Analog and Digital Circuit and Systems, Tata McGraw-Hill Education, 2017, 2nd edition.
2. U Bakshi& A. Bakshi, Basic Electrical Engineering, Technical Publications, 2019.
3. A Fitzgerald, Charles Kingsley, Stephen Umans, Electrical Machines, McGraw Hill Education, 2017, 6th edition.
4. Stephen.J.Chapman, Electric Machinery, McGraw Hill International Edition, 2017, 4th edition.
5. P.S. Bimhbra, Electrical Machinery - Theory, Performance & Applications, Khanna Publishers 2014, 7th edition.

Other Suggested Readings:

1. https://nptel.ac.in/courses/108/108/108108076/

CS1102 **0-1-4 (3)**

Design Thinking

Pre-Requisites: **None**

Course Outcomes:

CO-1	Identify user needs.
CO-2	Define problems to stimulate ideation.
CO-3	Ideate on problems to propose solutions by working collaboratively.
CO-4	Test aspects of proposed solutions.
CO-5	Improve solutions by gaining user feedback.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	3	3	1	2	2	2	2	3	2	3	1	3	c3	3	3
CO-2	2	3	3	2	2	2	2	2	3	2	2	1	3	3	3	3
CO-3	2	3	3	3	3	2	2	2	3	2	3	1	3	3	3	3
CO-4	-	-	3	3	3	1	1	2	3	2	3	2	3	3	3	3
CO5	-	-	3	3	3	2	2	2	2	2	3	2	3	3	3	3

Syllabus:

Introduction to Engineering: "Engineering" as a vehicle for social and economic development; the impact of science/engineering on our day-to-day lives; the process of engineering a product; various career options.

Introduction and identifying the need: Understanding the unique needs of the user - empathize - define - ideate - prototype - test. Case Studies - Develop an appreciation for the design process and its application in specific settings (Guest lectures, Videos, Field visits, Interplay lectures of design-based movies).

Problem Formulation: Framing a problem statement neutrally using adequate checks. Case studies.

Concept Generation: Generate multiple concepts using various creativity tools and thinking styles.

Prototyping: Select from ideas and make quick prototypes (mock-ups) using available material.
Evaluation: Iterative process of ideation, prototyping and testing-Take the mock-ups to users for feedback and iterate the process till users feel delighted.

Activities:

Some of the activities which are undertaken as a part of this course include:

- Field Visits
- Case Studies on innovation, failures etc
- Guest lecture
- Group Discussions
- Presentation by student
- Experiential learning workshops

Learning Resources:

Text Books:

1. Design Thinking: A guide to creative problem solving for everyone, Andrew Pressman, Routledge Taylor and Francis group, 2019, 1st Edition.
2. Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation, Tim Brown.

Reference Books:

1. George E. Dieter, Linda C. Schmidt Engineering Design 5th Edition, 2019.
2. Ulrich, K., Eppinger, S. and Yang, M.,. Product Design and development, 7th Edition, 2020.

CS1104 **3-0-0 (3)**

Optimization Techniques

Pre-Requisites: **CS1101, CS1106**

Course Outcomes:

CO-1	Prepare and solve linear programming model.
CO-2	Model transportation and flow through networks and compute optimal parameters.
CO-3	Optimize inventory levels.
CO-4	Solve real life problems using Meta-heuristic techniques.
CO-5	Generate random numbers and random variates.
CO-6	Verify and validate simulation models.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	1	-	-	-	1	-	-	-	-	1	1	1	1	2
CO-2	3	3	3	2	2	-	1	-	-	-	1	1	1	2	2	2
CO-3	2	2	2	1	1	-	1	-	-	-	3	-	2	1	1	2
CO-4	2	2	2	1	2	-	1	-	-	-	2	1	2	1	1	2
CO-5	2	2	2	1	1	-	-	-	-	-	-	1	1	1	2	-
CO-6	2	2	1	-	-	-	-	-	-	-	-	-	2	2	1	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Modelling with linear programming – The Simplex method, Sensitivity Analysis, Integer linear programming: Branch and Bound technique – Transportation Model and its variants, Network Model: CPM and PERT - Deterministic and non-deterministic inventory models.

Heuristic and Meta Heuristic Programming: Simulated Annealing, Genetic Algorithm, Particle warm Optimization algorithm and Teaching learning-based optimization algorithm - Non-Linear Programming algorithms.

Introduction to Quadratic Programming: Constrained Optimization Problem Solving, Convex Optimization Methods.

Simulation Modelling: Random number generation, Random variate generation – Verification and Validation of simulation models, Simulation of Computer Systems and Computer Networks.

Learning Resources:

Text Books:

1. Hamdy A Taha – “Operations Research-An Introduction”, 9th Edition, Pearson, 2017 (Chapters 1-8, 12, 14, 17)
2. Jerry Banks, Hon S Carson, Barry L Nelson, David M Nicol, “Discrete Event Simulation", 5th Edition, Pearson, 2010 (Chapters 8 – 12, 14, 15)

CS1106 **3-0-2 (4)**

Data Structures and Algorithms

Pre-Requisites: **CS1101**

Course Outcomes:

CO-1	Understand the concept of ADT, identify data structures suitable to solve problems.
CO-2	Develop and analyze algorithms for stacks, queues.
CO-3	Develop algorithms for binary trees and graphs.
CO-4	Implement sorting and searching algorithms.
CO-5	Implement symbol table using hashing techniques and multi-way search trees.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	3	1	1	1	-	-	-	-	-	-	3	3	2	3
CO-2	3	3	3	2	1	-	-	-	-	-	-	-	3	3	2	3
CO-3	3	3	3	2	1	2	-	-	-	-	-	-	3	3	2	3
CO-4	3	3	3	3	2	1	-	-	-	-	-	-	3	3	2	3
CO-5	3	3	3	3	1	2	-	-	-	-	-	-	3	3	2	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Iterative and Recursive Algorithms: Abstract Data Types (ADTs), Implementation and Applications of Stacks, Operations and Applications of Queues, Array Implementation of Circular Queues, Implementation of Stacks using Queues, Implementation Queues using Stacks, Linked Lists, Search and Update Operations on Varieties of Linked Lists, Linked List Implementation of Stacks and Queues.

Trees: Introduction, Implementation of Trees, Binary Trees, Tree Traversals with an Application, Binary Search Trees (BSTs), Query and Update Operations on BSTs, AVL Trees, Rotations, Search and Update Operations on Balanced BSTs, Splay Trees, B-trees, Trie, C-Trie.

Hashing: Implementation of Dictionaries, Hash Function, Collisions in Hashing, Separate Chaining, Open Addressing, Analysis of Search Operations.

Priority Queues: Priority Queue ADT, Binary Heap Implementation and Applications of Priority Queues, Disjoint Sets.

Sorting Algorithms: Stability and In Place Properties, Insertion Sort, Merge Sort, Quick Sort, Heap Sort, Lower Bound for Comparison Based Sorting Algorithms, Linear Sorting Algorithms: Counting Sort, Radix Sort, Bucket Sort.

Graph Algorithms: Graphs and their Representations, Graph Traversal Techniques: Breadth First Search (BFS) and Depth First Search (DFS), Applications of BFS and DFS, Minimum Spanning Trees (MST), Prim's and Kruskal's algorithms for MST, Connected Components, Dijkstra's Algorithm for Single Source Shortest Paths, Biconnected Components.

Learning Resources:

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Second Edition, PHI, 2009.
2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, Third Edition, Pearson Education, 2006

Reference Books:

1. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, Second Edition, Universities Press, 2011.
2. Michael T.Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis and Internet Examples, Second Edition, Wiley-India, 2006.

CS1108 **0-1-2 (2)**

Web Programming Lab

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand, analyze and build dynamic and interactive web sites.
CO-2	Understand current and evolving Web languages for integrating media and user interaction in both front end and back end elements of a Web site.
CO-3	Analysis and reporting of web data using web analytics.
CO-4	Applying different testing and debugging techniques and analyzing the web site effectiveness.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	-	-	2	1	2	1	-	-	-	2	2	1	2	3	1	1
CO-2	1	1	-	-	-	-	2	1	1	-	1	1	-	2	1	-
CO-3	1	1	-	2	3	-	-	1	-	1	1	-	2	1	1	-
CO-4	1	-	-	1	-	-	-	1	-	1	-	-	-	2	3	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to XHTML: Editing XHTML, First XHTML Example, W3C XHTML Validation Service, Headers, Linking, Images, Special Characters and More Line Breaks, Unordered Lists, Nested and Ordered Lists, Internet and World Wide Web Resources.

Dynamic HTML: Object Model and Collections- Introduction, Object Referencing, Collections all and children, Dynamic Styles, Dynamic Positioning, Using the frames Collection, navigator Object, Summary of the DHTML Object Model; Event Model- vent onclick, Event onload, Error Handling with onerror, Tracking the Mouse with Event onmousemove, Rollovers with onmouseover and onmouseout; Form Processing- Form Processing with onfocus and onblur, More Form Processing with onsubmit and onreset, Event Bubbling, More DHTML Events; Filters and transitions; Data binding with tabular data control, Structured graphics and active X control.

JavaScript: Functions; Program Modules in JavaScript, Programmer Defined Functions, Function Definitions, Random-Number Generation, Duration of Identifiers, Scope Rules, JavaScript Global Functions, Recursion, JavaScript arrays, JavaScript objects;

Learning Resources:

Text Books:

1. Deitel, Deitel and Nieto, Internet and Worldwide Web - How to Program, 5th Edition, PHI, 2011.
2. Bai and Ekedhi, The Web Warrior Guide to Web Programming, 3rd Edition, Thomson, 2008.

CS1110 **3-0-0 (3)**

Mathematical Foundations of Computer Science

Pre-Requisites: None

Course Outcomes:

CO-1	Apply formal methods of proof to solve discrete problems.
CO-2	Apply Propositional logic and First order logic to solve problems.
CO-3	Formulate and solve graph problems.
CO-4	Formulate and solve recurrence relations.
CO-5	Apply techniques for counting discrete event occurrences.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	2	1	2	1	-	-	-	-	1	-	1	3	-	-	-
CO-2	3	2	1	1	1	-	-	-	-	1	-	-	3	-	-	-
CO-3	3	3	2	2	1	-	-	-	-	1	-	-	3	-	2	-
CO-4	3	2	1	1	1	-	-	-	-	1	-	-	3	-	-	-
CO-5	3	3	-	1	1	-	-	-	-	-	-	-	3	-	-	2

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Mathematical Logic and Normal Forms: Statements and Notation, Connectives, Functionally Complete Set of Connectives, Methods of Proof of an Implication, Inference Theory of Propositional Logic, Normal Forms, Limitations of Propositional Logic, Quantified Propositions, Predicate Logic and other Methods of Proof, Rules of Inference for Quantified Propositions.

Graph Theory: Basic Definitions, Representation of Graphs, Connected Components, Connectivity of Graphs, Block Graphs, Trees, Graph Isomorphism, Planar Graphs, Euler's Formula, Eulerian Graphs, Hamiltonian Graphs, Chromatic Number of a Graph, Brook`s Theorem, Planar Graph Coloring, Applications of Graph Coloring, Domination Number of a Graph, Bounds on Domination Number, Applications of Domination.

Recurrence Relations: Solving Recurrence Relations by Substitution Method, The Method of Characteristic Roots, Solutions of Homogeneous and Inhomogeneous Recurrence Relations using Characteristic Roots Method, Closed Form Formula for nth Fibonacci Number, Counting Minimum Number of Nodes in an AVL Tree of Given Height, Generating Functions of Sequences, Solving Recurrence Relations using Generating Functions, Counting Number of Binary Search Trees using Generating Functions.

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutations with Constrained Repetitions, Binomial Coefficients, The Binomial and Multinomial Theorems, The Principle of Inclusion-Exclusion, Stirling Numbers of the Second Kind.

Learning Resources:

Text Books:

1. Kenneth H. Rosen, "Discrete Mathematics and Its Applications with Combinatorics and Graph Theory", McGraw Hill Education, 2011, 7th Edition.
2. Joe L. Mott, Abraham Kandel, Theodore P. Baker, "Discrete Mathematics for Computer Scientists and Mathematicians", PHI, 2001, 2nd Edition.

Reference Books:

1. Gary Chartrand and Ping Zhang, Introduction to Graph Theory, McGraw-Hill Higher Education, 2006, 1st Edition.
2. Tremblay J. P. and Manohar R., "Discrete Mathematical Structures", McGraw Hill Education, 2017, 1st Edition.

Other Suggested Readings:

1. NPTEL Course on Discrete Mathematical Structures by Prof. Kamala Krithivasan, IIT Madras.

3rd Semester

MA1264 **3-0-0 (3)**

Probability and Statistics

Pre-requisites: None

Course Outcomes:

CO-1	Check the dependence of random variables.
CO-2	Find the mean and variance of a given probability distribution.
CO-3	Test the hypothesis for small and large samples.
CO-4	Apply techniques for point, interval estimations of parameters of various distributions.
CO-5	Understand Markov Chains and stationary distributions.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	1	2	1	–	–	–	–	–	–	–	–	1	2	–
CO-2	3	3	1	2	1	–	–	–	–	–	–	–	–	1	2	–
CO-3	3	3	1	2	1	–	–	–	–	–	–	–	–	1	2	–
CO-4	3	3	1	2	1	–	–	–	–	–	–	–	–	1	2	–
CO-5	3	3	1	2	1	–	–	–	–	–	–	–	–	1	2	–

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Review of basics of Probability and descriptive statistics.

Random variables and their distributions: Random variables (discrete and continuous), Probability functions, Density and distribution functions, Mean and variance, two dimensional random variables joint probability mass functions, conditioning and independence

Chebyshev's inequality, Markov's inequality, Chernoff bounds, Weak law of large numbers, central limit theorem, strong law of large numbers.

Probability Distributions: Bernoulli and Poisson processes Special distributions (Binomial, Poisson, Geometric, Uniform, exponential and normal),

Hypothesis Testing: Parameter and statistic, Concept of null and alternative hypotheses, Acceptance and critical regions, Probabilities of Type I and Type II errors, Level and Power of a test. Concept of p-value, Large sample tests (tests for single mean, difference of means, single proportion, difference of proportions), Tests for small samples (t-test for single mean and difference of means, test for comparison of variances),

Estimation: Point and interval estimation.. Intervtimation for parameters of normal, binomial and Poisson distributions. Estimation of parameters by maximum Likelihood Estimation method

Stochastic processes, branching processes. Markov chains, classification of states, ideas of stationary distributions. Introduction to Martingales and stopping times.

Learning Resources:

Text Books:

1. S.C.Gupta and V.K.Kapoor, "Fundamentals of Mathematical Statistics", 12th Edition, S.Chand & Co, 2020.
2. R. A. Johnson, Miller and Freund's "Probability and Statistics for Engineers", Pearson Publishers, 9th Edition, 2017.

Reference Books

1. John E. Freund Emeritus, Gary A. Simon, Modern Elementary Statistics, PHI, 2006, Ninth Edition.
2. V.K. Rohatgi and A.K. Md. Ehsanes Saleh, An Introduction to Probability theory and Mathematical Sciences, Wiley, 2001.
3. Sheldon M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, Academic Press, 2014, Fifth Edition.

EC1263 **3-0-2 (4)**

Computer Organization and Architecture

Pre-Requisites: **None**

Course Outcomes:

CO-1	Identify functional units, bus structure and addressing modes.
CO-2	Design the hardwired and micro-programmed control units.
CO-3	Identify memory hierarchy and performance.
CO-4	Design Arithmetic Logic Unit.
CO-5	Interface I/O devices.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	1	1	2	1	-	-	1	-	-	-	1	-	2	1	1	2
CO-2	2	1	2	-	-	-	1	-	-	-	-	-	2	1	1	1
CO-3	2	1	-	-	1	1	-	-	-	-	-	-	2	2	1	1
CO-4	2	1	2	-	-	1	1	-	-	-	-	-	2	1	1	1
CO-5	1	1	2	1	-	-	1	-	-	-	1	-	1	1	1	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Basic Structures of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Software, Performance, Multiprocessors and multicomputer, Historical Perspective .

Machine instructions and Programs: Numbers, Arithmetic Operations and Characters, Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes.

Input/output Organization: Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces.

The Memory System: Some Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed Size and Cost, Cache Memories, Virtual Memories, Memory Management Requirements, Secondary Storage.

Arithmetic: Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-Operand Multiplication, Fast Multiplication, Integer Division, Floating Point Numbers and Operations, Implementing Floating Point Operations.

Basic Processing Unit: Some Fundamental Concepts, Execution of Complete Instruction, Multiple-Bus Organization, Hardwired Control, Micro programmed Control.

.

Pipelining: Basic Concepts, Data Hazards, Instruction Hazards, Influence on Instruction Sets, Data Path and Control Considerations, Super Scalar Operations.

Large Computer Systems: Forms of Parallel Processing, Array Processors, the Structure of General-Purpose Multiprocessors, Interconnection Networks.

Learning Resources:

Text Books:

1. Carl Hamacher, "Computer Organization", 5th Edition, McGraw Hill Publishers, 2002.
2. Wiiliam Stallings, "Computer Organization and Architecture Designing for Performance", 8th Edition, Pearson Education, 2010.

Reference Books:

1. John P Hayes, "Computer Architecture and Organization", 3rd revised Ed., McGraw-Hill, 1998.

CS1201 **3-1-0 (4)**

Theory of Computation

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand formal machines, languages and computations.
CO-2	Design finite state machines for acceptance of strings.
CO-3	Design context free grammars for formal languages.
CO-4	Develop pushdown automata accepting strings.
CO-5	Design Turing machine.
CO-6	Distinguish between decidability and undecidability.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	2	1	2	-	-	-	-	-	-	2	2	3	-	-
CO-2	3	2	2	1	2	-	-	-	-	-	-	2	2	3	-	-
CO-3	3	2	2	1	2	-	-	-	-	-	-	2	2	3	-	-
CO-4	3	2	2	1	2	-	-	-	-	-	-	2	2	3	-	-
CO-5	3	2	2	1	2	-	-	-	-	-	-	2	2	3	-	-
CO-6	2	2	2	1	2	-	-	-	-	-	-	2	2	3	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Automata: The Methods and the Madness: Why study automata theory? Introduction to formal proof; Additional forms of proof; Inductive proofs; The central concepts of automata theory.

Finite Automata: An informal picture of finite automata; Deterministic finite automata; Nondeterministic finite automata; An application: text search; Finite automata with epsilon-transitions.

Regular Expressions and Languages: Regular expressions; Finite automata and regular expressions; Applications of regular expressions; Algebraic laws for regular expressions.

Properties of Regular Languages: Proving languages not to be regular; Closure properties of regular languages; Decision properties of regular languages; Equivalence and minimization of automata.

Context-Free Grammars and Languages: Context-free grammars; Parse trees; Applications of context-free grammars; Ambiguity in grammars and languages.

Pushdown Automata: Definition of pushdown automata; The languages of a PDA; Equivalence of PDA's and CFG's; Deterministic pushdown automata.

Properties of Context-Free Languages: Normal forms for context-free grammars; The pumping lemma for context-free languages; Closure properties of context-free languages; Decision properties of CFL's.

Introduction to Turing Machines: Problems that computers cannot solve; The Turing machine; Programming techniques for Turing machines; Extensions to the basic Turing machine; Restricted Turing machine; Turing machines and computers.

Undecidability: A language that is not recursively enumerable; An undecidable problem that is RE; Undecidable problems about Turing machines; Post's correspondence problem; Other undecidable problems.

Learning Resources:

Text Books:

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education, 2006, Third Edition.

Reference Books:

1. Peter Linz, An Introduction to Formal Languages and Automata, Jones and Bartlett Learning, 2013, Sixth Edition.
2. K. L. P. Mishra and N. Chandrasekaran, Theory of Computer Science: Automata, Languages and Computation, PHI, 2006, Third Edition.

Other Suggested Readings:

1. NPTEL Course on Theory of Computation by Prof. Raghunath Tiwari, IIT Kanpur.

CS1203 **3-0-0 (3)**

Design and Analysis of Algorithms

Pre-Requisites: **CS1101, CS1106**

Course Outcomes:

CO-1	Analyze time and space complexities of algorithms.
CO-2	Identify algorithm design methodology to solve problems.
CO-3	Distinguish between P and NP classes of problems.
CO-4	Design and analyze approximation algorithms for NP-hard problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	1	-	-	-	1	-	-	-	-	1	3	1	1	2
CO-2	-	-	-	-	-	-	-	-	-	-	1	1	3	1	2	2
CO-3	-	-	-	-	-	-	-	-	-	-	3	-	3	1	1	2
CO-4	3	3	1	-	-	-	-	-	-	-	-	-	3	-	-	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Asymptotic Analysis: Asymptotic Notations, Worst-case running time of algorithms.

Divide and Conquer: Master Theorem, Maximum Element in an Unimodal Array, Maximum Subarray Sum Problem, Expected Running Time of Randomized Quick Sort, Strassen's Matrix Multiplication Algorithm, Karatsuba's Large Integer Multiplication and Selection in Worst Case Linear Time.

Dynamic Programming: Elements of Dynamic Programming, Matrix Chain Multiplication Problem, Optimal Binary Search Tree, Rod-Cutting Problem, 0-1 Knapsack Problem, Travelling Salesman Problem, All-Pairs Shortest Paths Problem and Optimal Vertex Cover of a Tree.

Greedy Method: Activity Selection Problem, Fractional Knapsack Problem, Correctness and Running Time Analysis of Prim's and Kruskal's Algorithms for Finding Minimum Spanning Tree and Dijkstra`s Algorihm for Single Source Shortest Path Problem.

Complexity Classes: P, NP, NP-hard, NP-complete, Example NP-complete Problems – Clique, Independent Set and Vertex Cover, Methods to cope-up with NP-hardness of a Problem.

Approximation Algorithms: Approximation Ratio, Absolute Approximation Algorithm for Planar Graph Coloring and 2-approximation Algorithm for Vertex Cover Problem.

Learning Resources:

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, PHI, 2009, Third Edition.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, Universities Press, 2011, Second Edition.

Reference Books:

1. Steven S Skiena, The Algorithm Design Manual, Springer, 2008, Second Edition.
2. Michael R. Garey and David S. Johnson, Computers and Intractability: A Guide the theory of NP-Completeness, W.H. Freeman & Co., 1979.

Other Suggested Readings:

1. NPTEL Course on Design and Analysis of Algorithms by Prof. Abhiram G.Ranade et al.
2. https://www.algorist.com/

CS1205 **3-0-0 (3)**

Object Oriented Programming

Pre-Requisites: **CS1101**

Course Outcomes:

CO-1	Understand object-oriented paradigms: abstraction, encapsulation, inheritance, polymorphism.
CO-2	Learn java concepts like exception handling, interfaces, object classes and various libraries.
CO-3	Design object-oriented solutions for real world problems.
CO-4	Implement the applications using the learnt concepts.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	1	2	-	-	1	-	3	-	-	-	2	1	-	-	3	-
CO-2	1	1	2	3	3	3	-	-	-	3	-	-	-	-	-	-
CO-3	2	1	1	1	1	1	-	-	3	-	3	3	1	3	2	2
CO-4	1	-	1	1	1	2	2	-	3	3	2	2	1	2	2	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Object Oriented Thinking – A way of Viewing the World, Computation as Simulation, Messages and Methods; - A Brief History of Object - Oriented Programming - The History of Java, The White Paper Description; – Object - Oriented Design - Responsibility Implies Noninterference, Programming in the Small and in the Large, Why Begin with Behavior? A Case Study in RDD, CRC Cards – Recording Responsibility, Components and Behavior, Software Components, Formalizing the Interface; - A Paradigm - Program Structure, The Connection to the Java World, Types, Access Modifiers, Lifetime Modifiers; - Ball Worlds – Data Fields, Constructors, Inheritance, The Java Graphics Model, The Class Ball, Multiple Objects of the Same Class; - A Cannon Game – The Simple Cannon Game, Adding User Interaction; Pinball Game Construction Kit – First Version of Game, Adding Targets : Inheritance and Interfaces, Pinball Game Construction Kit : Mouse Events Reconsidered; - Understanding Inheritance – An Intuitive Description of Inheritance, The Base Class Object, Subclass, Subtype, and Substitutability – Forms of Inheritance, Modifiers and Inheritance, The Benefits of Inheritance, The Costs of Inheritance; - A Case Study : Solitaire – The Class Card, The Game – Card Piles- Inheritance in Action, The Application Class, Playing the Polymorphic Game, Building a More Complete Game; - Polymorphism - Varieties of Polymorphism, Polymorphic Variables, Overloading, Overriding, Abstract Methods, Pure Polymorphism.

The AWT – The AWT Class Hierarchy, The Layout Manager, User Interface Components, Panels, Dialogs, The Menu Bar; - Input And Output Streams - Streams versus Readers and Writers, Input Streams, Stream Tokenizer, Output Streams, Object Serialization, Piped Input and Output.

Understanding Graphics - Colour, Rectangles, Fonts, Images, Graphic Contexts, A Simple Painting Program; - Applets And Web Programming – Applets and HTML, Security Issues, Applets and Applications, Obtaining Resources Using an Applet, Combining Applications and Applets.

.

Learning Resources:

Text Books:

1. Timothy Budd, "Object Oriented Programming with JAVA", Updated Edition, Pearson Education, 2009.
2. Herbert Schildt, "Java 2 Complete Reference", TMH, 2010.

Reference Books:

1. Barry J Homes and Daniel T Joyce, Object-Oriented Programming with JAVA, Jones and Bartlett publishers, 2001, 2nd Edition.

CS1207 **0-1-2 (2)**

Object Oriented Programming Laboratory

Pre-Requisites: **None**

Course Outcomes:

CO-1	Develop programs using objects and inheritance in Java Language.
CO-2	Design and implement GUI programs using components in Java Language.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	1	3	3	3	1	2	-	2	2	1	-	-	1	-	-	1
CO-2	1	3	3	3	1	2	-	2	2	-	-	-	1	-	-	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Java Programming:

1. Ball games
2. Cannon game
3. Pinball game
4. Cards game
5. User interface dialogs related programs
6. I/O processing programs.

Learning Resources:

Text Books:

1. Timothy Budd, "Object Oriented Programming with JAVA", Updated Edition, Pearson Education, 2009.
2. Herbert Schildt, "Java 2 Complete Reference", TMH, 2010.

CS1209 **0-1-2 (2)**

Statistical Tools Practice

Pre-Requisites: **MA1161, MA1264**

Course Outcomes:

CO-1	Apply statistical methods to data for inferences.
CO-2	Access online resources for R and import new function packages into the R workspace.
CO-3	Import, review, manipulate and summarize data-sets in R.
CO-4	Perform descriptive analytics over large scale data and apply appropriate statistical tests using R.
CO-5	Explore data-sets to create testable hypotheses and identify appropriate statistical tests.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	1	1	-	1	1	-	-	-	1	1	1	-	1	1	-	1
CO-2	2	-	-	-	-	1	-	-	-	-	2	-	-	-	-	-
CO-3	-	-	-	2	2	-	1	-	-	-	2	1	-	-	-	-
CO-4	2	-	1	1	1	1	-	-	2	1	-	1	1	2	-	1
CO-5	-	1	2	1	2	2	2	-	1	2	1	2	1	-	2	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Data Science and data visualization: Introduction, How to run R, R Sessions and functions, Basic Math, Variables, Data Types, Vectors, Conclusion, Advanced Data Structures, Data Frames, Lists, Matrices, Arrays, Classes**.**

R Programming: Structures, Control Statements, Loops, - Looping Over Nonvictory Sets,- If-Else, Arithmetic and Boolean Operators and values, Default Values for Argument, Return Values, Deciding Whether to explicitly call return- Returning Complex Objects, Functions are Objective, No Pointers in R, Recursion, A Quicksort Implementation-Extended Extended Example: A Binary Search Tree.

Doing Math and Simulation in R: Math Function, Extended Example Calculating Probability- Cumulative Sums and Products-Minima and Maxima- Calculus, Functions Fir Statistical Distribution, Sorting, Linear Algebra Operation on Vectors and Matrices, Extended Example: Vector cross Product- Extended Example: Finding Stationary Distribution of Markov Chains, Set Operation, Input /output, Accessing the Keyboard and Monitor, Reading and writer Files.

Graphics: Creating Graphs, The Workhorse of R Base Graphics, the plot () Function –Customizing Graphs, Saving Graphs to Files.

Probability Distributions: Normal Distribution- Binomial Distribution- Poisson Distributions Other Distribution, Basic Statistics, Correlation and Covariance, T-Tests,-ANOVA.

Linear Models: Simple Linear Regression, -Multiple Regression Generalized Linear Models, Logistic Regression, - Poisson Regression- other Generalized Linear Models-Survival Analysis, Nonlinear Models, Splines- Decision- Random Forests,

Learning Resources:

Text Books:

1. Norman Matloff, "*The Art of R Programmi*ng, No Starch Press, 2011.
2. Lander, "*R for Everyone*", Pearson.

Reference Books:

1. Paul Teetor, "*R Cookbook*", Oreilly
2. Rob Kabacoff, "*R in Action*", Manning

4th Semester

MS1262 **3-0-0 (3)**

Business Essentials for Engineers

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the basic concepts of management and its functions.
CO-2	Apply the functions of management for taking effective decisions.
CO-3	Analyze factors influencing management in competitive business environment.
CO-4	Identify business opportunities and challenges.
CO-5	Integrate functions of management for building a better organization.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	–	–	–	–	–	–	–	–	3	–	3	3	–	–	–	–
CO-2	–	–	–	–	–	–	–	–	2	–	3	3	–	–	–	–
CO-3	–	–	–	–	–	–	–	–	1	–	3	3	–	–	–	–
CO-4	–	–	–	–	–	–	–	–	1	–	2	2	–	–	–	–
CO-5	–	–	–	–	–	–	–	–	1	–	1	2	–	–	–	–

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction: The evolution of management theory, Business functions and their roles, Organizations and types, Levels of management, Types of markets and pillars of management- planning, organizing, leading and controlling.

People Management: Catalysts for organizational performance – Motivation & Leadership, Organization culture & Change, Human resource functions in a dynamic business environment and evolving dynamics in Industrial Relations.

Marketing Management: Nature and scope of marketing, Company's orientation towards market place, Importance of marketing concept, Marketing environment, 4p's of marketing, market segmentation, target market selection and positioning.

Financial Management: Financial accounting, Financial statements and analysis for decisions, Financial planning, Capital, Working capital, Capital structure and Sources of corporate finance, Investment decisions.

Project Management: Project screening and Selection, Techniques, Structuring concepts and Tools (WBS, OBS, and LRC, RACE). Project life cycle analysis. Appraisal of a project, Project Planning: Techniques, CPM, PERT- GAN - Time Cost Trade-off and Crashing Procedure, Project Monitoring: Monitoring Techniques and time control System, EVA Analysis

Quality & Strategy: Quality, Principles, Quality Awards, Standards of Quality culture, Quality metrics programs, Strategy, Vision and Mission, Porter's 5–forces, McKinsey's 7S Model, BCG Matrix, Competitive advantage - Value chain analysis & Resource based view.

Learning Resources:

Text Books:

1. Ronald J. Ebert, Ricky W. Griffin, Business Essentials, Pearson, 2019, 12th Edition
2. Harold Koontz, Heinz Weihrich, Mark V. Cannice, Essentials of Management, McGraw hill, 2020, 11th Edition.

Reference Books:

1. G. Shainesh Philip Kotler, Kevin lane Keller, Alexander Chernev, Jagdish N. Sheth, Marketing Management, Pearson, 2022, 16th Edition
2. Dessler, G., & Varkkey, B, Human Resource Management, Pearson Education, 2024, 17th Edition
3. Prasanna Chandra, Financial Management: Theory & Practice, Mc Graw Hill, 2022, 11th Edition
4. Poornima M Charantimath, Total Quality Management, Pearson, 2022, 4th Edition
5. IM Pandey, Financial Management, Vikas Publications, 2021, 12th Edition
6. Jack R. Meredith, Mantel, Project Management - A Managerial Approach, John Wiley, 2021,11th Edition

Other Suggested Readings:

1. https://nptel.ac.in/courses/110106050
2. https://nptel.ac.in/courses/110105146
3. https://nptel.ac.in/courses/110105069
4. https://nptel.ac.in/courses/110104068
5. https://ocw.mit.edu/courses/15-535-business-analysis-using-financial-statements-spring-2003/
6. https://ocw.mit.edu/courses/15-810-marketing-management-fall-2010/

CS1202 **3-0-2 (4)**

Compiler Design

Pre-Requisites: **CS1201**

Course Outcomes:

CO-1	Understand phases in the design of compiler.
CO-2	Design top-down and bottom-up parsers.
CO-3	Identify synthesized and inherited attributes.
CO-4	Develop syntax directed translation schemes.
CO-5	Develop algorithms to generate code for a target machine.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	-	-	-	-	-	-	-	-	-	-	-	-	1	2	-	-
CO-2	1	-	-	3	-	-	-	-	-	-	-	-	1	-	-	-
CO-3	2	2	2	-	-	2	-	-	-	-	-	-	1	-	-	-
CO-4	2	1	-	-	2	-	-	-	-	-	-	-	2	-	-	-
CO-5	1	-	-	1	-	-	-	-	-	-	-	-	3	-	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction: The structure of a compiler; The evolution of programming languages; The science of building a compiler; Application of compiler technology; Programming language basics.

A Simple Syntax-Directed Translator: Introduction; Syntax definition; Syntax-directed translation; Parsing; A translator for simple expressions; Lexical analysis; Symbol tables; Intermediate code generation.

Lexical Analysis: The role of the lexical analyzer; Input buffering; Specification of tokens; Recognition of tokens; The lexical-analyzer generator Lex; Finite automata; From regular expressions to automata; Design of a lexical-analyzer generator; Optimization of DFA-based pattern matchers.

Syntax Analysis: Introduction; Context-free grammars; Writing a grammar; Top-down parsing; bottom-up parsing; Introduction to LR parsing: simple LR; More powerful LR parsers; Using ambiguous grammars; Parser generators.

Syntax-Directed Translation: Syntax-directed definitions; Evaluation orders of SDD's; Applications of syntax-directed translation; Syntax-directed translation schemes; Implementing l-attributed SDD's.

Intermediate-Code Generation: Variants of syntax trees; Three-address code; Types and declarations; Translation of expressions; Type checking; Control flow; Backpatching; Switch-statements; Intermediate code for procedures.

Run-time environments: Storage organization; Stack allocation of space; Access to nonlocal data on the stack; Heap management; Introduction to garbage collection; Introduction to trace-based collection; Short-pause garbage collection; Advanced topics in garbage collection.

Code Generation: Issues in the design of a code generator; The target language; Addresses in the target code; Basic blocks and flow graphs; Optimization of basic blocks; A simple code generator; Peephole optimization; Register allocation and assignment; Instruction selection by tree rewriting; Optimal code generation for expressions; Dynamic programming code-generation.

Learning Resources:

Text Books:

1. Alfred V. Aho, Monical S.Lam, Ravi Sethi, and Jeffrey D. Ullman "*Compilers - Principles, Techniques and Tools"*, 2nd Edition, Pearson, 2007.

Reference Books:

1. Ken Kennedy and Randy Allen, Optimizing Compilers for Modern Architectures: A Dependence-Based Approach, Morgan Kaufmann Publishers, 2001.

Other Suggested Readings:

1. NPTEL Course on Compiler Design by Prof. Y. N. Srikanth, IISc. Bangalore, Prof. Sanjeev K. Aggarwal, IIT Kanpur, Prof. Santanu Chattopadhyay, IIT Kharagpur.

CS1204 **3-0-2 (4)**

Software Engineering

Pre-Requisites: **None**

Course Outcomes:

CO-1	Comprehend software development life cycle.
CO-2	Prepare SRS document for a project.
CO-3	Apply software design and development techniques.
CO-4	Identify verification and validation methods in a software engineering project.
CO-5	Implement testing methods for software.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	2	1	1	2	1	-	2	2	2	-	1	2	1	2
CO-2	2	1	2	-	-	2	2	-	2	2	-	-	-	1	-	2
CO-3	2	2	2	1	-	1	1	-	-	2	1	-	2	2	1	2
CO-4	2	2	2	-	2	2	2	-	2	2	1	-	-	1	-	3
CO-5	2	2	2	-	2	2	2	-	2	2	1	-	1	2	1	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

The Software Problem - Cost, Schedule, and Quality, Scale and Change ; Software Processes- Process and Project , Component Software Processes; Software Development Process Models – Waterfall Model, Prototyping , Iterative Development , Rational Unified Process , Time boxing Model , Extreme Programming and Agile Processes , Using Process Models in a Project. Software Requirements Analysis and Specification - Value of a Good SRS , Requirement Process , Requirements Specification.

Formal Specification- Formal Specification in the Software process, Sub-system interface specification, Behavioural Specification; Desirable Characteristics of an SRS - Components of an SRS, Structure of a Requirements Document; Functional Specification with Use Cases - Basics , Examples , Extensions, Developing Use Cases; Other Approaches for Analysis - Data Flow Diagrams , ER Diagrams , Validation.

Software Architecture - Role of Software Architecture, Architecture Views – Component and Connector View - Components, Connectors, An Example. Architecture Styles for C&C View - Pipe and Filter, Shared-Data Style , Client-Server Style, Some Other Styles, Documenting Architecture Design - Evaluating Architectures; Design - Design Concepts - Coupling , Cohesion , The Open-Closed Principle . Function-Oriented Design (from Pressman) - Structure Charts, Structured Design Methodology, An Example. Object- Oriented Design (from Jalote)- OO Concepts, Unified Modeling Language (UML) , A Design Methodology , Examples; Detailed Design - Logic/Algorithm Design, State Modeling of Classes; Verification - Metrics - Complexity Metrics for Function-Oriented Design, Complexity Metrics for OO Design; Coding and Unit Testing -Programming Principles and Guidelines - Structured Programming , Information Hiding, Some Programming Practices, Coding Standards; Incrementally Developing Code - An Incremental Coding Process ,Test-Driven Development, Pair Programming; Managing Evolving Code - Source Code Control and Build, Refactoring; Unit Testing -Testing Procedural Units, Unit Testing of Classes; Code Inspection - Planning, Self-Review, Group Review Meeting; Metrics - Size Measures, Complexity Metrics; Testing - Testing Concepts - Error, Fault, and Failure, Test Case, Test Suite, and Test Harness, Psychology of Testing , Levels of Testing.

Testing Process - Test Plan, Test Case Design, Test Case Execution; Black-Box Testing – Equivalence Class Partitioning, Boundary Value Analysis , Pairwise Testing, Special Cases, State-Based Testing; White-Box Testing - Control Flow-Based Criteria, Test Case Generation and Tool Support; Metrics -Coverage Analysis, Reliability, Defect Removal Efficiency.

Learning Resources:

Text Books:

1. Pankaj Jalote, "*Software Engineering Precise Approach*" , Wiley Publishers, 2012.
2. Rajib Mall, Fundamentals of Software Engineering, PHI, Fifth Edition , 2018.

Reference Books:

1. Ian Sommerville, "*Software Engineering", 8/e* Pearson Publishers, 2012.
2. Roger Pressman, "*Software Engineering*", 5th edition, MCgrawHill, 2002.

CS1206 **3-0-0 (3)**

Database Management Systems

Pre-Requisites: **CS1106**

Course Outcomes:

CO-1	Understand functional components of the DBMS.
CO-2	Devise queries using Relational Algebra, Relational Calculus and SQL
CO-3	Design database schema.
CO-4	Develop E-R model.
CO-5	Evaluate and optimize queries.
CO-6	Analyze transaction processing, concurrency control and recovery techniques.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	1	1	–	–	–	–	–	–	–	–	–	–	–	–	1	1
CO-2	2	1	–	–	2	1	–	1	–	2	2	1	–	1	–	–
CO-3	1	2	3	2	2	1	2	1	–	2	2	–	1	2	2	2
CO-4	1	3	3	2	2	1	2	1	–	2	2	–	1	2	2	2
CO-5	2	–	–	1	–	–	1	–	–	1	–	–	1	2	–	–
CO-6	–	1	2	1	1	–	1	1	–	1	2	–	–	2	–	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to DBMS: Historical perspective, File Versus a DBMS, Advantages of DBMS, Describing and storing data in DBMS, Architecture of a DBMS, Different Data Models.

Entity Relationship (ER) model: Features of ER model, conceptual design using ER model, design for large enterprises; Relational model–structure and operations, Integrity constraints over relations.

Query languages: Relational Algebra, Relational Calculus and SQL– Queries, Constraints, Form of SQL query, UNION, INTERSECT and EXCEPT, Nested queries, Aggregate Operators, Null values, Complex Integrity constraints in SQL, triggers and Embedded SQL.

Database Design: Mapping ER model to Relational form; Functional Dependency–Closer of functional dependencies, closer of attributes, canonical cover and Properties of Decompositions; Normalization process – 1NF, 2NF, 3NF and BCNF; Multivalued dependency– Closer properties of Multivalued dependency and 4NF; Join dependency– PJNF, Decomposition Algorithms.

Transaction Management: ACID properties, transactions, schedules and concurrent execution of transactions; Concurrency control – lock-based protocol, Serializability, recoverability, dealing with deadlocks and Concurrency control without locking.

Query Processing: Overview of Query Evaluation, operator evaluation; Algorithms for relational operations– Selection operation, General selection condition, Projection operation, Join operation, set operation and aggregate operation, Evaluation of relational operations; Query optimization: Alternative plans, functions of query optimizer, translating SQL queries into relational algebra, estimating the cost of a plan, relational algebra equivalences, and other approaches to query optimization.

Database Recovery: Failure classification, Recovery and atomicity, Log-based recovery shadow paging and Advanced Recovery Techniques.

Learning Resources:

Text Books:

1. Silberschatz, Korth and Sudarshan, "Database System Concepts", McGraw Hill, 2021, 7th Edition.
2. Elamsri and Navathe"Fundamentals of Database Systems", Pearson Education, 2017,7th Edition.

Reference Books:

1. Raghu Ramakrishnan, Johannes Gehrke, "Database Management Systems", McGraw Hill, 2002, 3rd Edition.

Other Suggested Readings:

1. MIT Open Course Ware: https://ocw.mit.edu/courses/6-830-database-systems-fall-2010/

CS1208 **3-0-0 (3)**

Operating Systems

Pre-Requisites: **CS1106, EC1263**

Course Outcomes:

CO-1	Distinguish functional architectures of operating systems and file systems.
CO-2	Develop algorithms for subsystem components.
CO-3	Design device drivers and multi-threading libraries for an OS.
CO-4	Develop application programs using UNIX system calls.
CO-5	Design and solve synchronization problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	2	3	2	-	2	-	2	2	2	2	2	3	-	2
CO-2	2	2	3	3	2	-	2	-	2	2	2	2	2	3	-	2
CO-3	2	2	2	3	2	-	2	-	2	2	2	2	2	3	-	2
CO-4	3	2	2	3	2	-	2	-	2	2	2	2	1	3	-	-
CO-5	2	2	2	3	1	-	2	-	2	2	2	2	2	3	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction: Batch, iterative, time sharing, multiprocessor, distributed, cluster and real-time systems, UNIX system introduction and commands.

Operating system structures: Computer system structure, Network structure, I/O Structure, Storage Structure, Dual mode operation, System components, Operating-System Services, System Calls, System Programs, System structure, Virtual Machines, System Design and Implementation, System Generation.

Processes and Threads : Process Concept, Process Scheduling, Operations on Processes, Cooperating Processes, Interprocess Communication, Communication in Client – Server Systems, Multithreading Models, Threading Issues, Pthreads Basic Concepts.

CPU Scheduling: Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Algorithm Evaluation, Process Scheduling Models.

Process Synchronization: Synchronization Background, the Critical-Section Problem, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Critical Regions, Monitors, OS Synchronization.

Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

Memory Management: Memory Management Background, Swapping, Contiguous Memory Allocation, Paging, Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Process Creation, Page Replacement, Allocation of Frames, Thrashing, Operating-System Examples, Other Considerations.

File System: File Concept, Access Methods, Directory Structure, File-System Mounting, File Sharing, Protection File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, Log-Structured File System, NFS.

I/O Systems: Hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O to Hardware Operations, STREAMS, Performance, Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management, RAID Structure, Disk Attachment, Stable-Storage Implementation,

Tertiary-Storage Structure

.Learning Resources:

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Principles", Wiley, 10th Edition, 2019.
2. Richard Stevens, Stephen Rago, "Advanced Programming in the UNIX Environment", Pearson Education, 2nd Edition, 2013.

CS1210 **0-1-2 (2)**

Database Systems Laboratory

Pre-Requisites: **None**

Course Outcomes:

CO-1	Design and Implement a database schema.
CO-2	Devise queries using DDL, DML, DCL and TCL commands.
CO-3	Develop application programs using PL/SQL.
CO-4	Design and implement a project using embedded SQL and GUI.
CO-5	Apply modified components for performance tuning in open source software.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	-	-	3	2	1	-	1	-	-	1	2	2	2	3	2	2
CO-2	2	-	2	1	2	-	2	1	1	1	1	-	-	1	1	-
CO-3	2	-	2	1	2	-	2	1	1	1	1	3	3	2	-	3
CO-4	-	1	2	1	2	-	1	-	-	2	2	2	2	2	1	2
CO-5	1	2	1	1		-	-	1	-	1	1	2	2	1	-	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Familiarization of Oracle RDBMS, SQL*Plus and Oracle developer,
SQL: query-structure; DDL-create, alter, drop, rename and Truncate; DML-select, insert, update, delete and lock; Set operations- union, intersection and except; join; Aggregate Operations- group-by and having; nested sub-queries and views; DCL-grant and revoke, TCL-Commit, save point, rollback and set transaction.

PL/SQL: Environment, block structure, variables, operators, data types, control structures; Cursors structures- Implicit and Explicit; Bulk statements- Bulk collect into and forall; Exception handling Compilation and Run-time, user-defined; Stored procedures- creation options, pass-by-value and functions-pass-by-value; Packages-package specification, body, package creation and usage; Triggers Data definition language triggers, Data manipulation triggers, Compound triggers and trigger restrictions;

Large objects: CLOB, NCLOB, BLOB and BFILE; Implementation of applications using GUI; group project;

Learning Resources:

Text Books:

1. James, Paul and Weinberg, Andy Oppel, "SQL: The Complete Reference", McGraw Hill, 2011, 3rd Edition.

CS1212 **0-1-2 (2)**

Operating Systems Laboratory

Pre-Requisites: **None**

Course Outcomes:

CO-1	Implement elementary UNIX system commands.
CO-2	Develop programs to test synchronization problems.
CO-3	Design and develop user level thread library.
CO-4	Design and implement file system.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	2	2	2	-	2	-	2	2	2	2	-	2	-	-
CO-2	2	2	2	3	3	-	2	-	2	2	2	2	2	3	-	2
CO-3	2	2	2	3	3	-	2	-	2	2	2	2	2	3	-	2
CO-4	2	2	2	3	3	-	2	-	2	2	2	2	2	3	-	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

1. Write Command Interpreter Programs which accepts some basic Unix commands and displays the appropriate result. Each student should write programs for at least six commands.
2. Study the concept of Signals and write a program for Context Switching between two processes using alarm signals.
3. Study pthreads and implement the following: Write a program which shows the performance improvement in using threads as compared with process. (Examples like Matrix Multiplication, Hyper quicksort, Merge sort, Traveling Sales Person problem)
4. Create your own thread library, which has the features of pthread library by using appropriate system calls (UContext related calls). Containing functionality for creation, termination of threads with simple round robin scheduling algorithm and synchronization features.
5. Implement all CPU Scheduling Algorithms using your thread library
6. Study the concept of Synchronization and implement the classical synchronization problems using Semaphores, Message queues and shared memory (minimum of 3 problems)
7. A complete file system implementation inside a disk image file.

Learning Resources:

Text Books:

1. Richard Stevens, Stephen Rago, "Advanced Programming in the UNIX Environment", Pearson Education, Second Edition, 2013.

5th Semester

CS1301 **4-0-0 (4)**

Data Warehousing and Data Mining

Pre-Requisites: **CS1206**

Course Outcomes:

CO-1	Understand stages in building a Data Warehouse.
CO-2	Apply preprocessing techniques for data cleansing.
CO-3	Analyze multi-dimensional modeling techniques.
CO-4	Analyze and evaluate performance of algorithms for Association Rules.
CO-5	Analyze Classification and Clustering algorithms.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	-	-	1	1	1	1	–	–	–	–	–	1	–	1	3	1
CO-2	2	2	3	3	3	1	–	2	–	–	–	1	2	1	2	-
CO-3	1	1	3	3	3	1	1	1	–	–	1	1	3	1	2	1
CO-4	1	1	3	3	3	1	1	2	–	–	1	1	3	2	3	1
CO-5	1	1	3	3	3	1	1	2	–	–	1	1	3	2	3	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Data Warehousing: KDD Process, Introduction to Data Warehouse, Data Preprocessing- Data Cleaning methods, Descriptive Data Summarization, Data Reduction, Data Discretization and Concept hierarchy generation, Overview of ETL and OLAP OLTP integration – comparison of OLAP with OLTP systems, ROLAP, MOLAP and DOLAP, Data Cube Computation methods, Advanced SQL support for OLAP, Multi-dimensional Modeling, Attribute-oriented Induction, Data Warehouse architecture and implementation - Parallel execution, Materialized views.

Data Mining: Introduction, Basic concepts of Association Rule Mining, Frequent Item set mining, Mining various kinds of association rules, Classification by decision tree induction, Bayesian Classification, Rule-based Classification, Classification Back-propagation, Associative Classification, Lazy Learners. Clustering methods, Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Measuring Data Similarity and Dissimilarity Partition based Clustering, Hierarchical based clustering, Density based clustering.

Learning Resources:

Text Books:

1. Jiawei Han and M Kamber, Data Mining Concepts and techniques, Morgan Kaufmann Publishers In, 2022, Fourth Edition.

Reference Books:

1. Pang-Ning Tan, Michael Steinbach, Anuj Karpatne and Vipin Kumar, Introduction to Data Mining, Pearson, 2018, Second Edition.
2. Alex Berson, Stephen Smith, Data Warehousing, Data Mining & OLAP, McGrawHill, 2017, First Edition.

CS1303 **3-0-2 (4)**

Full Stack Development

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand, analyze and build dynamic and interactive web sites.
CO-2	Understand current and evolving Web languages for integrating media and user interaction in both front end and back end elements of a Web site.
CO-3	Analysis and reporting of web data using web analytics.
CO-4	Applying different testing and debugging techniques and analyzing the web site effectiveness.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	-	-	2	1	2	1	-	-	-	2	2	1	2	3	1	1
CO-2	1	1	-	-	-	-	2	1	1	-	1	1	-	2	1	-
CO-3	1	1	-	2	3	-	-	1	-	1	1	-	2	1	1	-
CO-4	1	-	-	1	-	-	-	1	-	1	-	-	-	2	3	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

HTML: Introduction to HTML, What is HTML, HTML Documents, Basic structure of an HTML document, creating an HTML document, Mark up Tags, Heading-Paragraphs, Line Breaks, HTML Tags. Elements of HTML: Introduction to elements of HTML, Working with Text, Working with Lists, Tables and Frames, Working with Hyperlinks, Images and Multimedia, Working with Forms and controls.

Java Script: Introduction to JavaScript, Basic Syntax, Control Structures, Writing Functions, Working with Arrays, The Document Object Model, Events Handling, Client-side Validation, Form Validation & RegExps, ASP, Perl CGI, & Form Methods, SSI & Cookies, Frames & Windows, mimeTypes, plugins, & Java.

PHP: PHP installation and Introduction, Loops String Functions in PHP, PHP Email Function, PHP Basics, Variables Arrays in PHP with Attributes Date & Time, Image, Uploading File handling in PHP Functions in PHP, Errors handling in PHP.

Python: Introduction to Python, Python basics, Data Types and variables Operators, Looping & Control Structure List, Modules Dictionaries, String Regular Expressions, Functions and Functional Programming, Object Oriented Linux Scripting Environment – Classes, Objects and OOPS concepts, File and Directory Access, Permissions and Controls Socket, Libraries and Functionality Programming, Servers and Clients Web Servers and Client scripting, Exploit Development techniques, Writing plugins in Python, Exploit analysis, Automation Process, Debugging basics, Task Automation with Python
Application using Node JS and MongoDB: Introduction to Node.js; Events; Streams; Modules; Express; Socket.io; Persisting Data. Introduction to MongoDB, Accessing MongoDB from Node JS.

Learning Resources:

Text Books:

1. Deitel, Deitel and Nieto, "Internet and Worldwide Web - How to Program", 5th Edition, PHI, 2011.
2. Bai and Ekedhi, "The Web Warrior Guide to Web Programming", 3rd Edition, Thomson, 2008.

Reference Books:

1. Brad Dayley, Brendan Dayley, and Caleb Dayley , Node.Js, Mongodb and Angular Web Development: The Definitive Guide to Using the Mean Stack to Build Web Applications, 2nd Edition, –Pearson Education, 2018
2. Robertw. Sebesta, Programming World Wide Web, 8th Edition –Pearson Education, 2020.

CS1305 **3-0-0 (3)**

Computer Networks

Pre-Requisites: **CS1208**

Course Outcomes:

CO-1	Understand OSI and TCP/IP models.
CO-2	Analyze MAC layer protocols and LAN technologies.
CO-3	Design applications using internet protocols.
CO-4	Implement routing and congestion control algorithms.
CO-5	Develop application layer protocols.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-
CO-2	-	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-
CO-3	3	2	3	-	3	-	-	-	-	-	-	2	2	2	-	2
CO-4	1	2	2	1	-	-	-	-	-	-	-	-	2	2	-	-
CO-5	3	3	3	-	3	-	1	1	-	-	-	2	3	3	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction – network architecture - protocol implementation issues - network design. Reference models- The OSI Reference Model- the TCP/IP Model - A Comparison of the OSI and TCP/IP Models.

Datalink Layer-Ethernet, Token ring, wireless LANs-Issues with data link Protocols-Encoding framing and error detection and correction-sliding window Protocol-Medium access control.

Network layer – network layer design issues - Routing algorithms - Congestion control algorithms – Internetworking - The network layer in the internet - Internet Protocol (IP) - Unicast, multicast, and inter domain routing.

Transport layer - Elements of transport protocol - Congestion control – The Internet's Transmission Control Protocol (TCP) - Remote Procedure Call (RPC) – Implementation semantics of RPC – BSD sockets - client-server applications.

Application layer - Domain name server – Simple Mail Transfer Protocol – File Transfer Protocol - World wide web - Hypertext transfer protocol -Presentation formatting and data compression- Introduction to Network security - Web Services architectures for developing new application protocols.

.

Learning Resources:

Text Books:

1. Larry L Peterson, Bruce S Davis, "*Computer Networks"*, 5th Edition, Elsevier, 2012.
2. Andrew S. Tanenbaum, David J Wetherall, "*Computer Networks"*, 5th Edition, Pearson Edu, 2010.

CS1307 **3-0-0 (3)**

Distributed Computing

Pre-Requisites: **None**

Course Outcomes:

CO-1	Identify models of distributed computing.
CO-2	Analyze algorithms for coordination, communication, and synchronization in distributed systems.
CO-3	Design and implement distributed, fault-tolerant storage systems.
CO-4	Design and Implement distributed analytics using MapReduce.
CO-5	Design distributed algorithms for handling deadlocks.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	-	-	-	-	-	-	-	-	-	-	3	2	-	-
CO-2	2	1	2	2	-	-	-	-	-	-	-	-	3	3	-	1
CO-3	2	1	2	1	2	-	-	-	-	-	-	-	2	2	-	1
CO-4	2	1	2	1	2	-	-	-	-	-	-	-	3	3	-	1
CO-5	1	-	2	1	1	-	-	-	-	-	-	-	3	3	-	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction: Types of distributed systems, synchronous vs. asynchronous execution, design issues and challenges, model of distributed computation.

Logical Time: Logical clocks, scalar time, vector time, efficient implementation of vector clocks, virtual time.

Message ordering and group communication: Message ordering paradigms, group communication, causal order (CO), total order, propagation trees for multicast, publish-subscribe model.

Global state and snapshot recording algorithms: Chandy-Lamport algorithm and its variations, snapshot algorithms for non-FIFO channels, snapshots in a causal delivery system, monitoring global state, necessary and sufficient conditions for consistent global snapshots, finding consistent global snapshots in a distributed computation.

Distributed mutual exclusion algorithms: Lamport's algorithm, Ricart–Agrawala algorithm, Singhal's dynamic information-structure algorithm, Maekawa's algorithm, Agarwal–El Abbadi quorum-based algorithm, Raymond's tree-based algorithm.

Deadlock detection in distributed systems: System model, Knapp's classification of distributed deadlock detection algorithms, Mitchell and Merritt's algorithm for the single resource model, Chandy– Misra–Haas algorithm for the AND and the OR model, Termination detection algorithm.

Distributed Data Storage and Computation: Memory consistency models, replication, CAP theorem, distributed shared memory, distributed file systems, distributed transactions, eventual consistency, conflict-free replicated data types, failures, byzantine fault-tolerance, consensus protocols, distributed data analytics using MapReduce.

Learning Resources:

Text books

1. Ajay D. Kshemakalyani and Mukesh Singhal, Distributed Computing, Cambridge University Press, 2008.
2. Maarten Van Steen and Andrew S. Tanenbaum, Distributed Systems - Principles and Paradigms, 4th edition, online: https://www.distributed-systems.net/index.php/books/ds4/, 2024.

CS1381 **1-0-0 (0.5)**

Fractal Course I

Pre-Requisites: None

Course Outcomes:

CO-1	Acquire an in-depth understanding of the specific topic covered in the course, which could range from a particular technology, method, or recent advancement in Computer Science and Engineering.
CO-2	Develop hands-on skills relevant to the course topic, such as using specific tools, software, or techniques.
CO-3	Apply learned concepts to solve focused and practical engineering problems related to the course content.
CO-4	Understand the interdisciplinary aspects and applications of the course to solve real-world industrial problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	3	-	-	-	-	-	-	-	-	-	3	2	2	2
CO-2	-	-	-	2	2	3	3	-	-	-	-	-	2	3	3	2
CO-3	-	-	3	2	2	-	-	-	-	-	-	-	3	3	1	2
CO-4	-	-	-	-	-	-	-	-	1	3	-	2	3	2	1	2

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

This one-week course aims to bridge the gap between academic learning and industry application / latest research developments, providing students with a comprehensive understanding of the in Computer Science and Engineering field and enhancing their readiness for professional careers.

a. Structure: Lectures delivered by an expert from the Computer Science and Engineering Industry / R&D Organization / Academic Institution (SPARK Collaborators / Foreign Subject Experts in GIAN program / Adjunct Professors from Foreign Universities @ NITW).

b. Content: Topics covering current practices, case studies, technological advancements, and future trends.

c. Interactive Sessions: Q&A sessions, discussions, and case study analyses to foster interaction between students and the subject expert.

d. Assessment: the mode of assessment (Participation, a short reflective report / a presentation summarizing key takeaways from the lectures / Objective or descriptive type exam, etc.) will be decided by the subject expert

Learning Resources:

1. Course material and any learning resources suggested by the experts.

CS1309 **0-1-2 (2)**

Computer Networks Laboratory

Pre-Requisites: **None**

Course Outcomes:

CO-1	Develop programs for client-server applications.
CO-2	Perform packet sniffing and analyze packets in network traffic.
CO-3	Implement error detecting and correcting codes.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	2	3	-	2	-	2	-	-	-	-	2	2	2	-	-
CO-2	2	3	2	2	1	1	-	2	-	-	-	1	1	2	-	2
CO-3	1	1	1	2	-	-	-	-	-	-	-	-	2	-	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

1. Programs to implement error detection and correction
2. Client-Server applications using inter process communication mechanisms a) FIFO b) Message queues c) Shared memory
3. Connection-oriented Client-Server applications based on BSD sockets
4. Connectionless Client-Server applications
5. Implementation of Chat servers and mail Servers
6. Implementation of routing algorithms
7. Programs using Remote Procedure Call (RPC)
8. Client-Server applications based on Raw Sockets, IP Spoofing
9. Implementation of application layer protocols
10. Datalink layer Access, Packet Sniffing

Learning Resources:

Text Books:

1. W. Richard Stevens, "*UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI*", Prentice Hall, 1998
2. W. Richard Stevens, "*UNIX Network Programming, Volume 2, Second Edition: Interprocess Communications*", Prentice Hall, 1999

Reference Books:

1. W. Richard Stevens, Stephen Rago, "*Advanced Programming in the UNIX Environment*", Pearson Education, *Second Edition.*

CS1311 **0-1-2 (2)**

Knowledge Engineering Laboratory

Pre-Requisites: **None**

Course Outcomes:

CO-1	Build data cubes with SQL.
CO-2	Implement data preprocessing techniques on data.
CO-3	Implement OLAP operations and multi-dimensional modeling.
CO-4	Implement data mining algorithms.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	2	1	1	2	1	-	2	2	2	-	3	2	3	2
CO-2	2	1	2	-	-	2	2	-	2	2	-	-	2	-	3	2
CO-3	2	2	2	1	-	1	1	-	-	2	1	-	2	-	3	2
CO-4	2	2	2	-	2	2	2	-	2	2	1	-	3	2	3	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

1. Advanced SQL Analytic functions
2. Implementation of OLAP operations
3. Data preprocessing techniques
4. Cube computation methods
5. Concept hierarchy method
6. Write a program in any programming language to generate at least 10,000 transactions in a text file with at least three items.
7. Write a program to implement the *APRIORI* algorithm and test it thoroughly.
8. Write a program for each of the following to improve *APRIORI*
   A. Hash based Technique. B. Dynamic Item set Counting Algorithm. C. Partition Based Approach.
9. Write a program for *FPGROWTH* algorithm and test it.
10. Write a program to construct an optimized DECISION TREE for a given training data and by using any attribute selection measure.
11. Write a program for NAÏVE BAYESIAN algorithm for classifying the data.
12. Implement the K-Means Clustering algorithm for clustering the given data.

Learning Resources:

Text Books:

1. Jiawei Han and M Kamber, "*Data Mining Concepts and techniques"*, Third Edition, Elsevier Publications, 2011.

6th Semester

CS1302 **4-0-0 (4)**

Cryptography and Network Security

Pre-Requisites: **CS1305**

Course Outcomes:

CO-1	Understand encryption algorithms and their significance in modern systems..
CO-2	Understanding of number theory required for cryptographic algorithms.
CO-3	Understand cryptographic algorithms to build secure protocols.
CO-4	Design of secure protocols to solve real-world scenarios.
CO-5	Apply foot printing, scanning, enumeration, and similar techniques to discover network and system vulnerabilities.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	1	–	–	–	–	–	–	–	2	3	–	–	–	–	-	–
CO-2	–	–	–	3	–	–	–	–	–	–	2	–	1	–	–	–
CO-3	–	1	–	–	2	–	–	–	–	–	–	–	–	3	–	–
CO-4	–	–	1	–	–	–	–	2	–	-	–	–	–	–	–	3
CO-5	–	–	–	–	–	1	2	–	–	–	–	–	–	–	3	–

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Basic Cryptography- Security Services, Security Attacks, Basic cryptography methods, Classical Encryption Techniques, Substitution Ciphers, Transposition Ciphers, time pad, Cryptanalysis Security attacks, Symmetric cryptography algorithms- DES, 3DES, AES, Public-key cryptography.

Number Theory- Introduction to number theory – Modular Arithmetic; Finite fields; Number theory properties – Primality testing; Fetmat's and Euler's theorem; Chinese remainder theorem; Integer factorization; discrete logarithm.

Public Key Cryptography: RSA, ECC, Generalized ElGamal Public Key Cryptosystem, Key Management and Distribution: Symmetric Key Distribution, Diffie Hellman key exchange, Distribution of Public Keys, X.509 Certificates Functionalities of entity, content authentication; message digests and hashing schemes; Key management and Distribution-Certificate authorities; PKI; MAC; Hashing; Digital Signatures-Authentication protocols; Digital Signature Standard.

Network Security: IP Security, Transport-Level Security, Electronic Mail Security, Firewalls- Firewall Properties; Design of firewalls; VPN's; Filtering; Intrusion detection system

Learning Resources:

Text Books:

1. William Stallings, Cryptography and Network Security: Principles and Practice, 7th Edition, Pearson, 2017.
2. Dan Boneh and Victor Shoup, A Graduate Course in Applied Cryptography, Stanford University 4th edition, 2017.

Reference Books:

1. Jonathan Katz, Yehuda Lindell, Introduction To Modern Cryptography, Taylor & Francis Ltd, 2020,
2. Douglas R. Stinson Maura B. Paterson, Cryptography Theory and Practice, CRC Press, Fourth edition, 2004.

CS1304 **3-0-0 (3)**

Machine Learning

Pre-Requisites: **CS1106**

Course Outcomes:

CO-1	Understand instance-based learning algorithms.
CO-2	Design neural network to solve classification and function approximation problems.
CO-3	Build optimal classifiers using genetic algorithms.
CO-4	Design convolutional networks to solve classification problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	1	1	2	2	-	1	-	-	1	-	-	2	1	2	2
CO-2	2	1	2	2	2	-	1	-	-	1	-	-	2	2	2	2
CO-3	2	1	1	2	2	-	1	-	-	1	-	-	2	2	2	2
CO-4	2	2	2	2	2	1	2	-	-	1	1	-	2	2	3	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction – Well defined learning problems, Designing a Learning System, Issues in Machine Learning; - The Concept Learning Task - General-to-specific ordering of hypotheses, Find-S, List then eliminate algorithm, Candidate elimination algorithm, Inductive bias - Decision Tree Learning - Decision tree learning algorithm-Inductive bias- Issues in Decision tree learning; - Artificial Neural Networks – Perceptrons, Gradient descent and the Delta rule, Adaline, Multilayer networks, Derivation of backpropagation rule- Backpropagation Algorithm- Convergence, Generalization; – Evaluating Hypotheses – Estimating Hypotheses Accuracy, Basics of sampling Theory, Comparing Learning Algorithms; - Bayesian Learning – Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier, Bayesian belief networks, EM algorithm; - Computational Learning Theory – Sample Complexity for Finite Hypothesis spaces, Sample Complexity for Infinite Hypothesis spaces, The Mistake Bound Model of Learning; - Instance-Based Learning – k-Nearest Neighbour Learning, Locally Weighted Regression, Radial basis function networks, Case-based learning - Genetic Algorithms – an illustrative example, Hypothesis space search, Genetic Programming, Models of Evolution and Learning; Reinforcement Learning - The Learning Task, Q Learning, Support vector Machines, Deep learning networks – Deep Feedforward Networks – Regularization for Deep Learning – Optimization for Training Deep Models – Convolutional Network

.

Learning Resources:

Text Books:

1. Tom.M.Mitchell, "Machine Learning", McGraw Hill International Edition, 1997
2. C Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.

Reference Books:

1. Ian Goodfellow, Yoshua Bengio, AaronCourville, "Deep Learning", The MIT Press Cambridge, Massachusetts, London, England, 2016.

CS1306 **0-1-4 (3)**

Product Development

Pre-Requisites: **CS1102**

Course Outcomes:

CO-1	Comprehend software development life cycle.
CO-2	Prepare SRS document for a project/product.
CO-3	Develop a prototype of the product.
CO-4	Evaluate the entire product and the product based on testing with user.
CO-5	Explore the scope for protecting novelty of the product through patent.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	2	1	1	2	1	-	2	2	2	-	1	2	1	2
CO-2	2	1	2	-	-	2	2	-	2	2	-	-	-	1	-	2
CO-3	-	-	2	3	3	3	2	2	3	3	2	3	2	3	1	1
CO-4	-	-	2	3	3	3	2	2	3	3	2	3	1	-	-	3
CO-5	-	-	2	3	3	3	2	2	3	3	2	3	-	-	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Design Thinking process for Product Development: Review of five step design thinking process of Empathize- Define- Ideate- Prototype- Test.

Project Selection: Identification of the problem through empathy, formulate and ideate to solve the problem.

Product Development:User Interface (UI) and User Experience (UX) Design: Principles of UI d esign, UX design best practices, Choosing a language of programming, creating intuitive interfaces, and optimizing user interaction, Agile methodologies for iterative design, identification of tools, usability testing, Debugging tools and techniques, usage of Simulators.

Gathering feedback, SDLC, collaboration between design and development teams, and integrating design processes with agile development practices,

Iterative improvement of the product and Report writing: Development of assemblies/mock-up models/ working models/ prototypes/functional models/products, Testing and design review, Report writing.

Design Tools and Technologies, Project Management for Product Design, resource allocation, and risk management, privacy concerns, legal aspects and IPR Filing of IPR, implications, Ethics and societal implications of Intellectual Property Rights. , Case studies and real world applications, Business aspects of the product. Emerging product development-AI and IoT based products, Web and mobile based products.

Learning Resources:

Textbooks:

1. Pankaj Jalote, "Software Engineering Precise Approach" , Wiley Publishers, 2012.
2. Naresh Chauhan, “Software Testing- Principles and Practices”, Oxford University Press, Second Edition, 2016.
3. Martin Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, OREILLY, 2017
4. Marty Cagan, Inspired: How to Create Tech Products Customers Love, Wiley, 2018

Other Suggested Reading:

1. Self-Paced Tutorials: https://help.autodesk.com/view/fusion360/ENU/courses/
2. Product Documentation https://help.autodesk.com/view/fusion360/ENU/?guid=GUID1C665B4D-7BF7-4FDF-98B0-AA7EE12B5AC2

CS1382 **1-0-0 (0.5)**

Fractal Course II

Pre-Requisites: None

Course Outcomes:

CO-1	Acquire an in-depth understanding of the specific topic covered in the course, which could range from a particular technology, method, or recent advancement in Computer Science and Engineering.
CO-2	Develop hands-on skills relevant to the course topic, such as using specific tools, software, or techniques.
CO-3	Apply learned concepts to solve focused and practical engineering problems related to the course content.
CO-4	Understand the interdisciplinary aspects and applications of the course to solve real-world industrial problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	3	-	-	-	-	-	-	-	-	-	3	2	2	2
CO-2	-	-	-	2	2	3	3	-	-	-	-	-	2	3	3	2
CO-3	-	-	3	2	2	-	-	-	-	-	-	-	3	3	1	2
CO-4	-	-	-	-	-	-	-	-	1	3	-	2	3	2	1	2

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

This one-week course aims to bridge the gap between academic learning and industry application / latest research developments, providing students with a comprehensive understanding of the in Computer Science and Engineering field and enhancing their readiness for professional careers.

a. Structure: Lectures delivered by an expert from the Computer Science and Engineering Industry / R&D Organization / Academic Institution (SPARK Collaborators / Foreign Subject Experts in GIAN program / Adjunct Professors from Foreign Universities @ NITW).

b. Content: Topics covering current practices, case studies, technological advancements, and future trends.

c. Interactive Sessions: Q&A sessions, discussions, and case study analyses to foster interaction between students and the subject expert.

d. Assessment: the mode of assessment (Participation, a short reflective report / a presentation summarizing key takeaways from the lectures / Objective or descriptive type exam, etc.) will be decided by the subject expert

Learning Resources:

1. Course material and any learning resources suggested by the experts.

CS1308

0-1-2 (2)

Cryptography and Network Security Laboratory

Pre-Requisites: **None**

Course Outcomes:

CO-1	Implement Cryptographic algorithms.
CO-2	Analyze the network traffic and simulate the attacks.
CO-3	Design and implement security protocols.
CO-4	Analyze the protocol using security tools.
CO-5	Understand various vulnerabilities in designed software or hardware.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	–	1	–	–	–	–	–	–	–	3	–	–	–	–	2	–
CO-2	1	–	–	–	–	–	3	–	–	–	–	–	3	–	–	–
CO-3	–	–	–	1	–	–	–	–	3	–	–	2	–	–	–	–
CO-4	–	–	2	–	–	–	–	–	–	–	3	–	–	–	–	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Category: Cryptography experiments

Number theory and Cryptographic experiments - 1) Installation of GMP library 2) Euclidean algorithm for computing the GCD of two integers 3) Extended Euclidean algorithm 4) Modular Arithmetic over Zn 5) Polynomial Arithmetic over GF(2^n) 6) Substitution Technique 7) DES 8) AES 9) Chinese Remainder Theorem 10) RSA 11) Diffie-Hellman Key Exchange 12) Elgamal Cryptographic System 13) Elliptic curve cryptography 14) Elgamal and DSS Digital signature scheme

Cycle 2: Network Security experiments

1. Design and Implement a protocol with the details as given below
   - User A likes to allow to read his encrypted messages by User B without revealing his private key
   - User A generates a new key pair which is shared between User B and Proxy
   - User A delegates a proxy to reencrypt /partial decrypt the encrypted message of User A using new key in the key pair
   - Proxy sends the modified encrypted message to User B
   - User B decrypts the encrypted message using new key in the key pair
   - Use key exchange algorithm between User A and Proxy, User A and User B Encryption and decryption can be using any PKC

2. A network in which the nodes are logically connected using tree structure, a layered encryption and decryption is followed in the protocol. Design and code the protocol with the following details
   - The nodes are divided into left sub tree and right sub tree
   - Among the nodes in the same level one is elected as leader on both left and right subtree separately.
   - There is only one key is allotted at each level
   - The key is shared among all nodes in the same level
   - When the document is to be encrypted the leader collects the shares from other nodes and encryption is done by the leader
   - The encrypted document travels from lower level to upper level through leader in left sub tree and encrypted at each level
   - When the document reaches root, it travels from higher level to lower level
   - Each level the key for decryption is collected by the leader from the nodes in the same level
   - Decryption is done at each level and reaches a destination

3. A network in which each node can act as a client or server for the other computers in the network, allowing shared access to various resources such as files, peripherals, etc without the need for a central server.
    - Design and code security association/agreement between server and client
    - Procedure for key management
    - Design and code for authentication between server and client vice versa

4. Design and implement a communication system with the following details The users are divided logically into groups.
    - Each user can have many public and private key pairs.
    - Each users maintains a table in which public key is stored.
    - Each user maintains private keys in another table along with the id of the public key which is in encrypted form.
    - The encryption and decryption is based on the public key id sent along with the cipher text. Use any public key cryptography for encryption and decryption
5. Design and implement secure communication between two groups A and B:
    - A server is connected to n number of registered users.
    - The users are divided into two groups such as A and B. Members can be added and removed dynamically.
    - The communication is between two group leaders (Assume the leaders are already elected). Each group leader authenticates his members by using any authentication technique before any communication happens.
    - The server generates a common (public key, private key) for each group and divides the private key into shares and dispatches to the users of respective groups.
    - A user from group A can communicate to group B user through the leader and vice versa.
    - The encryption and decryption are using key par (public key, private key), which is the common practice.

6. Implement PGP email security - The design is available in the textbook

7. Implement Kerberos version 4 authentication protocol between server and client - The design is available in the textbook

Category 3: Security tools

1. Introduction to Packet sniffing tools
2. Penetration testing tools
3. Internet security protocols validation tool – AVISPA
4. Network intrusion detection and prevention system - Snort

Learning Resources:

Text Books:

1. A. Menezes, P. Van Oorschot, S. Vanstone, "Handbook of Applied Cryptography", CRC Press, 2004.
2. William Stallings, "Cryptography and Network Security", Pearson Education, 6th Edition, 2014.

Reference Books:

1. GNU-MP Manual

CS1310 **0-1-2 (2)**

Machine Learning Laboratory

Pre-Requisites: **None**

Course Outcomes:

CO-1	Design and implement algorithm using least means square learning rule to play checkers game.
CO-2	Design a classifier using Genetic Algorithm.
CO-3	Apply supervised learning using perceptron.
CO-4	Design and implement Convolutional Networks to solve classification problem.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	1	1	2	2	-	1	-	-	1	-	-	2	1	2	1
CO-2	2	1	2	2	2	-	1	-	-	1	-	-	2	1	2	2
CO-3	2	1	1	2	2	-	1	-	-	1	-	-	1	1	1	1
CO-4	2	2	2	2	2	1	2	-	-	1	1	-	2	2	2	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

1. Design and implement machine learning algorithm using least means square learning rule to play checkers game. The training experience should be generated by the system playing game with itself.
2. Implement a genetic algorithm program to successfully classify examples in the restaurant domain problem. Data for RESTAURANT domain given in this table.

Example	Friday	Patrons	Price	Rain	Res	Est	Type	Will Wait
X1	No	Some	High	No	Yes	0-10	French	Yes
X2	No	Full	Low	No	No	30-60	Thai	No
X3	No	Some	Low	No	No	0-10	Burger	Yes
X4	Yes	Full	Low	Yes	No	10-30	Thai	Yes
X5	Yes	Full	High	No	Yes	>60	French	No
X6	No	Some	Med	Yes	Yes	0-10	Italian	Yes
X7	No	None	Low	Yes	No	0-10	Burger	No
X8	No	Some	Med	Yes	Yes	0-10	Thai	Yes
X9	Yes	Full	Low	Yes	No	>60	Burger	No
X10	Yes	Full	High	No	Yes	10-30	Italian	No
X11	No	None	Low	No	No	0-10	Thai	No
X12	Yes	Full	Low	No	No	30-60	Burger	Yes

Res indicates reservation made and Est means estimated waiting time to get table, WillWait target attribute indicates whether to wait for table in that restaurant.

3. Design a feed forward neural network using back propagation algorithm to to solve and written character recognition problem for A to Z and 0 to 9 letters.
4. Implement perceptron learning algorithm and attempt to solve two input i) AND gate ii) Or Gate iii) EXOR gate problems.
5. Implement Adaline learning algorithm and attempt to solve two input i) AND gate ii) Or Gate iii) EXOR gate problems.
6. Implement a Genetic algorithm to generate solutions for 8-Queens problem.
7. Implement a machine learning program to play 5× 5 Tic tac toe game. The program should use least means square learning rule.
8. Design a Convolutional network for handwritten character recognition problem for A to Z and 0 to 9 digits.

Learning Resources:

Text Books:

1. Tom.M.Mitchell, "Machine Learning", McGraw Hill International Edition, 2017.
2. C Bishop, "Pattern Recognition and Machine Learning ", Springer, 2006.

Reference Books:

1. Ian Goodfellow, Yoshua Bengio, AaronCourville, "Deep Learning", The MIT Press, 2016.

7th Semester

CS1401 **3-0-0 (3)**

High Performance Computing

Pre-Requisites: **CS1208, CS1305**

Course Outcomes:

CO-1	Design and analyze the parallel algorithms for real world problems and implement them on available parallel computer systems.
CO-2	Optimize the performance of a parallel program to suit a particular platform.
CO-3	Design algorithms suited for Multicore processor systems using OpenCL, OpenMP, Threading techniques.
CO-4	Analyze the communication overhead of interconnection networks and modify the algorithms to meet the requirements.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	3	3	2	2	-	-	-	-	-	-	3	2	2	2
CO-2	3	2	3	3	2	2	-	-	-	-	-	-	2	2	-	3
CO-3	3	3	3	2	2	2	-	-	-	-	-	-	3	3	-	2
CO-4	2	3	3	2	2	2	-	-	-	-	-	-	2	3	-	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction: Implicit parallelism, Limitations of memory system performance, control structure, communication model, physical organization, and communication costs of parallel platforms, Routing mechanisms for interconnection networks, mapping techniques.

Parallel algorithm design: Preliminaries, decomposition techniques, tasks and interactions, mapping techniques for load balancing, methods for reducing interaction overheads, parallel algorithm models.

Basic communication operations: Meaning of all-to-all, all-reduce, scatter, and gather, circular shift and splitting routing messages in parts.

Analytical modeling of parallel programs: sources of overhead, performance metrics, the effect of granularity on performance, scalability of parallel systems, minimum execution time, minimum cost-optimal execution time, asymptotic analysis of parallel programs.

Programming using message passing paradigm: Principles, building blocks, MPI, Topologies and embedding, Overlapping communication and computation, collective communication operations, Groups and communicators.

Programming shared address space platforms: Threads, POSIX threads, Synchronization primitives, attributes of threads, mutex and condition variables, Composite synchronization constructs, OpenMP Threading Building blocks; An Overview of Memory Allocators, An overview of Intel Threading building blocks.

Basic parallel algorithms: prefix sums, Tree traversal algorithms, basic operations (insertion deletion and search) on trees, merging, maximum, graph colouring list ranking, Planar geometry and String algorithms.

Dense Matrix Algorithms: matrix vector multiplication, matrix-matrix multiplication, solving system of linear equations.

Sorting: Sorting networks, Bubble sort, Quick sort, Bucket sort and other sorting algorithms.

Graph algorithms: Minimum spanning tree, single source shortest paths, all-pairs shortest paths, Transitive closure, connected components, algorithms for sparse graphs.

Learning Resources:

Text Books:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, "Introduction to Parallel Computing", Second Edition Pearson Education, 2007.
2. Michael J. Quinn, "Parallel Programming in C with MPI and OpenMP", McGraw-Hill International Editions, Computer Science Series, 2004.

Reference Books:

1. Joseph Jaja, "An Introduction to Parallel Algorithms", Addison-Wesley, 1992.
2. S G Akl, The Design and Analysis of Parallel Algorithms, Prentice-Hall, 1989.

CS1403 **0-1-2 (2)**

High Performance Computing Laboratory

Pre-Requisites: **CS1208, CS1305**

Course Outcomes:

CO-1	Implementation of the parallel algorithms for real world problems on available parallel computer systems.
CO-2	Optimize the performance of a parallel program to suit a particular platform.
CO-3	Implementation of algorithms suited for Multicore processor systems using OpenCL, OpenMP, Threading techniques.
CO-4	Parallel Implementations and Analysis of the communication overhead of interconnection networks and modify the algorithms to meet the requirements.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	3	3	2	2	–	–	–	–	–	–	3	2	2	2
CO-2	3	2	3	3	2	2	–	–	–	–	–	–	2	2	-	3
CO-3	3	3	3	2	2	2	–	–	–	–	–	–	3	3	-	2
CO-4	2	3	3	2	2	2	–	–	–	–	–	–	2	3	-	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to OpenMP and MPI, Communication primitives, Multitasking, Parallel programming and debugging, Deadlocks, Performance measurement, Problem decomposition on multiprocessor network, Load Balancing.

Learning Resources:

Text Books:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, "Introduction to Parallel Computing", Second Edition Pearson Education – 2007
2. Michael J. Quinn (2004), "Parallel Programming in C with MPI and OpenMP", McGraw-Hill International Editions, Computer Science Series,

Reference Books:

1. Joseph Jaja, "An Introduction to Parallel Algorithms", Addison-Wesley, 1992

Other Suggested Readings:

1. Web Materials

CS1489 **0-0-0 (2)**

Seminar and Technical Writing

Pre-Requisites: **None**

Course Outcomes:

CO-1	Consolidate ideas based on expert talks attended.
CO-2	Prepare a well-organized report employing elements of critical thinking and technical writing.
CO-3	Demonstrate the ability to describe, interpret and analyze the subject matter and develop competence in presenting.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	–	2	–	–	3	3	–	2	3	–	3	–	–	–	–
CO-2	2	–	2	–	–	3	3	–	2	3	–	3	–	–	–	–
CO-3	2	–	2	–	–	3	3	–	2	3	–	3	–	–	–	–

1 - Slightly; 2 - Moderately; 3 - Substantially

Description:

In Seminar and Technical Writing, every student is expected to prepare a well-organized report based on one / all of the following:

- by attending at least 5 expert lectures/ invited talks/ Seminar/ Popular lectures etc. organized by the institute/any of the departments, ideally in a specific domain or with the same theme.
- prepare a business or marketing plan based on patent search

The student is expected to consolidate the ideas from these lectures/patent searches and may even include material from other sources to strengthen the content of the report. The student should prepare a well-organized report based on the above and present it to the panel constituted by the department, for evaluation.

Evaluation Criteria:

The student will be evaluated by the panel based on the below criteria.

Criteria	Description	Weightages
I	Clarity on the topic	15 Marks
II	List of lectures attended	20 Marks
III	Report	25 Marks
IV	Presentation	25 Marks
V	Response to questions	15 Marks

Evaluation Criteria-CO Mapping

CO / Criteria	CO1	CO2	CO3
I	X		
II	X		
III		X	
IV			X
V			X

CS1495 **0-0-0 (2)**

Minor Project

Pre-Requisites: **None**

Course Outcomes:

CO-1	Apply engineering principles to real-world projects.
CO-2	Plan and monitor project tasks individually or as a team.
CO-3	Demonstrate practical experience in project execution.
CO-4	Communicate project findings clearly through reports and presentations.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	3	3	3	–	–	–	2	1	–	3	–	–	–	–
CO-2	2	2	2	2	2	–	–	–	3	2	3	3	–	–	–	–
CO-3	2	2	2	2	2	–	–	–	3	2	3	3	–	–	–	–
CO-4	1	1	1	1	1	–	–	–	3	3	–	–	–	–	–	–

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Description:

Students are expected to choose real world or relevant problems and apply the engineering principles learned, to solve the problem through building prototypes or simulations or writing codes or establishing processes/synthesis/correlations etc. The department constituted panel can decide the suitability and worthiness of the project

Evaluation Criteria:

The student will be evaluated by the panel based on the below criteria. Weightage for each criterion will be determined by the panel and will be informed to the students.

Criteria	Description	Weightages
I	Identification of Problem Domain	15 Marks
II	Study of Existing Systems and establishing clear objectives	20 Marks
III	Planning of project and work distribution within the team	25 Marks
IV	Proper Documentation and Technical Writing	25 Marks
V	Presentation and Response to questions	15 Marks

Evaluation Criteria-CO Mapping

CO / Criteria	CO1	CO2	CO3	CO4
I	X			
II	X			
III		X	X	
IV				X
V				X

CS1491

0-0-0 (2)

Short Term Industrial / Epics / Research Experience

Pre-Requisites: **None**

Course Outcomes:

CO-1	Apply engineering principles to real-world problems, gaining practical experience.
CO-2	Plan, manage and execute the work with ethical consideration.
CO-3	Review the social and environmental impact of the work.
CO-4	Communicate the learnings through report and presentation.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	3	3	3	–	–	–	–	–	–	3	–	–	–	–
CO-2	2	2	2	2	2	–	–	3	3	–	3	3	–	–	–	–
CO-3	–	–	–	–	–	3	3	–	–	–	–	3	–	–	–	–
CO-4	–	–	–	–	–	–	–	–	–	3	–	3	–	–	–	–

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Description:

Every student has to undergo either a Summer Internship / EPICS / Research project. The summer internship may be undergone in an Industry/Research organization or any premier academic Institution, including NIT Warangal for 6-8 weeks. The EPICS/research project shall be registered under the guidance of any faculty member in the institute. The student is required to submit a report and present the work before an evaluation committee, nominated by the Head of the Department.

Evaluation Criteria:

The student will be evaluated by the panel based on the below criteria. Weightage for each criterion will be determined by the panel and will be informed to the students.

Criteria	Description	Weightages
I	Relevance of the area of work	15 Marks
II	Performance of the Task	25 Marks
III	Crucial learnings from the work	25 Marks
IV	Report Preparation	20 Marks
V	Presentation and Response to questions	15 Marks

Evaluation Criteria-CO Mapping

CO / Criteria	CO1	CO2	CO3	CO4
I	X			
II		X		
III			X	
IV				X
V				X

8th Semester

CS1498

0-0-0 (6)

Major Project

Pre-Requisites: **None**

Course Outcomes:

CO-1	Identify a domain specific and contemporary topic.
CO-2	Review literature to identify gaps and define objectives & scope of the work.
CO-3	Develop a prototype/model, experimental set-up or software systems to meet the objectives.
CO-4	Analyze the results to draw valid conclusions.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	2	–	–	2	1	–	1	3	1	1	2	3	–	2	–
CO-2	2	2	–	–	1	2	1	1	2	2	–	3	1	–	2	–
CO-3	2	2	3	3	3	2	2	2	2	1	3	2	2	3	2	2
CO-4	2	2	–	3	3	–	–	–	2	2	–	3	3	–	2	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Description:

Students are expected to choose real-world contemporary problem and apply the engineering principles learned, to solve the problem through building prototypes or simulations or writing codes or establishing processes/synthesis/correlations etc. The department constituted panel will decide the suitability and worthiness of the project

The B.Tech. Project work will be evaluated for 100 marks, with the following weightages:

Component	Weightage
Periodic evaluation by Guide	40 marks
Mid-term review	20 marks
End Semester viva-voce examination	40 marks
Total	100 marks

Evaluation Criteria:

The student will be evaluated by the panel based on the below criteria. Weightage for each criterion will be determined by the panel and will be informed to the students.

Criteria	Description	Weightages
I	Selection of Topic	10 Marks
II	Literature Survey	15 Marks
III	Objectives and Solution Methodology	15 Marks
IV	Performance of the Task and clarity on the work	30 Marks
V	Report Preparation	15 Marks
VI	Presentation and Response to questions	15 Marks

Evaluation Criteria-CO Mapping

CO / Criteria	CO1	CO2	CO3	CO4
I	X			
II		X		
III		X		

IV			X	
V				X
VI				X

Refer to B.Tech. – Regulations for any further information regarding Mid-term review, End Sem evaluation, Template for report preparation and plagiarism.

Professional Electives

CS1321 **3-0-0 (3)**

Advanced Algorithms

Pre-Requisites: **None**

Course Outcomes:

CO-1	Analyze time and space complexities of algorithms using asymptotic analysis.
CO-2	Analyze amortized time complexity.
CO-3	Classify problems into different complexity classes.
CO-4	Analyze approximation algorithms and determine approximation factor.
CO-5	Design FPT algorithms for some intractable problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	1	-	-	-	1	-	-	-	-	1	3	1	1	2
CO-2	2	2	1	-	-	-	-	-	-	-	-	-	3	1	1	2
CO-3	2	2	2	1	1	-	1	-	-	-	-	-	3	1	2	2
CO-4	3	2	2	2	-	-	1	-	-	-	-	-	2	-	1	-
CO-5	3	2	2	2	-	-	-	-	-	-	-	-	2	-	1	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Algorithm design techniques: Overview, Tree related dynamic programming, Backtracking algorithm for finding a maximum independent set, Enumerating independent Sets.

Amortized Analysis: Three approaches to amortized analysis, Disjoint-sets, Fibonacci heap.

NP-completeness: Polynomial-time solvability, Polynomial-time verification, NP-completeness and reducibility, Example NP-complete Problems – Set cover, Graph Coloring, Domination.

Approximation Algorithms: One way of coping with NP-hardness, The set cover problem, The traveling-sales person problem, The bin packing problem.

Fixed-Parameter Algorithms: Another way of coping with NP-hardness, Bounded search tree, Kernelization, The vertex cover problem.

Randomized Algorithms: Las Vegas and Monte Carlo algorithms, Karger's min cut algorithm and improvements to it by Karger and Stein, Randomized selection.

Learning Resources:

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "*Introduction to Algorithms*", Third Edition, PHI, 2009.
2. Vijay V. Vajirani, "*Approximation Algorithms*", Springer, 2001.

Reference Books:

1. Rodney G. Downey and M. R. Fellows, "*Parameterized Complexity*", Springer, 2012.
2. Rajeev Motwani and Prabhakar Raghavan, "*Randomized Algorithms*", Cambridge University Press, 1995.

CS1323 **3-0-0 (3)**

Advanced Computer Architecture

Pre-Requisites: **CS1208, CS1305**

Course Outcomes:

CO-1	Measure the performance of a computer system.
CO-2	Apply instruction scheduling techniques to improve the performance of a code segment.
CO-3	Understand the working of shared and distributed memory systems and cache coherence protocols.
CO-4	Evaluate the I/O system performance.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	1	2	1	1	-	1	–	1	1	1	-	2	1	2	-
CO-2	2	2	1	1	1	-	-	–	–	1	1	-	2	1	2	-
CO-3	2	2	2	2	1	1	1	–	–	1	1	1	2	1	3	-
CO-4	2	2	2	2	1	1	1	–	–	1	1	1	2	1	3	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Fundamentals of Computer Design: Trends in technology, power, integrated circuits, cost. Dependability, measuring the performance. Pipelining basic concepts, pipeline hazards and other issues, Memory hierarchy, Cache performance, virtual memory, memory protection

Instruction level parallelism: Basic compiler techniques, reducing branch costs, overcoming data hazards, multiple issue and static scheduling, dynamic scheduling, speculation, Limitations of instruction level parallelism, Multithreading and thread level parallelism, VLIW and Vector processors.

Multiprocessors and Thread-level parallelism: Symmetric shared memory architectures, performance of Symmetric shared memory multiprocessor, Distributed shared memory and directory based coherence, Synchronization, Memory consistency

Advanced memory hierarchy design: optimizations of cache performance, memory technology optimizations, protection, virtual memory and virtual machines, case studies Storage Systems: Advanced topics in disk storage, real faults and failures, I/O performance, Reliability measures, design and evaluation of I/O system

Learning Resources:

Text Books:

1. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, 4th Edition, Elsevier, 2007.
2. Kai Hwang, “Advanced Computer Architecture”, Second Edition, Tata McGraw Hill, 2008.

CS1325 **3-0-0 (3)**

Artificial Intelligence

Pre-Requisites: **None**

Course Outcomes:

CO-1	Learn various agent models.
CO-2	Learn and implement various searching techniques on live examples.
CO-3	Create and design knowledge-based agents.
CO-4	Create and design agents to work in uncertain environments.
CO-5	Perform statistical learning on real world problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	1	3	3	2	–	–	–	–	–	–	–	3	3	3	1
CO-2	3	3	3	3	2	1	–	–	2	–	–	3	3	3	3	3
CO-3	3	3	3	3	3	–	–	–	–	–	–	3	3	3	3	3
CO-4	3	3	3	3	3	–	–	–	2	–	–	3	3	3	3	3
CO-5	3	3	3	3	3	–	–	–	–	–	–	3	3	3	3	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction – Agents and Objects – Evaluation of Agents – Agent Design Philosophies - Multi- agent System – Mobile Agents – Agent Communication – Knowledge query and Manipulation Language – Case Study. What is AI? , The Foundations of Artificial Intelligence.

Intelligent Agents – Agents and Environments, Good Behavior: The Concept of Rationality, The Nature of Environments, The Structure of Agents.

Solving Problems By Search – Problem-Solving Agents, Formulating problems, Searching for Solutions, Uninformed Search Strategies, Breadth-first search, Depth-first search, Searching with Partial Information, Informed (Heuristic) Search Strategies, Greedy best-first search, A* Search: Minimizing the total estimated solution cost, Heuristic Functions, Local Search Algorithms and Optimization Problems, Online Search Agents and Unknown Environments.

Adversarial Search – Games, The minimax algorithm, Optimal decisions in multiplayer games, Alpha-Beta Pruning, Evaluation functions, Cutting off search, Games with an Element of Chance.

Logical Agents – Knowledge-Based agents, The Wumpus World, Logic, Propositional Logic: A Very Simple Logic, Reasoning Patterns in Propositional Logic, Resolution, Forward and Backward chaining.

First Order Logic – Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic.

Inference In First Order Logic – Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

Uncertainty – Acting under Uncertainty, Basic Probability Notation, The Axioms of Probability, Inference Using Full Joint Distributions, Independence, Bayes' Rule and its Use.

Probabilistic Reasoning – Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distribution, Exact Inference in Bayesian Networks, Approximate Inference in Bayesian Networks.

Statistical Learning Methods – Statistical Learning, Learning with Complete Data, Learning with Hidden Variables: EM Algorithm.

Learning Resources:

Text Books:

1. Stuart Russell, Peter Norvig, Artificial Intelligence -A Modern Approach, Pearson, 2020, 4th Edition.
2. Nils J Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann Publications, 2000.

Reference Books:

1. Kevin Knight, Elaine Rich and Shivshankar B. Nair, Artificial Intelligence, McGraw Hill, 2017, 3rd Edition.

CS1327 **3-0-0 (3)**

Computational Number Theory

Pre-Requisites: **None**

Course Outcomes:

CO-1	Analyze large integer computations in Z_n.
CO-2	Analyze primality testing and integer factorization algorithms.
CO-3	Develop algorithms for computations in groups, rings and fields.
CO-4	Develop algorithms for computations in polynomial rings.
CO-5	Develop algorithms for computations in finite fields.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	2	2	1	1	-	-	-	-	1	-	-	-	1	-	-
CO-2	2	3	2	2	3	-	-	1	-	1	-	1	-	-	2	-
CO-3	2	1	3	2	2	1	1	-	2	-	1	1	-	1	-	1
CO-4	2	2	2	1	2	1	1	-	1	1	-	-	2	-	-	1
CO-5	2	3	3	3	3	1	-	1	1	-	1	1	1	-	-	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Basic properties of the integers - Divisibility and primality, Ideals and greatest common divisors, unique factorization, Congruences - Basic properties, Solving linear congruences, Residue classes, Euler's phi function, Fermat's little theorem, Arithmetic functions and Mobius inversion.

Computing with large integers - Asymptotic notation, Machine models and complexity theory, Basic integer arithmetic, Computing in Zn, Faster integer arithmetic; Euclid's algorithm - basic Euclidean algorithm, extended Euclidean algorithm, computing modular inverses and Chinese remaindering, Speeding up algorithms via modular computation, Rational reconstruction and applications.

The distribution of primes - Chebyshev's theorem on the density of primes, Bertrand's postulate, Mertens' theorem, sieve of Eratosthenes, prime number theorem and beyond.

Probabilistic algorithms – definitions, Approximation of functions, generating a random number from a given interval, generating a random prime, generating a random non-increasing sequence, generating a random factored number, RSA cryptosystem.

Algebraic Structures - Subgroups, Cosets and quotient groups, Group homomorphisms and isomorphisms, Cyclic groups, structure of finite abelian groups, Rings - Definitions, basic properties, and examples, Polynomial rings, Ideals and quotient rings, Ring homomorphisms and isomorphisms; Modules and vector spaces - Submodules and quotient modules, Module homomorphisms and isomorphisms, Linear independence and bases, Vector spaces and dimension; Matrices - linear maps, inverse of a matrix, Gaussian elimination, Applications of Gaussian elimination; Algebras - The field of fractions of an integral domain, Unique factorization of polynomials, Polynomial congruences, Polynomial quotient algebras, General properties of extension fields, Formal power series and Laurent series, Unique factorization domains.

Primality testing - Trial division, structure of Z_n, The Miller–Rabin test, Generating random primes using the Miller–Rabin test, Perfect power testing and prime power factoring, Factoring and computing Euler's phi function; Deterministic primality testing - The algorithm and its analysis.

Finding generators and discrete logarithms - Finding a generator for Z_p, Computing discrete logarithms Z_p, The Diffie–Hellman key establishment protocol

Quadratic residues and quadratic reciprocity - Quadratic residues, Legendre symbol, Jacobi symbol; Computational problems related to quadratic residues - Computing the Jacobi symbol, Testing

quadratic residuosity, Computing modular square roots, The quadratic residuosity assumption. **Subexponential-time discrete logarithms and factoring** - Smooth numbers, algorithm for discrete logarithms, algorithm for factoring integers, Practical improvements,

Polynomial arithmetic and applications - Basic arithmetic, Computing minimal polynomials, Euclid's algorithm, Computing modular inverses and Chinese remaindering, Rational function reconstruction and applications, Faster polynomial arithmetic; Linearly generated sequences and applications - Basic definitions and properties, Computing minimal polynomials, Solving sparse linear systems, The algebra of linear transformations,

Finite fields - The existence of finite fields, The subfield structure and uniqueness of finite fields, Conjugates, norms and traces; Algorithms for finite fields - Testing and constructing irreducible polynomials, Computing minimal polynomials in F[X]/(f), Factoring polynomials: the Cantor–Zassenhaus algorithm, Factoring polynomials: Berlekamp's algorithm, Deterministic factorization algorithms, Faster square-free decomposition.

Learning Resources:

Text Books:

1. Victor Shoup, "A Computational Introduction to Number Theory and Algebra", Cambridge University Press, 2008.
2. Henri Cohen, "A Course in Computational Algebraic Number Theory", Springer-Verlag, 2000.

Reference Books:

1. Abhijit Das, "Computational Number Theory", Cambridge University Press, 2013.
2. Eric Bach and Jeffrey Shallit, "Algorithmic Number Theory, Volume 1: Efficient Algorithms", MIT Press, 1996.
3. J. P. Buhler, P. Stevenhagen, "Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography", Cambridge University Press, 2008.

CS1329 **3-0-0 (3)**

Programming Language Concepts

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understanding the concepts of evolution of programming languages.
CO-2	Understanding the concepts of object oriented languages, functional and logical programming languages.
CO-3	Analyzing the methods and tools to define syntax and semantics of a languages.
CO-4	Analyzing the design issues involved in various constructs of programming languages.
CO-5	Apply the concepts and identify the issues involved in other advanced features of programming languages.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	1	2	1	1	1	1	1	1	1	1	2	1	3	3
CO-2	1	2	2	3	2	1	1	1	1	1	1	1	2	1	3	3
CO-3	3	3	2	3	1	1	1	1	2	1	1	1	2	1	3	3
CO-4	2	3	1	3	3	2	1	1	1	1	1	1	2	2	2	3
CO-5	2	2	1	2	2	2	1	1	1	1	1	1	2	1	3	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction- The Origins of Programming Languages- Abstractions in Programming Languages - Computational Paradigms -Language Definition - Language Translation.

Language Design Criteria – Efficiency, regularity, security and extensibility - C++: An Object-Oriented Extension of C-Python: A General-Purpose Scripting Language.

Functional Programming - Programs as Functions - Scheme: A Dialect of Lisp - ML: Functional Programming with static typing -Delayed Evaluation- Haskell- Overloading.

Logic Programming-Logic and Logic Programs - Horn Clauses -Resolution and Unification-The Language Prolog - Problems with Logic Programming.

Object-Oriented Programming- Software Reuse and Independence Smalltalk Java C++ - Design Issues in Object-Oriented Languages - Implementation Issues in Object-Oriented Languages.

Syntax-Lexical Structure of Programming Languages -Context-Free Grammars and BNFs -Parse Trees and Abstract Syntax Trees - EBNFs and Syntax Diagrams - Parsing Techniques and Tools-Lexics vs. Syntax vs. Semantics.

Basic Semantics -Attributes, Binding, and Semantic Functions - Declarations, Blocks, and Scope - The Symbol Table - Name Resolution and Overloading - Allocation, Lifetimes, and the Environment Variables and Constants Aliases, Dangling References, and Garbage.

Data Types-Data Types and Type Information - Simple Types - Type Constructors - Type Nomenclature in Sample Languages -Type Equivalence- Type Checking -Type Conversion-Polymorphic Type Checking- Explicit Polymorphism.

Control Expressions and Statements –Expressions - Conditional Statements and Guards-Exception Handling- Procedure Definition and Activation-Procedure Semantics- Parameter-Passing Mechanisms- Procedure Environments, Activations, and Allocation-Dynamic Memory Management-Exception Handling and Environments.

Abstract Data Types and Modules - The Algebraic Specification of Abstract Data Types- Abstract

Data Type Mechanisms and Modules -Separate Compilation in C, C++ Namespaces, and Java Packages- Ada Packages -Modules in ML- Problems with Abstract Data Type Mechanisms.

Formal Semantics- A Sample Small Language- Operational Semantics -Denotational Semantics- Axiomatic Semantics- Proofs of Program Correctness.

Parallel Programming- Introduction to Parallel Processing- Parallel Processing and Programming Languages- Threads – Semaphores- Monitors –Message Passing.

Learning Resources:

Text Books:

1. Kenneth C. Louden, "Programming Language Principles and Practices", 2nd Edition, Thomson 2003.
2. Carlo Ghezzi, Mehdi Jazayeri, "Programming Language Concepts", 3rd Edition, John Wiley & Sons, 1997.

Reference Books:

1. Robert W. Sebesta, Concepts of Programming Languages 12th Edition, Pearson, 2019.
2. David A. Watt, Programming Language Concepts and Paradigms, Prentice-Hall, 1990.

CS1331 **3-0-0 (3)**

Software Testing

Pre-Requisites: **None**

Course Outcomes:

CO-1	Analyze Various test processes and continuous quality improvement.
CO-2	Analyze Types of errors and fault models.
CO-3	Modeling the behaviour using FSM.
CO-4	Application of software testing techniques in commercial environments.
CO-5	Analyze various test tools.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	1	2	1	1	1	2	2	2	2	1	2	1	3	3
CO-2	2	2	2	3	2	1	1	1	1	1	2	1	2	1	3	3
CO-3	3	1	2	3	2	1	1	1	2	1	2	1	3	3	3	3
CO-4	2	2	1	3	3	2	2	2	2	1	2	1	3	2	2	3
CO-5	2	2	1	2	1	1	1	1	1	2	1	1	2	1	3	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Software Testing: Importance of Testing, Testing Process, Terminologies in Testing, Software Development Life Cycle (SDLC) vs. Software Testing Life Cycle (STLC); Principles of Software Testing, Fundamentals of Testing, Psychology of Testing, General Principles of Testing, Levels of Testing; Types of Testing, Manual Testing, Automation Testing, Black Box Testing, White Box Testing, Grey Box Testing; Static Testing, Reviews, Walkthroughs, Inspections; Test Design Techniques, Black Box Test Design Techniques, Equivalence Partitioning, Boundary Value Analysis, Decision Table Testing, State Transition Testing, White Box Test Design Techniques, Statement Coverage, Decision Coverage, Condition Coverage, Path Coverage; Dynamic Testing, Unit Testing, Integration Testing, System Testing, Acceptance Testing; Levels of Testing, Component Testing, Integration Testing, System Testing, Acceptance Testing; Test Management, Test Planning, Test Estimation, Test Monitoring and Control, Configuration Management, Risk and Testing, Incident Management; Test Automation, Introduction to Test Automation, Tools for Test Automation, Scripting Languages for Test Automation, Benefits and Risks of Test Automation; Specialized Testing, Performance Testing, Load Testing, Stress Testing, Usability Testing, Security Testing, Compatibility Testing, Regression Testing; Testing Tools, Test Management Tools, Requirement Management Tools, Test Execution Tools, Performance Testing Tools; Quality Metrics and Measurements, Software Quality, Quality Attributes, Metrics for Quality, Process Metrics, Product Metrics; Case Studies and Practical Examples, Real-World Scenarios, Practical Examples, Case Studies in Testing; Trends in Software Testing, Current Trends in Software Testing, Future of Software Testing, Innovations in Testing Techniques.

Learning Resources:

Text Books:

1. Naresh Chauhan, “Software Testing- Principles and Practices”, Oxford University Press, Second Edition, 2016.
2. Baris Beizer, "Software Testing techniques", Dreamtech, Second Edition, 1990.

Reference Books:

1. Paul Ammann George and Jeff Offutt , “*Introduction to Software Testing*”, Cambridge University Press, Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo, Second Edition, 2008.

CS1322 **3-0-0 (3)**

Advanced Data Mining

Pre-Requisites: **None**

Course Outcomes:

CO-1	Analyze Algorithms for sequential patterns.
CO-2	Determine patterns from time series data.
CO-3	Develop algorithms for Temporal Patterns.
CO-4	Apply Graph mining algorithms to Web Mining.
CO-5	Compare and contrast distributed algorithms for data mining tasks.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	-	-	3	3	1	1	–	–	–	–	–	1	–	–	3	1
CO-2	2	2	2	3	1	1	–	-	–	–	–	1	2	–	2	-
CO-3	1	1	3	3	-	1	1	1	–	–	1	1	3	–	2	1
CO-4	1	1	3	3	1	1	1	1	–	–	1	1	3	–	3	1
CO-5	1	1	3	3	1	1	1	1	–	–	1	1	3	–	3	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Sequential Pattern Mining concepts, primitives, scalable methods; Transactional Patterns and other temporal based frequent patterns, Mining Time series Data, Periodicity Analysis for time related sequence data, Trend analysis, Similarity search in Time-series analysis; Graph Mining, Mining frequent subgraphs, finding clusters in large graphs; Web Mining, Mining the web page layout structure, mining web link structure, Automatic classification of web documents and web usage mining; Distributed Data Mining, Distribute data mining framework, Distributed data source, Distributed data mining techniques, Distributed classifier learning, distributed clustering, distributed association rule mining.

Learning Resources:

Text Books:

1. Jiawei Han and M Kamber, Data Mining Concepts and techniques, Morgan Kaufmann Publishers In, 2022; Fourth Edition.
2. Chris Chatfield, The Analysis of Time Series: An Introduction, Chapman & Hall/CRC, 2003, Sixth Edition.

Reference Books:

1. Bing Liu, Web Data Mining, Springer, 2011, Second Edition.
2. Pang-Ning Tan, Michael Steinbach, Anuj Karpatne and Vipin Kumar, Introduction to Data Mining, Pearson, 2018, Second Edition.

CS1324 **3-0-0 (3)**

Advanced Databases

Pre-Requisites: **None**

Course Outcomes:

CO-1	Design distributed database for application development.
CO-2	Apply query optimization principles for optimizing query performance in centralized and distributed database systems.
CO-3	Design distributed database schema using principles of fragmentation and allocation.
CO-4	Apply distributed transaction principles for handling transactions in distributed database applications.
CO-5	Apply distributed database administration principles for managing distributed database.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	3	1	1	-	-	1	1	1	1	1	3	2	-	1
CO-2	2	2	2	1	1	-	-	1	-	-	1	1	1	1	-	2
CO-3	1	2	3	1	1	-	-	1	1	1	1	1	3	2	-	1
CO-4	1	1	1	1	2	-	-	1	1	1	1	1	1	2	1	1
CO-5	-	1	1	1	2	-	-	1	1	1	1	1	1	2	1	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Database-System Architectures: Centralized Database Systems, Server System Architectures, Parallel Systems, Distributed Systems, Transaction Processing in Parallel and Distributed Systems, Cloud-Based Services Parallel and Distributed Storage: Data Partitioning, Dealing with Skew in Partitioning, Replication, Parallel Indexing, Distributed File Systems, Parallel Key-Value Stores

Parallel and Distributed Query Processing: Parallel Sort, Parallel Join, Other Operations, Parallel Evaluation of Query Plans, Query Processing on Shared-Memory Architectures, Query Optimization for Parallel Execution, Parallel Processing of Streaming Data, Distributed Query Processing

Parallel and Distributed Transaction Processing: Distributed Transactions, Commit Protocols, Concurrency Control in Distributed Databases, Replication, Extended Concurrency Control Protocols, Replication with Weak Degrees of Consistency, Coordinator Selection, Consensus in Distributed Systems

Advanced Indexing Techniques: Bloom Filter, Log-Structured Merge Tree and Variants, Bitmap Indices, Indexing of Spatial Data, Hash Indices.

Block chain Databases: Block chain Properties, Achieving Block chain Properties via Cryptographic Hash Functions, Consensus, Data Management in a Block chain, Smart Contracts, Performance Enhancement, Emerging Applications

No SQL databases: Types, CAP theorem, Key-values storage, Column value storage, Scalability and Performance, Use cases, AI and Machine Learning Integrations

Learning Resources:

Text Books:

1. M T Ozsu, Patrick Valduriez, "Principles of Distributed Database Systems", Prentice Hall, 1999.
2. S. Ceri and G. Pelaggati, "Distributed Database System Principles and Systems", MGH, 1985.

CS1326 **3-0-0 (3)**

Bio-Informatics

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand various biometrics.
CO-2	Derive features from different biometric traits.
CO-3	Identify the acceptance issues associated with the design and implementation of biometric systems.
CO-4	Identify various Biometric security issues.
CO-5	Design biometric systems in various applications.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	1	2	1	-	-	-	-	-	-	-	1	1	1	1
CO-2	3	3	1	2	1	-	-	-	-	-	-	-	-	-	1	-
CO-3	3	3	1	2	1	-	-	-	-	-	-	-	1	1	-	-
CO-4	3	2	1	2	1	-	-	-	-	-	-	-	1	-	1	1
CO-5	2	3	1	3	1	-	-	-	-	-	-	-	-	-	-	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Person Recognition, Biometric Systems, Biometric Functionalities, Biometric System Errors, Design cycle of Biometric Systems, Applications of Biometric Systems, Security and Privacy issues. **Fingerprint recognition**, friction and ridge pattern, fingerprint acquisition, sensing techniques, feature extraction, ridge orientation and frequency estimation, singularity extraction, minutiae extraction, matching, fingerprint indexing, synthesis, palmprint recognition.
Face recognition sensors, cameras, Video sequences, Viola Jones face detection, appearance based, model based, texture-based face recognition, Principal Component Analysis, Linear Discriminant Analysis, Independent Component Analysis, Heterogeneous face recognition.

Iris Recognition: Design, segmentation methods, normalization methods, encoding and matching.

Additional biometric traits: Ear, Gait, Hand geometry, Ocular biometrics, Online and offline Signature authentication, soft biometrics.

Multibiometric: multi-sensor, multi-algorithm, multi-instance, multi-sample, multi-modal systems, processing architectures, processing sequence, sensor level, feature level, score level, rank level, and decision level fusion, Attacks on biometric systems, adversary attacks, user interface, processing, template databases.

Learning Resources:

Text Books:

1. Anil K. Jain, Arun A. Ross, Karthik Nandakumar, "Introduction to Biometrics", Springer, 2011.
2. Arun A.Ross, Karthik Nandakumar, Anil K.Jain, "Handbook of Multibiometrics", Springer, 2006.

Reference Books:

1. John Chirillo, Scott Blaul, "Implementing Biometric Security", 1st Edition, Wiley Eastern Publication, 2005.
2. Anil K Jain, Patrick Flynn and Arun A Ross, "Handbook of Biometrics", Springer, USA, 2010.
3. Samir Nanavati, Michael Thieme and Raj Nanavati, "Biometrics – Identity Verification in a Networked World", John wiley & Sons, New Delhi, 2003.
4. Paul Reid, "Biometrics for Network Security", Pearson Education, New Delhi, 2004.
5. David D Zhang, "Automated Biometrics: Technologies and Systems", Kluwer Academic Publishers, New Delhi, 2000.

CS1328 **3-0-0 (3)**

Computer Vision and Image Processing

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand Image representation and modeling.
CO-2	Apply Image transformation methods.
CO-3	Implement image processing algorithms.
CO-4	Design of face detection and recognition algorithms.
CO-5	Analyze the features and propose new features of images.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	2	2	2	2	2	1	2	2	2	1	3	2	3	2
CO-2	3	2	3	2	2	2	3	1	3	3	3	1	3	3	2	2
CO-3	3	2	3	2	3	2	3	1	3	3	3	1	2	3	2	2
CO-4	2	2	3	2	2	2	3	1	3	3	3	1	3	3	2	2
CO-5	2	3	2	3	1	2	2	1	2	2	2	1	3	2	3	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

The image model and acquisition, image shape, sampling, intensity images, color images, range images, image capture, scanners. Statistical and spatial operations, Gray level transformations, histogram equalization, multi image operations. Spatially dependent transformations, templates and convolution, window operations, directional smoothing, other smoothing techniques. Segmentation and Edge detection, region operations, Basic edge detection, second order detection, crack edge detection, edge following, gradient operators, compass & Laplace operators. Morphological and other area operations, basic morphological operations, opening and closing operations, area operations, morphological transformations. Image compression: Types and requirements, statistical compression, spatial compression, contour coding, quantizing compression. Representation and Description, Object Recognition, 3-D vision and Geometry, Digital Watermarking. Texture Analysis.

Learning Resources:

Text Books:

1. D. A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, PHI Learning, 2009.
2. Milan Soanka, Vaclav Hlavac and Roger Boyle, Digital Image Processing and Computer Vision, Cengage Learning, 2014

Reference Books:

1. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Pearson Education, 2007.

CS1330 **3-0-0 (3)**

Data Networks

Pre-Requisites: **None**

Course Outcomes:

CO-1	Analyze layered network architecture and passage of data over communication links.
CO-2	Analyze delay models in Data Networks using Queueing Systems for messaging and delay sensitive applications.
CO-3	Design and analyze routing algorithms for Internet and multi-hop autonomous networks.
CO-4	Analyze flow and rate control algorithms between a sender and receiver in wide area networks.
CO-5	Design and analyze software defined networking algorithms for data forwarding through Internet.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	3	3	2	-	-	-	2	2	2	3	2	2	-	2
CO-2	3	3	3	3	2	-	-	-	2	2	2	3	2	2	-	3
CO-3	3	3	3	3	2	-	-	-	2	2	2	3	3	3	-	3
CO-4	3	3	3	3`	2	-	-	-	2	2	2	3	3	3	-	3
CO-5	3	3	3	3	3	-	-	-	2	2	2	3	3	3	-	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Physical Layer Channels: Frequency- and Time- Division Multiplexing; Error Detection; ARQ; Framing.

Point-to-Point Protocols at Network Layer: Error Recovery, The X.25 Network Layer Standard, The Internet Protocol.

Transport Layer: Transport Layer Standards, Addressing and Multiplexing TCP, Error Recovery in TCP; Flow Control in TCP/IP, Asynchronous Transfer Mode (ATM).

Delay Models in Data Networks: The M/M/1 Queueing System, M/M/m, M/M/∞, M/M/m/m and other Markov Systems, Networks of Transmission Lines, Networks of Queues – Jackson's Theorem.

Multi-Access Communication: Packet Radio Networks, Splitting Algorithms, Carrier Sensing, Multi-access Reservations.

Routing in Data Networks: Wide-Area Network Routing, Interconnected Network Routing, Network Algorithms and Shortest Path Routing, Broadcasting Routing Information, Flow Models, Optimal Routing and Topological Design; Characterization of Optimal Routing.

Flow Control: Main Objectives of Flow Control, Window Flow Control, Rate Control Schemes, Rate Adjustment Algorithms, Classification of TCPs.

Software Defined Networks: Fundamental Characteristics of SDN, the OpenFlow Specification, SDN via Hypervisor-Based Overlays, SDN in the Data Center, SDN in Wide Area Networks, SDN in Mobile Networks, SDN Ecosystem and Network Virtualization.

Learning Resources:

Text Books:

1. Bertsekas, Dimitri, and Robert Gallager, "Data Networks" (2nd Edition). Upper Saddle River, NJ, USA, Second Edition, Pearson Education/Prentice Hall, 1992.
2. Walrand and Varaiya, "High Performance Communication Networks", San Francisco, CA:

Morgan Kaufmann Publishers, 1996.

Reference Books:

1. Stevens, "TCP/IP Illustrated. Reading", MA: Addison-Wesley Pub. Co., c1994-c1996.
2. Paul Goransson, Chuck Black, "Software Defined Networking: a comprehensive approach", Morgan Kaufmann (Elsevier), 2014.

CS1332 **3-0-0 (3)**

Design Patterns

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand common design patterns in the context of incremental/iterative development.
CO-2	Evaluate and retractor software source code using patterns.
CO-3	Analyze and combine design patterns to work together in software design.
CO-4	Implement the design patterns in an object oriented language.
CO-5	Understand the benefits of a pattern approach over program in a software application.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	3	2	3	2	2	2	1	2	3	2	1	2	2	2	1
CO-2	2	2	3	2	2	2	2	1	3	3	3	1	2	2	2	2
CO-3	2	3	3	3	2	2	3	1	2	3	2	1	1	2	3	2
CO-4	2	2	3	2	3	3	2	1	3	3	3	1	3	3	2	1
CO-5	2	2	2	2	1	2	2	1	2	2	2	1	2	2	2	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction: What is a Design Pattern, Design Patterns in Smalltalk MVC, Describing Design Patterns, the Catalogue of Design Patterns, Organizing the catalogue, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.

A Case Study: Designing a Document Editor: Design Problems, Document Structure, Formatting, Embellishing the User Interface, and Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations, Spelling Checking and Hyphenation, Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton.

Structural Pattern: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy.

Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor, a Brief History, and the Pattern Community

Learning Resources:

Text Books:

1. Erich Gamma, "Design Patterns", Addison-Wesley, 1994.
2. Frank Buschmann, RegineMeunier, Hans Rohnert, Peter Sommerlad, Michael Stal, "Pattern- Oriented Software Architecture: A System of Pattern", John Wiley & Sons; 1996.

CS1334 **3-0-0 (3)**

Foundations of Blockchain Technology

Pre-Requisites: **CS1106, CS1205, CS1305, CS1302**

Course Outcomes:

CO-1	To understand the essential concepts and structural components of blockchain and the rationale behind its implementation.
CO-2	To understand the core technology, many types of blockchains and protocols that operate the blockchain.
CO-3	To articulate and develop and test blockchain-compatible diverse applications with smart contracts.
CO-4	To understand and analyze the advantages and disadvantages of employing blockchain technology in various industries and technologies.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	2	2	-	-	-	-	-	-	2	2	2	2	2	2
CO-2	2	2	2	2	2	2	2	-	2	2	2	2	2	2	3	3
CO-3	3	3	3	3	3	3	3	-	3	3	3	3	3	3	3	3
CO-4	3	3	2	2	2	-	-	-	3	3	3	3	3	3	3	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Blockchain- Key Concepts of Blockchain, Features of Blockchain, Importance of Blockchain, Blockchain 1.0, 2.0, and 3.0, Issues to Centralized System, Centralized to Decentralized and Distributed System, Building Blocks of Blockchain- Distributed Ledgers & P2P Networks, Block Header, Transaction Organization.

Cryptographic Primitives, Basic Crypto Primitives- Hash Functions- Properties of Hash Function, Nonce, Merkle Trees, Hash Pointers, Public Key Cryptography- Public/private keys, Signature schemes, Signature correctness,

Decentralization- Distributed shared ledger, Distributed Consensus-Distributed Consensus Protocol. Classical theory of consensus, Byzantine Generals Problem possibility and impossibility results, Asynchronous consensus and Byzantine Fault Tolerance

Types of Blockchain- Permission-less Blockchain- Bitcoin-Introduction to Bitcoin, Bitcoin Transaction, Bitcoin Protocol, Bitcoin Wallets, Bitcoin Block, Bitcoin Scripts, Bitcoin Network, Bitcoin Mining-Nakamoto Consensus- Proof-of-work, Mining target T, Proof-of-work equation, Mining Algorithm, Mining and reward, Block freshness, Partial and full nodes, Attacks on Bitcoin- Double-spend attacks, Selfish mining, **Ethereum Blockchain,** Introduction to Ethereum, Ethereum Networks, Ethereum Wallets, Ethereum Clients, Ethereum accounts, Transactions and State, Smart contracts, Privacy-preserving smart contracts, Proof-of-stake, Variants of Ethereum blockchain

Permissioned Blockchain- Hyperledger Fabric-, State Machine Replication, Distributed State Machines, MSP, Consensus- Raft Consensus Algorithm, Safety and liveness, Privacy based blockchain, ZCash, Zero-knowledge-proof, R3 Corda, Corda Network

Blockchain Security- Attacks on Blockchain and their Countermeasures, Application & Use cases of Blockchain

Learning Resources:

Text Books:

1. Bitcoin and Cryptocurrency Technologies, A. Narayanan, J. Bonneau, E. Felten, A. Miller and S. Goldfeder, Princeton University Press. Henceforth termed as PUP (Princeton university press).
2. Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, 3rd Edition, Imran Bashir, Packt Publishing, 2020, ISBN: 9781839213199.

Reference Books:

1. William Magnuson, “Blockchain Democracy- Technology, Law and the Rule of the Crowd”, Cambridge University Press, 2020.
2. Pethuru Raj, Kavita Saini, Chellammal Surianarayanan, “Blockchain Technology and Applications”, CRC Press, 2021.

Other Suggested Readings:

1. Introduction to Cryptocurrencies, a basic online course by Haseeb Qureshi.

CS1336 **3-0-0 (3)**

Information Retrieval
Mobile Computing

Pre-Requisites: **None**

Course Outcomes:

CO-1	Identify mobile computing societal applications and communication constraints in wireless environment.
CO-2	Analyze mobile IPv4 and IPv6 architectures with agents and proxies.
CO-3	Design MAC protocols for wireless networks.
CO-4	Evaluate the performance of TCP protocols in Wireless Networks with mobile nodes.
CO-5	Design and analyze the existing routing protocols for multi-hop wireless networks.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	1	3	2	2	2	-	-	-	1	1	2	2	1	1	-	2
CO-2	1	2	2	2	2	-	-	-	1	1	2	2	1	2	-	2
CO-3	2	2	3	2	3	-	2	-	1	1	2	2	2	3	-	3
CO-4	2	2	3	2	3	-	2	-	1	1	2	2	1	3	-	3
CO-5	2	2	3	2	3	-	2	-	1	1	2	2	2	3	-	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Basic communication Technologies, Introduction to Mobile Networks, Introduction to different categories of Wireless networks (MANET: Mobile ad-hoc networks- Communication Architectures of a typical MANET, Applications of MANET, WSN: Wireless Sensor Networks- topologies in WSN- Linear, Grid and Cluster based topologies, communication architectures in a WSN, applications of WSNs, VANET: Vehicular Ad-hoc Networks- communication architectures in VANET, Applications of VANET, PAN: Personal Area Networks- the Bluetooth technology, the blue tooth specifications, DTN: Delay Tolerant Network-delay tolerant network architecture, applications of DTN), Wireless Communication Fundamentals, Cellular Wireless Networks.

Medium Access Control Layer- Hidden terminal problem, Exposed terminal problem, Collision avoidance, Congestion Avoidance, Congestion control, Energy Efficiency, MACA and MACAW protocols, Wireless LAN and IEEE 802.11- Network architecture, the physical layer, the MAC layer, security.

Detailed network layer functionalities in multi-hop wireless networks- Mobile Ad-hoc Networks- broadcasting in a MANET, flooding generated broadcast storm problem, rebroadcasting schemes, Issues in providing multicasting in MANET, Multicast routing protocols, Geocasting- Geocast routing protocols. Mobile Network Layer (Mobile IP), DHCP (Dynamic host configuration protocol), Routing in Mobile Ad hoc Networks (MANET)- Topology-based versus position based approaches, Proactive routing protocols, Reactive routing protocols, Hybrid routing protocols, position based routing issues and forwarding strategies, AODV (Ad-hoc On-Demand Distance Vector Routing Protocol)- Analysis of AODV under mobility and Faults in a network, DSR (Dynamic Source Routing)-Analysis of DSR under mobility and Faults in a network, Secure routing protocols in MANET, Wireless Sensor Networks: (Routing protocols, Localization methods, Sensor Deployment Strategies), traffic flow pattern in WSN- one to many, many to one and many to many, Routing protocols for Delay Tolerant Networks, Routing protocols for Vehicular Ad-hoc Networks, Wireless Access Protocol, GPS (Global positioning system) and applications, RFID and its applications.

Learning Resources:

Text Books:

1. Jochen Schiller, "Mobile Communications", Second Edition, Pearson Education, 2003.
2. C D M Cordeiro, D. P. Agarwal, "Adhoc and Sensor Networks: Theory and applications", World Scientific, 2006.

Reference Books:

1. Asoke K Talukder and Roopa R. Yavagal, "Mobile Computing – Technology, Applications and Service Creation"; TMH Pub., New Delhi, 2006.

CS1340 **3-0-0 (2)**

Natural Language Processing

Pre-Requisites: **CS1304**

Course Outcomes:

CO-1	Understand the Text representation and Text pre-processing techniques.
CO-2	Understand language modeling with N-Grams.
CO-3	Apply syntactic parsing to produce parse trees
CO-4	Design NLP Systems for Text Summarization, Classification and Translation.
CO-5	Evaluate the performance NLP System.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	1	-	-	1	-	-	-	-	-	-	-	2	3	2	1
CO-2	3	2	-	-	3	-	-	-	-	-	-	-	3	2	2	1
CO-3	3	2	1	3	1	-	-	-	-	-	-	-	3	3	2	1
CO-4	-	-	3	-	3	-	-	1	-	-	-	-	2	3	2	1
CO-5	-	-	-	-	2	-	-	-	-	-	-	-	2	2	1	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction and Basic Text Processing, Text Representation- one-hot encoding, TF-IDF, Bag-of-Words, Word2Vec, Glove Embedding, Text-Preprocessing- Tokenization, Stemming-Porters Stemming algorithm, Lemmatization, Normalization, Spell Checker- Spelling Correction, Edit Distance, Language Modeling- Probabilistic Language Modelling- N-gram Modelling, Language Models Evaluation, Text Parsing, Part-of-Speech Tagging, POS with Hidden Markov model (HMM), Text Summarization-Abstractive and Extractive Text Summarization. Text Classification- Text Classification Techniques – Topic Modelling, Sentiment Analysis, Named Entity Recognition(NER), Machine Translation, Introduction to Large Language Models (LLMs)

Learning Resources:

Text Books:

1. Daniel Jurafsky and James H. Martin, "Speech and Language Processing", 3rd Edition, 2008.
2. Uday Kamath, John Liu, James Whitaker, “Deep Learning for NLP and Speech Recognition”, Springer, 2020.

Reference Books:

1. Allen, James, "Natural Language Understanding", Second Edition, Benjamin/ Cumming, 1995.
2. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana, “Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems”, O’Reilly, 2020.

CS1342 **3-0-0 (3)**

Program Analysis and Verification

Pre-Requisites: **None**

Course Outcomes:

CO-1	Apply the theory of abstract interpretation.
CO-2	Examine existing techniques.
CO-3	Combine algorithms for program analysis.
CO-4	Experiment with Soot and Java software packages.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	2	2	2	2	1	1	3	2	3	1	3	3	2	2
CO-2	2	2	2	2	3	2	1	1	3	2	3	1	2	2	3	1
CO-3	3	3	3	3	3	2	1	1	3	2	3	1	3	2	2	2
CO-4	2	2	2	2	3	2	1	1	2	2	2	1	2	2	2	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction - Nature of Program Analysis, Data Flow Analysis, Equational Approach, and Constraint Based, Type and Effect Systems, Effect Systems, Algorithms.

Data Flow Analysis – Intraprocedural Analysis, Available Expressions Analysis, Reaching Definitions Analysis, Very Busy Expressions Analysis, Live Variables Analysis, Structural Operational Semantics, Correctness of Live Variables Analysis, Monotone Frameworks, Equation Solving, Interprocedural Analysis, Shape Analysis.

Constraint Based Analysis - Abstract 0-CFA Analysis, Theoretical Properties, Constraint Based 0-CFA Analysis, Adding Context Information.

Abstract Interpretation – Correctness, Approximation of Fixed Points, Galois Connections, Induced Operations.

Type and Effect Systems - Control Flow Analysis, Theoretical Properties, Inference Algorithms, Effects, Behaviours.

Algorithms - Worklist Algorithms, Iterating in Reverse Postorder.

Learning Resources:

Text Books:

1. Flemming Nielson, Hanne R. Nielson and Chris Hankin, "Principles of Program Analysis", Springer, 2005.
2. Edmund M. Clarke, Jr., Orna Grumberg, Daniel Kroening, Doron Peled and Helmut Veith, "*Model Checking*", MIT Press, Second Edition, 2018.

Reference Books:

1. Aaron R. Bradley and Zohar Manna, "The Calculus of Computation", Springer, 2007.
2. Daniel Kroening and Ofer Strichman, "Decision Procedures: An Algorithmic Point of View", Springer, 2008.

CS1344 **3-0-0 (3)**

Software Metrics and Software Project Management

Pre-Requisites: **CS1204**

Course Outcomes:

CO-1	Determine the software measurement attributes and metrics.
CO-2	Plan and evaluate software projects.
CO-3	Analyze factors involved in implementation of software projects.
CO-4	Understand project monitoring and control techniques.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	1	2	1	2	1	1	1	1	1	1	1	1	2	1	3	3
CO-2	1	2	2	3	2	1	1	1	1	1	1	1	2	1	3	3
CO-3	1	1	2	3	1	1	1	1	1	1	1	1	2	1	3	3
CO-4	1	1	1	3	3	1	1	1	2	1	1	1	2	2	2	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to software project management: The characteristics of software projects, Reasons for IT project failure, Objectives of project management, components of Project Management, Stakeholders, Stages of Project, Activities Covered by Software Project Management, Project and Product Life Cycles, project management techniques, role of project manager.

Software Metrics- Monitoring & measurement of SW development, cost, size and time metrics, methods and tools for metrics, issues of metrics in multiple projects.

Project Planning: Planning process, definition, estimation, understanding organizations, stakeholder's management, project phases & project life cycles, objectives of project planning, Project schedule, Iterative steps for planning, Project Management Plan, types of risk, managing risk, risk planning and control.

Project Monitoring & Control: Project Control, effort data, Monitoring and Control, Quantitative techniques, Monitoring Process, Tools and techniques,

Software Quality- Quality in SW development, quality assurance, quality standards and certifications, the process and issues in obtaining certifications, the benefits and implications for the organization and its customers, change management.

Configuration Management: Configuration management process; Software configuration items; Version control; change control; Configuration audit; Status reporting.

Case Studies and Practical Examples: Real-World Scenarios, Case Studies on SPM tools.

Learning Resources:

Text Books

1. Norman E. Fenton, Shari Lawrence Pfleeger, "Software Metrics - A Rigorous and Practical Approach", 2nd Edition, PWS Pub, 1996.
2. Walker Royce, "Software Project Management", Addison Wesley, 1998.

Reference Books

1. Bob Hughes, Mike Cotterell and Rajib Mall, Software Project Management, Third Edition, Tata McGraw-Hill, 2009.
2. Pankaj Jalote, "Software Project Management in Practice", Pearson Education Inc. Delhi, 2002.

CS1421 **3-0-0 (3)**

Cloud Computing

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand cloud computing concepts and history.
CO-2	Mastering principles of parallel and distributed computing.
CO-3	Proficiency in virtualization techniques.
CO-4	Analyzing cloud computing architectures.
CO-5	Familiarity with industry cloud platforms.
CO-6	Exploring advanced cloud computing topics and tools.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	2	–	2	–	2	–	–	–	–	2	2	–	–	2
CO-2	3	3	2	2	2	1	2	–	2	2	–	2	3	3	2	2
CO-3	3	3	2	3	2	2	2	–	2	2	2	2	3	3	2	2
CO-4	3	3	2	2	2	1	2	–	2	2	–	2	3	2	–	3
CO-5	–	–	3	–	2	2	2	–	2	2	2	2	2	–	1	2
CO-6	–	–	3	–	3	2	2	–	–	2	–	2	2	–	2	2

1 – Slightly; **2 – Moderately;** **3 – Substantially**

Syllabus:

Introduction: Cloud computing at a glance; Historical developments; Building cloud computing environments

Principles of Parallel and Distributed Computing: Eras of computing; Parallel vs. distributed computing; Elements of parallel computing; Elements of distributed computing; Technologies for distributed computing

Virtualization: Introduction; Characteristics of virtualized environments; Taxonomy of virtualization techniques; Virtualization and cloud computing; Pros and cons of virtualization; Technology examples

Cloud Computing Architecture: Introduction; The cloud reference model; Types of clouds; Economics of cloud; Open challenges

Cloud Platforms in Industry: Amazon web services; Google AppEngine; Microsoft Azure

Cloud Applications: Scientific applications; business and consumer applications

Advanced Topics in Cloud Computing: Energy efficiency in clouds; Market-based management of clouds; Federated clouds/Intercloud; Third-party cloud services; Resource allocation; Task scheduling; Service management; Data management; Resource management; Security and privacy; Edge computing; Fog computing; Osmotic computing

Toolkits: CloudAnalyst; CloudSim; iFogSim; Haizea – An open source VM-based lease manager

Learning Resources:

Text Books:

1. Rajkumar Buyya, Christian Vecchiola and S. Thamarai Selvi, Mastering Cloud Computing: Foundations and Applications Programming, Morgan Kaufmann, 2013.

Reference Books:

1. Barrie Sosinsky, Cloud Computing Bible, Wiley Publishing, 2011.
2. Tim Mather, Subra Kumaraswamy and Shahed Latif, Cloud Security and Privacy, O'Reilly, 2009.

Other Suggested Readings:

1. NPTEL Course on Cloud Computing by Prof. Soumya Kanti Ghosh, IIT Kharagpur, Prof. Rajiv Mishra, IIT Patna, Prof. Bidisha Chaudhuri, Prof. Amit Prakash, IIIT Bangalore.

CS1423 **3-0-0 (3)**

Data Science

Pre-Requisites: **None**

Course Outcomes:

CO-1	Apply statistical methods to data for inferences.
CO-2	Analyze data using Classification, Graphical and computational methods.
CO-3	Understand Data Wrangling approaches.
CO-4	Perform descriptive analytics and data visualization over massive data.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	1	1	2	2	–	1	–	–	1	–	–	2	1	2	2
CO-2	2	1	2	2	2	–	1	–	–	1	–	–	2	2	3	2
CO-3	2	1	1	2	2	–	1	–	–	1	–	–	2	2	3	3
CO-4	2	2	2	2	2	1	2	–	–	1	1	–	2	2	2	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to data science, data science life cycle, data science process, roles, tools, and technologies, data collection, data wrangling, focusing on techniques for data collection, cleaning, pre-processing, and transformation. Overview of Random variables and probability distributions. Statistical learning: Assessing model accuracy, Bias-Variance Trade-Off, Descriptive Statistics, Dependent and Independent events; Linear Regression: Simple and multiple linear regressions, regularization, Lasso, Ridge, and Elastic-Net Regression. Comparison of Linear regression with K-nearest neighbours. Logistic Regression, LDA, QDA. PCA and SVD. Hypothesis Testing, Student's t-test, paired t and U test, correlation and covariance, tests for association; association rules and correlations; hypothesis testing, correlation and causation, ANOVA, and statistical significance. Exploratory data analysis (EDA), descriptive statistics, data visualization techniques, and identifying patterns and trends, Histograms and frequency polygons, Box-plots, Quartiles, Scatter Plots, Heat Maps. Matrix visualization, Scientific Design Choices in Data Visualization, Higher-dimensional Displays and Special Structures, Visual data mining. Data Wrangling: Data Acquisition, Data Formats, Imputation, split-apply-combine paradigm. Descriptive Analytics: Data Warehousing and OLAP, Data Summarization, Data de- duplication, Data Visualization using CUBEs.

Learning Resources:

Text Books:

1. Gareth James Daniela Witten Trevor Hastie, Robert Tibshirani, An Introduction to Statistical Learning with Applications in R, February 11, 2013, web link: www.statlearning.com (1 to 4 Chapters)
2. Mark Gardener, Beginning R : The Statistical Programming Language, Wiley, 2015.

Reference Books:

1. Han , Kamber, and J Pei, Data Mining Concepts and Techniques, 3rd edition, Morgan Kaufman, 2012. (Chapter 2 and Chapter 4)
2. C Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.

CS1425 **3-0-0 (3)**

Deep Learning

Pre-Requisites: **MA1161, MA1264**

Course Outcomes:

CO-1	Design Multi-Layer neural network to solve Supervised Learning problems.
CO-2	Design Autoencoders to solve Unsupervised Learning problems.
CO-3	Apply Classical Supervised methods CNN'S, FCN, RCNN etc. for Image Denoising, Segmentation and Object detection problems.
CO-4	Use Long Shot Term Memory (LSTM) Networks, GRU for time series analysis classification problems.
CO-5	Apply Generative Adversarial Networks, GAN, VAE to solve Supervised and Unsupervised Learning Problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	3	3	2	3	1	–	–	1	1	1	–	3	2	2	1
CO-2	2	3	3	2	3	1	–	–	1	1	1	–	3	2	3	2
CO-3	3	2	2	3	3	2	–	–	1	1	1	–	3	2	3	2
CO-4	2	3	3	3	3	2	–	–	1	1	1	–	3	2	3	2
CO-5	3	2	2	3	3	2	–	–	1	1	1	–	3	2	3	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

History of Deep Learning, Deep Learning Success Stories, McCulloch Pitts Neuron, Multilayer Perceptrons (MLPs), Representation Power of MLPs, Sigmoid Neurons, Gradient Descent, Feed Forward Neural Networks, Back propagation, Optimizers: Gradient Descent (GD), Momentum Based GD, Nesterov Accelerated GD, Stochastic GD, ADAM; Auto encoders and relation to PCA, Regularization in auto encoders, Denoising auto encoders, Sparse auto encoders; Regularization: Bias Variance Tradeoff, L2 regularization, Early stopping, Dataset augmentation, Dropout, Drop connect; Greedy Layer wise Pretraining, Better activation functions, Better weight initialization methods, Batch Normalization; Learning Vectorial Representations of Words; Convolutional Neural Networks, LeNet, AlexNet, ZFNet, VGGNet, GoogLeNet, ResNet, DenseNet; Recurrent Neural Networks, Back propagation through time (BPTT), Vanishing and Exploding Gradients, Truncated BPTT, GRU, LSTMs; Semantic Segmentation, Instance Segmentation, FCN, Unet ; Object Localization, Region Proposal Networks, RCNN, RFCN, DeYolo; Encoder Decoder Models, Generative Adversarial Networks, GAN, VAE, One Shot Learning, Deep Reinforcement Learning, Attention Mechanism, Attention over images and Transformers.

Learning Resources:

Text Books:

1. Deep Learning- Ian Goodfelllow, Yoshua Benjio, Aaron Courville, The MIT Press.
2. C Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.

Reference Books:

1. Simon Hayking, "Neural Networks and Learning Machines", Pearson, 1999
2. Charu C. Aggarwal. Neural Networks and Deep Learning: A Textbook. Springer. 2019.

CS1427 **3-0-0 (3)**

Design of Secure Protocols

Pre-Requisites: **None**

Course Outcomes:

CO-1	Identify security goals and risks.
CO-2	Analysis of risks and threat modeling.
CO-3	Integrate different technologies to achieve security goals.
CO-4	Develop security protocols and policies.
CO-5	Implement security protocols and secure coding.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2
CO-2	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2
CO-3	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2
CO-4	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2
CO-5	2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Needham-Schroeder public-key protocol. Introduction to finite-state checking, SSL/TLS case study, IP security. Internet Key Exchange (IKE) protocol, Introduction to process algebra, Just Fast Keying (JFK) protocol, Security as observational equivalence. JFK protocol in applied pi calculus, Protocols for anonymity, Probabilistic model checking, Probabilistic contract signing protocols, Floyd-Hoare logic. Compositional protocol logic, Paulson's inductive method, Analyzing SET with the inductive method, Symbolic constraint solving, Formal definitions of security for symmetric ciphers, Formal model for secure key exchange, Simulatability-based proofs of protocol security, Probabilistic polynomial-time process calculus, Formal analysis of denial of service, Formal verification of routing protocols, Computational soundness of formal models, Multicast security, Spoofing and identity theft, Fair exchange and contract signing protocols, Trusted computing, Privacy preserving data mining, Automatic proofs of strong secrecy, Game-based verification of contract signing protocols, Wireless security, Game-based analysis of denial-of-service protection, Analysis of Internet voting protocols, Privacy-preserving graph algorithms, Universal composability framework, Analysis of Group Diffie-Hellman protocols

Learning Resources:

Text Books:

1. Oded Goldreich, "Foundations of Cryptography, Vol. I and II", Cambridge University Press, 2007.
2. Jonathan Katz and Yahuda Lindell, "Introduction to Modern Cryptography", CRC press, 2008.

Reference Books:

1. Van Oorschot, Paul Scott, A Vanstone, A J Menezes, "Handbook of Applied Cryptography", CRC Press, 2004.

CS1429 **3-0-0 (3)**

Internet of Things

Pre-Requisites: CS1106, CS1203, CS1305, CS1338

Course Outcomes:

CO-1	Analyze the protocol Stack for Internet of Things to address the heterogeneity in devices and networks.
CO-2	Develop smart IoT Applications using smart sensor devices and cloud systems.
CO-3	Development of smart mobile apps for societal applications.
CO-4	Design secure protocols for IoT systems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	2	3	-	-	-	-	-	-	2	2	2	2	2	2
CO-2	3	3	3	3	3	2	2	-	2	2	2	2	2	2	3	3
CO-3	3	3	3	3	3	2	2	-	2	2	2	2	2	3	2	3
CO-4	2	3	2	2	2	-	-	-	-	-	2	2	2	3	-	3

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction to IoT- Concepts, Services, Characteristics, Challenges and Applications of IoT- Smart City, Health-Care, Architecture of IoT - ITU, IWF, Integration of OT and IT technologies, IoT Data Flow,

IoT Protocols- Protocols Architecture of IoT**,** IoT Protocol Standards, Categorization of IoT protocols, Non-IP Network Technologies of IoT, IP Network Technologies, Service Discovery Protocols for IoT, Application Protocols, Protocol Stack of Wireless IoT.

IoT System Design- Components of IoT System, Communications Models of IoT, IoT Platforms- Open-Source & Proprietary Platforms

Sensor and Identification Technologies of IoT- Edge Devices of IoT- Sensors and Actuators, WSN, RFID, Integration of RFID and WSN Network Technologies

Connectivity of IoT-Wireless IoT, Non-IP Wireless Connectivity- 802.11ah, 802.15 technologies, Wireless Embedded Internet, and IoT Routing, IP Connectivity- Mobile IP, BLE over IPv6, ZigBee over IPv6, RFID over IPv6.

Service Technologies of IoT- Edge and Fog Computing in IoT, Cloud Computing in IoT.

IoT Security- Security Requirements, Vulnerabilities and Threat Analysis, Layered Attacker model and Blockchain in IoT Security

Learning Resources:

Text Books:

1. Olivier Hersent, “The Internet of Things Key Applications and Protocols”, Wiley, 2012.
2. Sudip Misra, “Introduction to IoT”, Cambridge University Press; First edition , 2021.

Reference Books:

1. Arshdeeep Bahga, Vijay Madisetti, "Internet of Things: A Hands-on Approach", Universities Press, 2015.
2. Raj Kamal, "Internet of Things: Architecture and Design Principles", McGraw Hill Education private limited, 2017.

CS1431 **3-0-0 (3)**

Intrusion Detection Systems

Pre-requisites: **None**

Course Outcomes:

CO-1	Explore the concepts of Network Protocol Analysis, and analyze information systems and networked systems.
CO-2	Identify system vulnerabilities and attacks, and troubleshoot system problems.
CO-3	Design and Develop intrusion detection systems& intrusion prevention systems and identify their signatures.
CO-4	Select technologies and tools for intrusion detection and intrusion prevention.
CO-5	Exercises and use cases for testing and evaluating various IDS techniques.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	-	1	-	-	-	1	-	2	-	-	-	2	-	2	1	-
CO-2	2	1	-	1	-	1	-	1	-	-	-	2	-	1	1	2
CO-3	1	1	1	1	2	-	1	-	1	2	-	-	1	1	-	2
CO-4	-	2	-	-	1	-	-	-	1	2	-	1	-	1	-	1
CO-5	2	-	1	2	2	-	1	-	-	-	1	-	2	-	1	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Firewall Planning and Design, Developing a Security Policy, System Configuration Strategies, Working with Proxy Servers and Application-Level Firewalls, Authenticating Users, Encryption and Firewalls. Intrusion detection, Audit, Internal and external threats to data, attacks, Information sources - Host based information sources, and Network based information sources; Types and classification of IDS. Intrusion Prevention Systems, Network Systems, Network IDs protocol based IDs, Hybrid IDs, Analysis schemes, models for intrusion analysis, techniques, mapping responses to policy vulnerability analysis, credential analysis non-credential analysis IDS using SNORT, NIDS, NNID and HIDS; Discovery and Detection: Identify IDS signatures such as anomaly detection, pattern matching and statistical analysis; Machine Learning models for IDS, Distributed IDS models; Architecture models of Intrusion Detection and intrusion prevention.

Learning Resources:

Text Books:

1. 1.Rafeeq Rehman : " Intrusion Detection with SNORT, Apache, MySQL, PHP and ACID," 1st Edition, Prentice Hall , 2003.
2. 2. Carl Endorf, Eugene Schultz and Jim Mellander, "Intrusion Detection & Prevention", Tata McGraw-Hill, 2004

Reference Books:

1. Christopher Kruegel,Fredrik Valeur, Giovanni Vigna: "Intrusion Detection and Correlation Challenges and Solutions", 1st Edition, Springer, 2005.
2. Stephen Northcutt, Judy Novak, "Network Intrusion Detection", New Riders Publishing, 2002.

CS1433 **3-0-0 (3)**

Quantum Computing

Pre-Requisites: MA1161, CS1101, CS1203

Course Outcomes:

CO-1	Understand quantum computing principles.
CO-2	Design and analyze quantum circuits and algorithms.
CO-3	Implement quantum circuits and algorithms using Qiskit.
CO-4	Understand fundamentals of quantum communication.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	1	–	–	–	–	–	–	–	–	1	2	1	–	–
CO-2	2	3	3	2	–	–	–	–	–	–	–	3	3	–	–	–
CO-3	2	2	2	2	–	–	–	–	–	–	–	–	2	2	–	1
CO-4	2	1	2	2	–	–	–	–	–	–	–	–	–	–	–	–

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Preliminaries: Review of linear algebra and complex numbers.

Quantum Computation: Introduction to qubits, Multiple qubits, Dirac notation, Bloch sphere, Reversable Gates, Basic single- qubit gates, Two-qubit gates, Measurements, Quantum circuits, Bell state circuit, No-cloning theorem, Teleportation, Amplitude Amplification, Superdense coding, Physical realizations of qubits and Qiskit.

Quantum Algorithms: Introduction to query complexity, Deutsch algorithm, Deutsch-Josza algorithm, Bernstein-Vajirani algorithm, Simon`s algorithm, Quantum Fourier Transform, Quantum phase estimation, Grover`s search Algorithm, Order finding using phase estimation, Shor`s algorithm, Quantum key distribution, BB84 protocol and HHL algorithm.

Learning Resources:

Text Books:

1. Michael A. Nielsen and Issac L. Chuang, Quantum Computation and Quantum Information,. Cambridge, 2010, 10th Anniversary Edition.
2. Noson S. Yanofsky and Mirco A. Mannucci, Quantum Computing for Computer Scientists, Cambridge University Press, 2008, 1st Edition.

Reference Books:

1. Eleanor G. Rieffel and Wolfgang H. Polak, Quantum Computing: A Gentle Introduction, MIT Press, 2011.
2. Quantum Computing in Practice with Qiskit(R) and IBM Quantum Experience(R): Practical recipes for quantum computer coding at the gate and algorithm level with Python, Hassi Norlén, Packt Publishing, 2020.

Other Suggested Readings:

1. IBM Quantum Learning Courses: https://learning.quantum.ibm.com/.
2. YouTube Link: Introduction to Quantum Computing Complete Course - Quantum Soar.

CS1435 **3-0-0 (3)**

Real Time Systems

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the use of multi-tasking techniques in real time systems.
CO-2	Evaluate the performance of soft and hard real time systems.
CO-3	Analyze multi task scheduling algorithms for periodic, aperiodic and sporadic tasks.
CO-4	Design real time operating systems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	-	2	-	-	-	-	-	-	-	-	2	-	-	-
CO-2	3	3	3	-	-	3	-	2	-	-	-	-	2	-	-	2
CO-3	2	2	3	2	-	3	-	-	-	-	-	-	2	-	-	2
CO-4	3	3	3	3	-	3	-	2	2	-	-	-	2	2	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Real-Time Systems, Typical Real-Time Applications, Hard Versus Soft Real-Time Systems, A Reference Model of Real-Time Systems.
Commonly Used Approaches to Hard Real-Time Scheduling, Clock-Driven Scheduling, Priority-Driven Scheduling of Periodic Tasks, Scheduling Aperiodic and Sporadic Jobs in Priority- Driven Systems.
Resources and Resource Access Control, Multiprocessor Scheduling and Resource Access Control.
Scheduling Flexible Computations and Tasks with Temporal Distance Constraints.
Real-Time Communications, Operating Systems.

Learning Resources:

Text Books:

1. Jane Liu, "*Real-Time Systems*", Prentice Hall, 2000.
2. Philip.A.Laplante, "*Real Time System Design and Analysis*", 3rd Edition, PHI, 2004.

CS1437 **3-0-0 (3)**

Secure Software Engineering

Pre-Requisites: **None**

Course Outcomes:

CO-1	Evaluate secure software engineering problems, including the specification, design, implementation, and testing of software systems.
CO-2	Elicit, analyze and specify security requirements through SRS.
CO-3	Design and Plan software solutions to security problems using various paradigms.
CO-4	Model the secure software systems using Unified Modeling Language Sec(UMLSec).
CO-5	Develop and apply testing strategies for Secure software applications.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	3	3	2	2	1	1	1	2	1	1	3	3	2	-
CO-2	2	2	1	2	2	2	1	1	1	2	1	1	2	2	3	-
CO-3	2	3	3	3	2	2	2	1	2	3	2	1	3	3	2	-
CO-4	2	2	3	2	3	2	2	1	2	2	2	1	3	2	2	-
CO-5	2	2	3	2	3	2	2	1	2	2	2	1	3	3	2	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Software assurance and software security, threats to software security, sources of software insecurity, benefits of detecting software security, managing secure software development
Defining properties of secure software, how to influence the security properties of software, how to assert and specify desired security properties.

Secure software Architecture and Design: Software security practices for architecture and design: Architectural risk analysis, software security knowledge for Architecture and Design: security principles, security guidelines, and attack patterns, secure design through threat modeling, Writing secure software code: Secure coding techniques, Secure Programming: Data validation.

Secure Programming: Using Cryptography Securely, Creating a Software Security Programs.
Secure Coding and Testing: code analysis- source code review, coding practices, static analysis, software security testing, security testing consideration through SDLC

Learning Resources:

Text Books:

1. Julia H Allen, Sean J Barnum, Robert J Ellison, Gary McGraw, Nancy R Mead, "Software Security Engineering: A Guide for Project Managers", Addison Wesley, 2008.
2. Ross J Anderson, "Security Engineering: A Guide to Building Dependable Distributed Systems", 2nd Edition, Wiley, 2008.

Reference Books:

1. Howard, M. and LeBlanc, D., "Writing Secure Code", 2nd Edition, Microsoft Press, 2003.

CS1439 **3-0-0 (3)**

Security and Privacy

Pre-Requisites: **None**

Course Outcomes:

CO-1	Evaluate the risks and vulnerabilities in protocols/Standards.
CO-2	Design and security analysis of cryptographic algorithms.
CO-3	Design and security analysis of authentication, message integrity and authenticated encryption protocols.
CO-4	Develop techniques for Privacy preserving Data analysis.
CO-5	Compute lower bounds for differential privacy.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	2	2	2	3	2	2	1	2	2	2	1	2	2	2	2
CO-2	3	2	3	2	2	2	2	1	3	2	2	1	3	3	2	2
CO-3	3	2	3	2	2	2	2	1	3	2	2	1	3	3	2	2
CO-4	3	3	3	3	3	2	2	1	3	2	3	1	2	2	2	3
CO-5	3	3	3	3	3	2	2	1	3	2	3	1	2	2	3	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Security – risks, threats and vulnerabilities, Cryptography. Symmetric key Cryptography – Encryption, Block ciphers, Chosen plaintext attacks, Stream Ciphers – One-time Pad (OTP), Perfect secrecy, Pseudo-random generators (PRG), Attacks on stream ciphers and OTP, Real world stream ciphers, Semantic security, Case Study- RC4, Salsa 20, CSS in DVD encryption, A5 in GSM, Block ciphers- DES, attacks, AES, Block ciphers from PRG, Modes of operation – one-time key and many-time keys, CBC, CTR modes,

Message Integrity – MAC, MAC based on PRF, NMAC, PMAC, Collision resistance – Birthday attack, Merkle-Damgard construction, HMC, Case study:SHA-256, Authenticated encryption, Key exchange algorithms, Public key cryptosystems – Public key tools, Public key encryption, Chosen ciphertext secure public- key encryption, Digital signature, Fast hash based signatures, RSA, ElGamal, Elliptic curve cryptosystems – PKC, key exchange, IBE, Analysis of number theoretic assumptions, Case studies – HTTPS – SSL/TLS, SSH, IPSec, 802.11i WPA.

Protocols - Protocols for identification and login, Identification and signatures from sigma protocols, Proving properties in zero-knowledge, Authenticated key exchange, Key establishment with online trusted third parties, Two-party and multi-party secure computation

Privacy preserving Data analysis - Basis Techniques - Randomized response, the Laplace mechanism, the exponential mechanism, Composition theorems, and sparse vector technique.

Releasing Linear Queries with Correlated Error, Mechanisms via α-nets, Iterative construction mechanism, Boosting for queries algorithm, Stability and privacy, Lower bounds for differential privacy. System design and analysis – Survivable distributed storage system, Electronic voting system, Digital Cash, Bit coin.

Learning Resources:

Text Books:

1. J. Katz and Y. Lindell, "Introduction to Modern Cryptography", CRC press, 2008.
2. C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy", now Publishers, 2014.

Reference Books:

1. Van Oorschot, Paul Scott, A Vanstone, A J Menezes, "Handbook of Applied Cryptography", CRC Press, 2004.

CS1441 **3-0-0 (3)**

Service Oriented Architecture

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand software-oriented architectures.
CO-2	Design medium scale software project development using SOA principles.
CO-3	Develop SOA messages from business use cases.
CO-4	Design and implementation of modern SOA and SOA-specific methodologies, technologies and standards.
CO-5	Create composite services by applying composition style.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	3	2	3	1	2	2	1	2	2	2	1	3	2	3	2
CO-2	2	2	3	2	2	2	2	1	3	3	3	1	2	3	2	3
CO-3	3	2	3	2	2	2	3	1	3	3	3	1	2	3	2	3
CO-4	3	2	3	2	2	2	2	1	3	3	3	1	2	3	2	3
CO-5	3	2	3	2	2	2	2	1	3	3	3	1	2	3	2	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to SOA, Evolution of SOA: Fundamental SOA; Common Characteristics of contemporary SOA; Common tangible benefits of SOA; An SOA timeline (from XML to Web services to SOA); The continuing evolution of SOA (Standards organizations and Contributing vendors); The roots of SOA (comparing SOA to Past architectures).

Web Services and Primitive SOA: The Web services framework· Services (as Web services); Service descriptions (with WSDL); Messaging (with SOAP).

Web Services and Contemporary SOA – I Message exchange patterns; Service activity; Coordination; Atomic Transactions; Business activities; Orchestration; Choreography. Web Services and Contemporary SOA-2: Addressing; Reliable messaging; Correlation; Polices; Metadata exchange; Security; Notification and eventing.

Principles of Service - Orientation: Services orientation and the enterprise; Anatomy of a service oriented architecture; Common Principles of Service orientation; how service orientation principles interrelate; Service orientation and object orientation; Native Web service support for service orientation principles.

Service Layers: Service orientation and contemporary SOA; Service layer abstraction; Application service layer, Business service layer, Orchestration service layer; Agnostic services; Service layer configuration scenarios.

Business Process Design: WS-BPEL language basics; WS Coordination overview; Service oriented business process design; WS addressing language basics; WS Reliable Messaging language basics.

SOA Platforms: SOA platform basics; SOA support in J2EE; SOA support in. ET; Integration considerations

Learning Resources:

Text Books:

1. Thomas Erl, "Service-Oriented Architecture: Concepts, Technology and Design", Prentice Hall Publication, 2005.
2. Michael Rosen, Boris Lublinsky, "Applied SOA Service Oriented Architecture and Design Strategies", Wiely India Edition, 2008.

CS1443 **3-0-0 (3)**

Software Reliability Techniques

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand Software Reliability during different phases of Software Development Life Cycle.
CO-2	Analyze Software Reliability parameters using Markovian Modelling.
CO-3	Estimate Software Reliability parameters using Maximum Likelihood and Least Square Method.
CO-4	Evaluate performance of Binomial-Type, Poison-Type and Markovian Models.
CO-5	Predict Software Reliability using Intelligent Techniques.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	3	2	2	-	-	-	2	2	2	2	-	3	-	2
CO-2	3	2	3	2	2	-	-	-	2	2	2	2	-	3	-	3
CO-3	3	2	3	2	2	-	-	-	2	2	2	2	-	3	-	3
CO-4	3	2	3	2	2	-	-	-	2	2	2	2	-	3	-	2
CO-5	3	2	3	2	2	-	-	-	2	2	2	2	-	3	-	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Software Reliability: The need for Software Reliability, Some Basic Concepts, Software Reliability and Hardware Reliability, Availability, Modelling and General Model Characteristics.

Software Reliability Modeling: Halstead's Software Metric, McCabe's Cyclomatic Complexity Metric, Error Seeding Models, Failure Rate Models, Curve Fitting Models, Reliability Growth Models, Markov Structure Models, Time Series Models, Non-homogeneous Poison Process Models.

Markovian Models: General Concepts, General Poison-Type Models, Binomial -Type Models, Poison- Type Models, Comparison of Binomial-Type and Poison-Type Models, Fault Reduction Factor for Poison-Type Models.

Descriptions of Specific Models: Finite Failure Category Models, Infinite Failure Category Models. Parameter Estimation: Maximum Likelihood Estimation, Least Squares Estimation, Bayesian Inference.

Comparison of Software Reliability Models: Comparison Criteria, Comparison of Predictive Validity of Model Groups, Evaluation of other Criteria.

Software Reliability Prediction: Problems associated with different Software Reliability Models, Software Reliability prediction parameters, Intelligent Techniques for Software Reliability Prediction.

Learning Resources:

Text Books:

1. M. Xie, "Software Reliability Modelling", World Scientific; 1991.
2. John D. Musa, Anthony Iannino, Kazuhira Okumoto, "Software Reliability Measurement, Prediction, Application", McGraw-Hill Book Company; 1987.

Reference Books:

1. Hoang Pham, "System Software Reliability", Springer; 2005
2. Hamdy A Taha , Operations Research-An Introduction, 2017 (Chs 1-8, 12, 14, 17) , 9th Edition.

CS1422 **3-0-0 (3)**

Advanced Theoretical Computer Science

Pre-Requisites: **None**

Course Outcomes:

CO-1	Identify and explain fundamental mathematical constraints for developing algorithms to solve problems in high dimensional space.
CO-2	Develop methods to study how to draw good samples efficiently and how to estimate statistical and linear algebra quantities, with such samples.
CO-3	Apply learning models and algorithms with provable guarantees on learning error and time.
CO-4	Build models to understand and to capture essential properties of large structures, like the web and social networks

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	2	2	1	-	1	-	-	1	-	-	3	1	2	-
CO-2	2	3	2	1	3	-	-	1	-	1	-	1	3	1	3	-
CO-3	2	2	2	2	3	1	1	-	1	1	-	-	3	3	2	1
CO-4	2	3	3	3	3	1	-	1	1	-	-	1	3	3	2	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

High-Dimensional Space - The Law of Large Numbers, The Geometry of High Dimensions, Properties of the Unit Ball, Volume of the Unit Ball, Volume Near the Equator, Generating Points Uniformly at Random from a Ball, Gaussians in High Dimension, Random Projection and Johnson-Lindenstrauss Lemma, Separating Gaussians, Fitting a Spherical Gaussian to Data.

Best-Fit Subspaces and Singular Value Decomposition (SVD) - Singular Vectors, Best Rank-k Approximations, Left Singular Vectors, Power Method for Singular Value Decomposition, A Faster Method, Singular Vectors and Eigenvectors, Applications of Singular Value Decomposition, Centering Data, Principal Component Analysis, Clustering a Mixture of Spherical Gaussians, Ranking Documents and Web Pages, Application of SVD to a Discrete Optimization Problem.

Random Walks and Markov Chains - Stationary Distribution, Markov Chain Monte, Metropolis-Hasting Algorithm, Gibbs Sampling, Areas and Volumes, Convergence of Random Walks on Undirected Graphs, Using Normalized Conductance to Prove Convergence, Electrical Networks and Random Walks, Random Walks on Undirected Graphs with Unit Edge Weights, Random Walks in Euclidean Space, The Web as a Markov Chain.

Machine Learning - The Perceptron algorithm, Kernel Functions, Generalizing to New Data, Overfitting and Uniform, illustrative Examples and Occam's Razor, Learning Disjunctions, Occam's, Application: Learning Decision Trees, Regularization: Penalizing Complexity, Online, Online to Batch Conversion, Support-Vector Machines, VC-Dimension, Strong and Weak Learning – Boosting, Stochastic Gradient Descent, Combining (Sleeping) Expert Advice, Deep Learning, Further Current Directions.

Algorithms for Massive Data Problems: Streaming, Sketching, and Sampling - Frequency Moments of Data Streams, Matrix Algorithms using, Sketches of Documents

Clustering - Two General Assumptions on the Form of Clusters, k-Center, Spectral Clustering, Approximation Stability, High-Density Clusters, Kernel Methods, Recursive Clustering based on Sparse Cuts, Dense Submatrices and Communities, Community Finding and Graph Partitioning, Spectral clustering applied to social network graphs

Random Graphs - The G (n; p) Model, Phase Transitions, Giant, Cycles, Phase Transitions for Increasing, Branching Processes, CNF-SAT, Nonuniform Models of Random Growth Models, Small World Graphs Topic Models, Nonnegative Matrix Factorization, Hidden Markov Models.

Graphical Models - Topic Models, An Idealized Model, Nonnegative Matrix Factorization – NMF, NMF with Anchor Terms, Hard and Soft Clustering, The Latent Dirichlet Allocation Model for Topic, The Dominant Admixture, Finding the Term-Topic Matrix, Hidden Markov, Graphical Models and Belief, Bayesian or Belief, Markov Random, Factor, Tree Algorithms, Message Passing in General, Graphs with a Single Cycle, Belief Update in Networks with a Single, Maximum Weight Matching, Warning.

Other Topics - Ranking and Social Choice, Compressed Sensing and Sparse, Applications, Uncertainty Principle

Learning Resources:

Text Books:

1. Brian Steele, John Chandler, Swarna Reddy, "Algorithms for Data Science", Springer, 2016
2. Noga Alon and Joel H Spenser, "Probabilistic Method", Third Edition,. John Wiley & Sons, 2008.

Reference Books:

1. Rajeev Motwani and Prabhakar Raghavan, "Randomized Algorithms", Cambridge University Press, 1995.

CS1424 **3-0-0 (3)**

Agri Bioinformatics

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the theoretical basis behind Agriinformatics and bioinformatics.
CO-2	Compute homologues, analyze sequence alignment and similarity.
CO-3	Understand the Management Information System, Decision support system, Expert system.
CO-4	Analyze various gene sequences, gene prediction, and categories of gene prediction.
CO-5	Determine and model biological information and apply to the solution of biological problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	-	2	-	3	-	-	1	-	-	-	-	1	1	2	-
CO-2	-	2	3	-	-	1	2	3	-	2	-	-	1	1	3	1
CO-3	3	-	2	-	3	-	-	-	3	-	-	3	-	2	3	1
CO-4	2	3	2	3	2	-	-	3	-	1	2	-	1	1	3	1
CO-5	-	1	3	-	1	-	3	-	2	2	3	3	1	2	2	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Bioinformatics: What is a Database?, Types of Databases, Biological Databases, Biological Databases, Information Retrieval from Biological Databases.

Pair wise Sequence Alignment: Evolutionary Basics, Sequence homology versus similarity, Sequence similarity versus Identity, Scoring Matrices, Statistical Significance of Sequence alignment.

Database similarity searching: Unique requirement of Database searching, Heuristic Database searching.

Basic alignment search tool: Comparison of FASTA and BLAST, Multiple Sequence Alignment, Scoring Function, Gene Prediction, Categories of gene prediction, Gene prediction in prokaryotes and Eukaryotes.

Introduction to Agri-informatics: Importance of Agriculture to Development and the Contribution of ICTs, Critical Information Flows and the Agricultural Knowledge System, The Feasibility of ICT in Rural Areas, Financial Aspects: Up-front Costs and Long-term Benefits of ICT Approaches, Technical Aspects of ICT Feasibility in Rural Areas, Promising Emerging Technologies, Some major ICT trends, Some high impact innovations, Hardware and Connectivity, Ubiquitous Telecommunication Infrastructure, Utility or Cloud Computing, Software and Content Management, Interaction with Biology, Biotechnology, Nanotechnology, Materials Science and ICTs, Information and Communication Technology in Agricultural Development, Convergence of ICT with agricultural development, Areas of IT convergence, Drivers of ICT in Agriculture, Low-Cost and Pervasive Connectivity, Adaptable and More Affordable Tools, New Business Models and Public-Private Partnerships, Democratization of Information, the Open Access Movement and Social Media, Use Appropriate Technologies, Making ICT Infrastructure, Appliances and Services, Accessible and Affordable in Rural Areas, Key Challenges and Enablers, Partnerships, Regulation and Policy Challenges, Infrastructure, Local Loop or Last Mile Connectivity, Appliances, Services, Anytime, Anywhere: Mobile Devices and Services, Applications in Agriculture, Information Systems in Agriculture: Introduction, Information systems and communication networks, Management Information Systems, Internal Information Systems, External Information Systems, Farm Management Information System (FMIS), Farm Management Information System. Decision Support Systems: Introduction, Historical Overview of Decision Support Systems (DSS), Future trends, Decision Support Systems in Agriculture. Expert Systems and Applications: Introduction, what is an Expert System, Expert Systems and Human Reasoning, Structural

Nature of Expert Systems, Characteristics of Expert System Applications, Knowledge Acquisition, Knowledge Representation, Explanation, Expert System Strengths and Limitations, Extensions to Expert Systems.

Learning Resources:

Text Books:

1. Jin Xiong, "Essential Bioinformatics", 1st Edition, Cambridge University Press, 2011.
2. Arthur M Lesk, "Introduction to Bioinformatics", 2nd Edition, Oxford University Press, 2007.

Reference Books:

1. Zhumur Ghosh “Bioinformatics” 2nd Edition, Oxford University Press, 2009.
2. Herdon, Miklos “Agriinformatics” 2nd Edition Cambridge University Press 2013.

CS1426 **3-0-0 (3)**

Algorithmic Game Theory

Pre-Requisites: **None**

Course Outcomes:

CO-1	Computation of equilibrium based on complete and incomplete information about the players.
CO-2	Develop Combinatorial Algorithms for Market Equilibria.
CO-3	Analyze mechanism design.
CO-4	Design and analyze combinatorial auctions.
CO-5	Design of Scalable Resource Allocation Mechanisms.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	2	3	2	3	1	3	1	3	3	3	1	3	3	2	3
CO-2	3	2	3	2	3	1	3	1	3	3	3	1	3	3	2	3
CO-3	2	3	2	3	2	1	3	1	2	2	2	1	3	2	3	2
CO-4	3	2	3	2	2	1	3	1	3	3	3	1	3	3	2	3
CO-5	3	2	3	2	2	1	3	1	3	3	3	1	3	3	2	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Basic Solution Concepts and Computational Issues - Games, Strategies, Costs, and Payoffs, Basic Solution Concepts, Finding Equilibria and Learning in Games, Refinement of Nash: Games with Turns and Subgame Perfect Equilibrium, Nash Equilibrium without Full Information: Bayesian Games, Cooperative Games, Markets and Their Algorithmic Issues

The Complexity of Finding Nash Equilibria - NP-Completeness, the Lemke–Howson Algorithm, the Class PPAD, Succinct Representations of Games, Reduction, Correlated Equilibria

Equilibrium Computation for Two-Player Games in Strategic and Extensive Form - Bimatrix Games and the Best Response Condition, Equilibria via Labeled Polytope, Lemke–Howson Algorithm, Integer Pivoting, Degenerate Games, Extensive Games and Their Strategic Form, Subgame Perfect Equilibria, Reduced Strategic Form, Sequence Form, Computing Equilibria with the Sequence Form Learning, Regret Minimization, and Equilibria - External Regret Minimization, Regret Minimization and Game Theory, Generic Reduction from External to Swap Regret, The Partial Information Model, On Convergence of Regret-Minimizing Strategies to Nash Equilibrium in Routing Games Combinatorial Algorithms for Market Equilibria - Fisher's Linear Case and the Eisenberg–Gale Convex Program, Checking If Given Prices Are Equilibrium Prices, Primal-Dual Schema in the Enhanced Setting, Tight Sets and the Invariant, Balanced Flows, Linear Case of the Arrow–Debreu Model, Auction-Based Algorithm, Resource Allocation Markets, Algorithm for Single-Source Multiple-Sink Markets

Computation of Market Equilibria by Convex Programming - Fisher Model with Homogeneous Consumers, Exchange Economies Satisfying WGS, Specific Utility Functions, Models with Production Graphical Games - Computing Nash Equilibria in Tree Graphical Games, Graphical Games and Correlated Equilibria, Graphical Exchange Economies.

Cryptography and Game Theory - Cryptographic Notions and Settings, Game Theory Notions and Settings, Contrasting MPC and Games, Cryptographic Influences on Game Theory, Game Theoretic Influences on Cryptography

Mechanism Design - Social Choice, Mechanisms with Money, Implementation in Dominant Strategies, Characterizations of Incentive Compatible Mechanisms, Bayesian–Nash Implementation Combinatorial Auctions - The Single-Minded Case, Walrasian Equilibrium and the LP Relaxation, Bidding Languages, Iterative Auctions: The Query Model, Communication Complexity, Ascending Auctions

Inefficiency of Equilibria - Fundamental Network Examples, Inefficiency of Equilibria as a Design Metric

Routing Games - Models and Examples, Existence, Uniqueness, and Potential Functions, Price of Anarchy of Selfish Routing, Reducing the Price of

Network Formation Games and the Potential Function Method - Local Connection Game, Potential

Games and a Global Connection Game, Facility Location
Selfish Load Balancing - Pure Equilibria for Identical Machines, Pure Equilibria for Uniformly Related Machines, Mixed Equilibria on Identical Machines, Mixed Equilibria on Uniformly Related Machines

Learning Resources:

Text Books:

1. N. Nisan, T. Roughgarden, E. Tardos, V.V. Vazirani, "Algorithmic Game Theory", Cambridge University Press, 2007..
2. Y. Shoham and K. Leyton-Brown, "Multi-agent Systems: Algorithmic, Game-Theoretic, and Logical Foundations", Cambridge University Press, 2009.

Reference Books:

1. T. Roughgarden, "Twenty Lectures on Algorithmic Game Theory", Cambridge University. Press, 2016.

CS1428 **3-0-0 (3)**

Algorithmic Techniques for Bigdata

Pre-Requisites: **None**

Course Outcomes:

CO-1	Identify various statistical information required to be drawn from Big Data.
CO-2	Develop mathematical models for Big Data.
CO-3	Design sub-linear randomized algorithms and analyze their quality of answer.
CO-4	Derive lower bounds for randomized algorithms.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	3	3	3	2	2	1	2	1	1	-	2	3	3	2
CO-2	3	2	2	3	3	3	3	1	1	1	1	-	2	3	3	1
CO-3	3	3	3	3	3	3	3	2	1	1	1	-	3	2	3	3
CO-4	2	2	2	2	2	2	1	2	1	1	1	-	2	1	2	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Preliminaries: The Data Model, Basic Setup, Quality of an Algorithm's Answer, Variations of the Basic Setup.

Finding Frequent Items Deterministically: Problem, Frequency Estimation, The Misra–Gries Algorithm, Analysis of the Algorithm,

Estimating the Number of Distinct Elements: Problem, Tidemark Algorithm, Quality of the Algorithm's Estimate, Median Trick, A Better Estimate for Distinct Elements - BJKST Algorithm, Analysis: Space Complexity, Analysis: The Quality of the Estimate, Optimality.

Approximate Counting: Problem, Algorithm, Quality of the Estimate, Median-of-Means Improvement

Finding Frequent Items via (Linear) Sketching: Problem, Sketches and Linear Sketches, Count Sketch, Quality of the Basic Sketch's Estimate, Count-Min Sketch, Quality of the Algorithm's Estimate, Comparison of Frequency Estimation Methods.

Estimating Frequency Moments: Background and Motivation, AMS Estimator for Fk, Analysis of the Basic Estimator, Estimator and Space Bound, Soft-O Notation, Tug-of-War Sketch, Sketch, Quality of the Estimate Geometric Interpretation.

Estimating Norms Using Stable Distributions: Median of a Distribution and its Estimation Accuracy of the Estimate

Weight-Based Sampling: Problem, Lo-Sampling, Idealized Algorithm, Quality of the Output. L2 Sampling - L2-sampling Algorithm, Analysis.

Finding the Median: Problem, Munro–Paterson Algorithm, Computing a Core, Utilizing a Core, Analysis: Pass/Space Tradeoff.

Geometric Streams and Coresets: Extent Measures and Minimum Enclosing Ball, Coresets and Their Properties, Coreset for MEB , Data Stream Algorithm for Coreset, Construction.

Metric Streams and Clustering: Metric Spaces, Cost of a Clustering: Summarization Costs, Doubling Algorithm, Metric Costs and Threshold Algorithms, Guha's Cascading Algorithm, Space Bounds, Quality of the Summary.

Graph Streams: Basic Algorithms, Streams that Describe Graphs, Semi-Streaming Space Bounds , Connectedness Problem, Bipartiteness Problem, Shortest Paths and Distance Estimation via

Spanners, Quality of the Estimate, Space Complexity: High-Girth Graphs and the Size of a Spanner .

Finding Maximum Matchings: Maximum Cardinality Matching, Maximum Weight Matching .

Graph Sketching: Value of Boundary Edges, Testing Connectivity Using Boundary Edges , Testing Bipartiteness, The AGM Sketch: Producing a Boundary Edge, Counting Triangles, Sampling-Based Algorithm, Sketch-Based Algorithm.

Communication Complexity and Lower Bounds: Communication Games, Protocols, Complexity, Specific Two-Player Communication Games, Definitions, Results and Some Proofs: Deterministic Case, Proofs: Randomized Case, Data Streaming Lower Bounds, Lower Bound for Majority, Lower Bound for Frequency Estimation, Further Reductions and Lower Bounds - Importance of Randomization, Multi-Pass Lower Bounds for Randomized Algorithms, Graph Problems, importance of approximation, Space verses approximation quality.

Learning Resources:

Text Books:

1. Oded Goldreich, "Introduction to Property Testing", Cambridge University Press, 2017.
2. Jure Leskovec, Anand Rajaraman, Jeff Ullman, "Mining of Massive Datasets", Dreamtech Press, 2016

Reference Books:

1. Amit Chakrabarthy, "Data Stream Algorithms", Lecture Notes, Dartmouth College, 2020
2. Relevant research papers

CS1430 **3-0-0 (3)**

Computational Neuro Science

Pre-Requisites: **None**

Course Outcomes:

CO-1	Simulate simple models of neurons, and their populations using computing languages.
CO-2	Understand the working of neural networks to store and process information.
CO-3	Construct computational models for hypothesis testing.
CO-4	Perform literature surveys and evaluate evidence for the impact of neuroscience on specific computational and cognitive neuroscience theories.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	3	3	3	2	–	–	–	–	2	2	3	3	–	1
CO-2	2	2	1	3	3	3	3	3	1	1	–	–	3	3	2	2
CO-3	1	1	3	3	3	3	2	1	–	–	2	2	3	3	3	2
CO-4	–	–	2	2	–	2	–	2	1	3	3	3	1	1	1	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Analyzing And Modeling Neural Responses: Introduction-Properties of Neurons, Recording Neuronal Responses, From Stimulus to Response Spike Trains and Firing Rates-Measuring Firing Rates, Tuning Curves, Spike-Count Variability What Makes a Neuron Fire?-Describing the Stimulus, The Spike-Triggered Average, White-Noise Stimuli, Multiple-Spike-Triggered Averages and Spike-Triggered correlations, Spike Train Statistics-The Homogeneous Poisson Process, The Spike-Train Autocorrelation Function, The Inhomogeneous Poisson Process, The Poisson Spike Generator, Comparison with Data, The Neural Code-Independent-Spike, Independent Neuron and Correlation Codes, Temporal Codes

Information Theory: Entropy and Mutual Information, Entropy, Mutual Information, Entropy and Mutual Information for Continuous Variables Information and Entropy Maximization-Entropy Maximization for a Single Neuron, Populations of Neurons, The Whitening Filter, Filtering Input Noise, Temporal Processing in the LGN, Cortical Coding Entropy and Information for Spike Trains

Modeling Neurons And Networks: Levels of Neuron Modeling-Levels of Neuron Modeling, Single-Compartment Models-Integrate-and-Fire Models, Spike-Rate Adaptation and Refractoriness, Hodgkin-Huxley model, Firing-rate Models- Feed forward Networks-Neural Coordinate Transformations, Recurrent Networks, Network Stability, Associative Memory, Excitatory-Inhibitory Networks-Homogeneous Excitatory and Inhibitory Populations, Phase Plane Methods and Stability Analysis, The Olfactory Bulb, Oscillatory Amplification, Stochastic Networks

Plasticity and Learning: Synaptic Plasticity Rules-The Basic Hebb Rule, the Covariance Rule, the BCM Rule, Synaptic Normalization, Subtractive Normalization, Multiplicative Normalization and the Oja Rule, Timing-Based Rules, Unsupervised Learning, Supervised Learning Supervised Hebbian Learning, Classification and the Perceptron, Function Approximation Supervised Error-Correcting Rules, the Perceptron Learning Rule, the Delta Rule-Contrastive Hebbian Learning.

Learning Resources:

Text Books:

1. Peter Dayan and L F Abbott, Theoretical Neuroscience, MIT Press, 2001.
2. Christopher Koeli, Electrophysics of Neuron, MIT Press, 2004, 1st Edition

Reference Books:

1. Computational Neuroscience: A Comprehensive Approach, Chapman and Hall, 2020, 3rd edition.

2. Wulfram Gerstner, Werner M. Kistler, Richard Naud and Liam Paninski, Neuronal Dynamics From single neurons to networks and models of cognition, Cambridge University Press, 1st edition.

Other Suggested Readings:

1. NPTEL Course: BT6270-Introduction to Computational Neuroscience by Prof. V Srinivasa Chakravarthy, IIT Madras.

CS1432 **3-0-0 (3)**

Cyber Laws and Intellectual Property Rights

Pre-Requisites: **CS1305, CS1302**

Course Outcomes:

CO-1	Understand cyberspace, the issues therein and the need for a cyber law.
CO-2	Understand facets of India IT act in addressing e-trade and e-governance.
CO-3	Understanding of issues and problems arising out of online transactions.
CO-4	Understanding crimes with case law.
CO-5	Understand of intellectual property issues and development of the law in this regard.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	-	-	-	-	1	-	-	-	1	-	-	1	1	-	-	-
CO-2	-	1	-	-	2	-	2	-	-	-	1	-	-	-	2	-
CO-3	-	-	2	-	1	-	-	-	1	1	1	1	1	-	2	1
CO-4	-	-	-	-	-	-	2	-	-	1	-	-	-	-	1	-
CO-5	-	-	1	1	1	-	-	-	1	-	2	-	-	2	1	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Cyber Space- Fundamental definitions -Interface of Technology and Law – Jurisprudence and-Jurisdiction in Cyber Space - Indian Context of Jurisdiction -Enforcement agencies – Need for IT act - UNCITRAL – E-Commerce basics; Information Technology Act, 2000 - Aims and Objects — Overview of the Act – Jurisdiction -Electronic; Governance – Legal Recognition of Electronic Records and Electronic Evidence - Digital Signature Certificates - Securing Electronic records and secure digital signatures - Duties of Subscribers - Role of Certifying Authorities - Regulators under the Act -The Cyber Regulations Appellate Tribunal - Internet Service Providers and their Liability– Powers of Police under the Act – Impact of the Act on other Laws; Cyber Crimes -Meaning of Cyber Crimes –Different Kinds of Cyber crimes – Cyber crimes under IPC; Cr.P.C and Indian Evidence Law - Cyber crimes under the Information Technology Act,2000 - Cyber crimes under International Law - Hacking Child Pornography, Cyber Stalking, Denial of service Attack, Virus Dissemination, Software Piracy, Internet Relay Chat (IRC) Crime, Credit Card Fraud, Net Extortion, Phishing etc - CyberTerrorism- Violation of Privacy on Internet - Data Protection and Privacy – Indian Court cases; Intellectual Property Rights – Copyrights- Software – Copyrights vs Patents debate - Authorship and Assignment Issues - Copyright in Internet - Multimedia and Copyright issues - Software Piracy - Trademarks - Trademarks in Internet – Copyright and Trademark cases, Patents - Understanding Patents - European Position on Computer related Patents, Legal position on Computer related Patents - Indian Position on Patents – Case Law, Domain names -registration - Domain Name Disputes-Cyber Squatting-IPR cases

Learning Resources:

Text Books:

1. Justice Yatindra Singh, Cyber Laws, Universal Law Publishing Co., New Delhi, 2010
2. Farooq Ahmed, Cyber Law in India, New Era publications, New Delhi, 2005

Reference Books:

1. S.R.Myneni, Information Technology Law(Cyber Laws), Asia Law House, Hyderabad, 2014
2. Chris Reed, Internet Law-Text and Materials, Cambridge University Press, 2004

CS1434 **3-0-0 (3)**

Cyber Security

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the cyber security fundamentals.
CO-2	Identify & Evaluate cyber security threats and vulnerabilities in Information Systems and apply security measures to real time scenarios.
CO-3	Design and implement appropriate security techniques and cyber policies to protect computers and digital information.
CO-4	Identify common trade-offs and compromises that are made in the design and development process of Information Systems.
CO-5	Demonstrate the use of standards and cyber laws to enhance information security in the development process and infrastructure protection.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	-	1	-	-	2	-	2	-	-	-	1	-	-	-	2	-
CO-2	-	-	2	-	1	-	-	-	1	1	1	1	1	-	2	1
CO-3	-	-	-	-	-	-	-	2	-	-	1	-	-	-	-	1
CO-4	1	-	-	1	1	1	-	-	-	1	-	2	-	-	2	1
CO-5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Cyber Security Fundamentals: Network and Security Concepts-Information Assurance Fundamentals, Basic Cryptography, Symmetric and Asymmetric Encryption, Public Key Encryption, The Domain Name System (DNS), Firewalls, Virtualization, Radio-Frequency Identification.

Threats and vulnerabilities: Types of Threats- Malware, Phishing, Ransomware, Adware and Spyware, Trojan, Virus, Worms, Man-in-the-middle-attack, Scareware, Distributed Denial-Of- Service Attack, Rootkits, click-fraud. Vulnerability-Shellcode, Integer Overflow Vulnerabilities, Buffer Overflows, SQL Injection.

Defense and mitigation measures: Anti-virus scanners, static and dynamic methods, anti- analysis, evading obfuscations and run-time attacks.

Cyber Forensics: Memory and network Forensics for Windows and Linux internals, Forensic tools, OS hardening and RAM dump analysis, data acquisition, data extraction, volatility analyses for OS artifacts and other information. Automated malicious code analysis.

Cybersecurity law and Regulations: Introduction, Cyber Warfare, Deception in the Cyber World, Legal Framework of Cyber Security.

Learning Resources:

Text Books:

1. James Graham, Richard Howard, Ryan Olson, "Cyber Security Essentials", CRC Press, 2016.
2. David Salomon, Foundations of Computer Security, Springer, 2006.

Reference Books:

1. MarttiLehto, PekkaNeittaanmäki, Cyber Security: Analytics, Technology and Automation, Springer, 2015.

CS1436 **3-0-0 (3)**

Fog and Edge Computing

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the basic requirements of fog and edge computing.
CO-2	Understand the key architectures and applications in fog and edge computing.
CO-3	Perform fog and edge computing services.
CO-4	Implement software using standard open-source fog and edge computing software for data analytics.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	3	2		3	-	-	-	-	-	-	-	-	-	-
CO-2	2	2	3	2		3	-	-	-	-	-	-	-	-	-	-
CO-3	2	2	3	2		3	-	-	-	-	-	-	-	-	-	-
CO-4	2	2	3	2		3	-	-	-	-	-	-	-	-	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Fog Computing, Limitation of Cloud Computing, Differences between Cloud and Fog Computing, Advantages, Business Models, Architecture, Opportunities and Challenges, Challenges in Fog Resources: Taxonomy and Characteristics, Resource Management Challenge, Optimization Challenges, Miscellaneous Challenges, IoT and Fog: Programming Paradigms, Research Challenges and Research Directions, Fog Protocols, Management and Orchestration of Network Slices in 5G, Fog, Edge and Clouds, Data Management and Analysis in Fog Computing, Case Studies. Introduction to Edge Computing, Origins of Edge, Edge Helping Low-End IoT Nodes, Architecture, Edge Helping Higher-Capability Mobile Devices: Mobile Offloading, Edge Helping the Cloud, Edge for Augmented Reality, Data Processing on the Edge, Dispersed Learning with Edge/Fog Computing, Video Analytics on the Edge, Edge Computing Applications.

Learning Resources:

Text Books:

1. Rajkumar Buyya, Satish Narayana Srirama, “Fog and Edge Computing”, Wiley Publications, 2019.
2. Wei Change and Jie Wu, “Fog/Edge Computing for Security, Privacy and Applications”, Springer, 2021.

CS1438 **3-0-0 (3)**

Formal Methods in Software Engineering

Pre-Requisites: **None**

Course Outcomes:

CO-1	Model the state of a software component using the unifying concept of mathematical relation.
CO-2	Design of automatic verification tools to establish the validity of a given software property.
CO-3	Apply automatic software verification tools based on model checking.
CO-4	Design tools for the deductive verification of programs annotated with contracts.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	3	3	3	2	2	2	1	3	2	3	1	3	3	2	3
CO-2	3	2	3	2	2	2	2	1	3	3	3	1	3	3	2	3
CO-3	3	2	3	2	3	2	2	1	3	2	3	1	3	3	2	3
CO-4	3	2	3	2	3	2	2	1	3	3	3	1	3	3	2	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Specification and Modeling: Introduction: the role of formal methods in software engineering; the role of abstraction in formal modeling; propositional and first-order logic, Requirement-Specification-Analysis, Formal Specification, Petri Net.

Relational logic: syntax and semantics; modeling using relations; introduction to the relational calculus; taxonomy and relational algebra.

Alloy: specification of invariants and operations using pre- and post-conditions using relational logic; idioms for modeling dynamic behaviour; semantics and type system; automatic verification techniques; comparison with other modeling languages, Specification of reactive systems: temporal logic (LTL and CTL); SPIN; explicit state model checking; symbolic model checking; tools for model checking.

Theorem proving: Introduction to the interactive construction of proofs. First order theories: employing SMT solvers.

Deductive verification: program logics; verification condition generation; behavioral interface specification languages and design by contract. Tools covered: Dafny; Frama-C; SPARK.

Model Checking: symbolic model checking; partial order reduction; bounded model checking. Tool covered: SMV. Software Model Checking: bounded model checking of software; existential abstraction mechanisms; predicate abstraction; abstraction refinement. Tools covered: CBMC; BLAST.

Fuzzing: Black-box, Grey-box, and White-box fuzzings. Smart Fuzzers. Tool: AFL
Symbolic Execution: Dynamic Symbolic Execution, Dynamic Symbolic Execution with pruning. Tools: KLEE, TracerX.

Learning Resources:

Text Books:

1. Daniel Jackson, "Software Abstractions: Logic, Language, and Analysis", Revised edition, MIT Press, 2012.
2. Christel Baier and Joost-Pieter Katoen, "Principles of Model Checking", MIT Press, 2008.

Reference Books:

1. Michael Huth and Mark Ryan, "Logic in Computer Science: Modelling and Reasoning about

Systems", Cambridge University Press, New York, NY, USA. 2004.
2. Edmund M. Clarke, Jr., Orna Grumberg, and Doron A. Peled, "Model Checking", MIT Press, Cambridge, MA, USA. 2000.

CS1440 **3-0-0 (3)**

Game Theory

Pre-Requisites: **None**

Course Outcomes:

CO-1	Analyze games based on complete and incomplete information about the players.
CO-2	Analyze games where players cooperate.
CO-3	Compute Nash equilibrium.
CO-4	Apply game theory to model network traffic.
CO-5	Analyze auctions using game theory.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	2	1	1	1	-	1	-	-	1	-	-	3	1	1	-
CO-2	2	3	2	2	3	-	-	1	-	1	-	1	2	2	1	-
CO-3	2	1	3	2	2	1	1	-	2	-	1	1	2	2	2	1
CO-4	2	2	2	1	2	1	1	-	1	1	-	-	2	1	2	1
CO-5	2	3	2	3	3	1	-	1	1	-	-	1	2	2	1	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Noncooperative Game Theory: Games in Normal Form - Preferences and utility, examples of normal-form, Analyzing games: Pareto optimality, Nash equilibrium, Maxmin and minmax strategies, dominated strategies, Rationalizability, Correlated equilibrium

Computing Solution Concepts of Normal-Form Games: Computing Nash equilibria of two-player, zero-sum games, Computing Nash equilibria of two-player, general-sum games, Complexity of computing Nash equilibrium, Lemke–Howson algorithm, Searching the space of supports, Computing Nash equilibria of n-player, general-sum games, Computing maxmin and minmax strategies for two- player, general-sum games, Computing correlated equilibria

Games with the Extensive Form: Perfect-information extensive-form games, Subgame-perfect equilibrium, Computing equilibria, Imperfect-information extensive-form games, Sequential equilibrium

Other Representations: Repeated games: Finitely repeated games, Infinitely repeated games, automata, Stochastic games Bayesian games: Computing equilibria.

CoalitionalGameTheory: Transferable Utility, Analyzing Coalitional Games, Shapley Value.

Core Mechanism Design: Strategic voting, unrestricted preferences, Implementation, quasilinear setting, efficient mechanisms, and Computational applications of mechanism design, Task scheduling, Bandwidth allocation in computer networks.

Auctions: Single-good auctions, Canonical auction families, Bayesian mechanisms, Multiunit auctions, combinatorial auctions,

Learning Resources:

Text Books:

1. Shoham, Y. and Leyton–Brown, K., "Multiagent Systems: Algorithmic, Game Theoretic, and Logical Foundations". Cambridge University Press, 2008.
2. Osborne, M. J., and Rubinstein, A., "A Course in Game Theory", Cambridge, MA: MIT Press, 1994.

Reference Books:

1. D. Fudenberg and J. Tirole, "Game Theory", The MIT Press, 2005.

CS1442 **3-0-0 (3)**

GPU Architecture and Computing

Pre-Requisites: None

Course Outcomes:

CO-1	Understand the architecture of GPU.
CO-2	Analyze for the performance of GPU memory hierarchy.
CO-3	Develop parallel programs using CUDA library.
CO-4	Develop parallel programs using OpenCL library.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	3	2	2	2	2	-	-	-	-	-	-	2	2	-	-
CO-2	3	2	2	2	2	2	-	-	-	-	-	-	2	2	-	3
CO-3	3	3	3	3	2	2	-	-	-	-	-	-	3	3	2	2
CO-4	3	3	3	3	3	3	-	-	-	-	-	-	3	3	-	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

GPU architecture: Review of Traditional Computer Architecture, Basic five stage RISC Pipeline, Cache Memory, Register File, SIMD instructions, Architecture of modern GPU, Streaming multiprocessor, cache hierarchy, Graphics pipeline, Data parallelism, device global memory and other types of memory.

Introduction to CUDA programming: CUDA C program structure, data transfer, kernel functions and threading, kernel launch, CUDA thread organization, mapping threads, resource assignment, quering device properties, thread scheduling, Thread warps and SIMD hardware, Matrix multiplication example, Floating point data representation, Linear solvers example, Prefix sums example, Histogram example, Merge sort example

Introduction to OpenCL: Data parallelism model, device architecture, kernel functions, device management and kernel launch, Examples

Introduction to OpenACC: Execution model, Basic constructs – Parallel, Loop, Kernels, Data management constructs, Asynchronous computation and data transfer, Examples.

Learning Resources:

Text Books:

1. David B. Kirk and Wen-mei W. Hwu, Programming massively parallel processors: A hands-on approach, Morgan Kaufmann, 2013, 2nd Ed.
2. David Kaeli, Perhaad Mistry, Dana Schaa, Dong Ping Zhang, Heterogeneous Computing with OpenCL 2.0, Morgan Kaufmann, 2015, 3rd Ed.

Reference Books:

1. Jason Sanders and Edward Kandrot, CUDA By Example – An Introduction to General-Purpose GPU Programming, Addison Wesley, 2011.

CS1444 **3-0-0 (3)**

Heterogeneous Computing

Pre-Requisites: **CS1208, CS1305**

Course Outcomes:

CO-1	Understand different models of parallel programming and the usage of MPI and OpenMP libraries.
CO-2	Analyze for the performance of GPU memory hierarchy.
CO-3	Develop parallel programs using OpenCL library.
CO-4	Generate parallel programs for matrix, graph and sorting problems using Cuda library.
CO-5	Develop mixed mode programs for Multicore and GPGPU systems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	3	2	2	2	2	–	–	–	–	–	–	2	2	-	-
CO-2	3	2	2	2	2	2	–	–	–	–	–	–	2	2	-	3
CO-3	3	3	3	3	2	2	–	–	–	–	–	–	3	3	2	2
CO-4	3	3	3	3	3	3	–	–	–	–	–	–	3	3	-	2
CO-5	2	3	3	3	3	2							3	3	-	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Programming using message passing paradigm: Principles, building blocks, MPI, Topologies and embedding, Overlapping communication and computation, collective communication operations, Groups and communicators

Programming shared address space platforms: Threads, POSIX threads, Synchronization primitives, attributes of threads, mutex and condition variables, Composite synchronization constructs, OpenMP Threading Building blocks; An Overview of Memory Allocators, An overview of Intel Threading building blocks.

GPU Computing - Introduction: Introduction to General Purpose Computing on Graphic Processing Units (GPGPU); GPU as parallel computers – CUDA enabled NVIDIA GPUs; AMD-ATI-OpenCL, GPGPU Architecture of a Modern GPU – Threaded Streaming Multi-processors; communication bandwidth; Unified Graphics and Computing Processors; GPGPU- GPU computing – Scalable GPUs; Speed-up & Parallelism; CPU/GPU programing; SPMD programming model

CUDA APIs & CUDA Threads: GPUs-Data Parallelism; GPU-CUDA Program Structure; GPU device memories & Data transfer; Kernel functions and threading; CUDA Runtime API; CUDA Thread Execution; CUDA Thread organization; Synchronization; Thread Scheduling;

OpenCL (Open Computing Language): Heterogeneous Computing – Programming; Data Parallelism Model – OpenCL; OpenCL, Device Architecture; OpenCL Kernel Functions; OpenCL APIs – MatrixMatrix, Computations using different partitioning techniques– OpenCL; OpenCL – Device Management and Kernel launch; Compilation Model and programming features of OpenCL – Device query; Object Query, and task parallelism model

Mixed Programming: Multi-Core Processors & GPUs : Heterogeneous computing – mixed programming (Message Passing-MPI and Shared Memory Programming (Pthreads, OpenMP); Heterogeneous computing - mixed programming – CPU (Pthreads, OpenMP) & GPU (CUDA, OpenCL); MPI-OpenCL & MPI-CUDA ; Programming for Dense Matrix Computations

Learning Resources:

Text Books:

1. Benedict R Gaster, Lee Howes, David R Kaeli Perhaad Mistry Dana Schaa, "Heterogeneous Computing with OpenCL", MGH, 2011.
2. Jason Sanders, Edward Kandrot, "CUDA By Example – An Introduction to General-Purpose GPU Programming", Addison Wesley, 2011.

Reference Books:

1. Michael J Quinn, "Parallel Programming in C with MPI and OpenMP", TMH, 2004.

CS1446 **3-0-0 (3)**

Human Computer Interaction

Pre-Requisites: None

Course Outcomes:

CO-1	Understand usability and the factors affecting universal usability.
CO-2	Apply the principles of design process in human computer interaction.
CO-3	Analyze and evaluate the interaction styles in human computer interaction.
CO-4	Analyze the design issues in human computer interaction.
CO-5	Create an interface based on the concepts of human computer interaction.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	1	1	–	2	–	–	–	–	–	–	–	–	–	–	–	–
CO-2	2	2	2	2	2	1	–	–	–	–	–	–	1	2	–	–
CO-3	2	2	2	2	2	–	–	–	–	–	–	–	2	–	–	–
CO-4	2	3	2	2	2	–	–	–	–	–	–	–	1	–	–	–
CO-5	2	1	3	1	2	–	–	–	–	–	–	–	2	2	–	–

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction: Usability of Interactive Systems, Universal Usability, Guidelines, Principles, and Theories

Design Processes: Design, Evaluation and the User Experience, Design Case Studies

Interaction Styles: Direct Manipulation and Immersive Environments, Fluid Navigation, Expressive Human and Command Languages, Devices

Design Issues: Advancing the User Experience, The Timely User Experience, Documentation and User Support, Information Search, Data Visualization.

Learning Resources:

Text Books:

1. Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, Nicholas Diakopoulos, Designing the User Interface: Strategies for Effective Human Computer Interaction, Pearson, 2021, 6th Edition.

Reference Books:

1. Wilbert O Galitz, The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, Wiley, 2007, 3rd Edition.

CS1448 **3-0-0 (3)**

IoT Security

Pre-Requisites: **CS1302**

Course Outcomes:

CO-1	Ability to Define and implement a security policy.
CO-2	Apply the knowledge to design and build the next generation of smart devices and networked systems.
CO-3	Identify various security issues to detect real-time and capture sensitive data.
CO-4	Understanding of the range of wireless and application protocols that support different IoT application domains.
CO-5	Design or build the reliable, secure system for IoT networks.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	–	1	–	–	2	–	–	–	–	–	–	–	–	–	3	–
CO-2	1	–	–	–	–	–	–	–	–	–	–	–	2	–	–	–
CO-3	–	–	–	1	–	–	2	–	2	–	–	–	–	–	–	–
CO-4	–	–	–	–	–	1	–	–	–	2	–	–	–	–	–	3
CO-5	–	–	1	–	–	–	–	–	–	–	–	–	3	–	–	–

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to IoT, IoT protocols, Security Requirements, IoT applications and their security/various vulnerabilities, Sensor-based attacks, IoT device-based attacks, Network attacks, attacks against IoT systems (hardware + software) Security issues in SCADA systems, ZIGBEE & BLE, Power analysis attacks, Invasive attacks, Perturbation -attacks, Electromagnetic side-channel attacks, fault injection attack, timing attack, covert channel attacks. M2M Security, RFID Security, Active vulnerability analysis tools, Port scanning, Operating system fingerprinting and version scanning, Penetration testing, Attack surface mapping, Hardware Security, PUF

Privacy Issues of IoT, Trust management in IoT, IoT device authentication, Lightweight authentication techniques in IoT, Blockchain For IoT, Challenges of IoT Addressing Security, Ethics, Privacy, and Laws, Machine Learning-Enabled IoT Security

Learning Resources:

Text Books:

1. Shancang Li, Li Da Xu, Securing the Internet of Things", Syngress, Elsevier, 2017.
2. Fei HU, "Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations", CRC Press,2016.

Reference Books:

1. Fei Hu, Security and Privacy in Internet of Things (IoTs) Models, Algorithms, and Implementations", CRC Press, 2016.

CS1450 **3-0-0 (3)**

Medical Image Processing

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand image processing techniques.
CO-2	Apply restoration and segmentation techniques in medical images.
CO-3	Understand the representations of features and classification methods.
CO-4	Apply deep learning methods for medical data analysis and generative models for generate synthetic data.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	1	1	2	2	–	1	–	–	1	–	–	2	1	2	2
CO-2	3	2	2	3	3	2	–	–	1	1	1	–	3	2	3	2
CO-3	2	1	1	2	2	–	1	–	–	1	–	–	3	2	2	1
CO-4	3	2	2	3	3	2	–	–	1	1	1	–	3	2	3	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to medical imaging, Image Sensing and Acquisition, Image Sampling and Quantization, Various modalities of Medical Imaging-CT, MRI, PET; Basic image processing techniques- medical image enhancement, image histogram equalization, image edge enhancement, noise reduction, filtering and image restoration; Image registration - Rigid and Non-Rigid models, Application and demonstration; Image representations, Classification methods, Clustering methods; Image segmentation - Histogram-based methods, Statistical shape model, PDE based methods, Multi-scale segmentation, semi-automated methods, clustering-based methods, classification-based methods; multi-model segmentation application and demonstration; Computer Aided Diagnosis – Case Study; Deep Learning for Medical image analysis – 3D Convolutional Neural Networks; Deep Learning for Medical image analysis – Generative models for synthetic data.

Learning Resources:

Text Books:

1. Digital Image Processing, Second edition Rafael C. Gonzalez, Richard E. Woods, (Prentice Hall)
2. Jerry L. Prince and Jonathan Links, "Medical Imaging Signals and Systems", First Edition, Prentice Hall, 2005.

Reference Books:

1. C Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.
2. Anil.K.J “Fundamentals of Digital Image Processing”, Prentice Hall of India Pvt. Ltd., New Delhi, 1995.
3. S Sridhar, Digital Image Processing Second Edition, Oxford University Press, 2016.

Othe Suggested Readings:

1. https://nptel.ac.in/courses/102106094

CS1452 **3-0-0 (3)**

Semantic Web

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the standards and data formats used in the Semantic Web
CO-2	Comprehend technologies including XML and XSLT
CO-3	Design semantic web meta data and RDF schema
CO-4	Develop ontology programming with Jena API

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	1	-	-	-	1	-	-	2	-	2	-	-	-	-	1	1
CO-2	-	1	2	-	2	1	2	-	-	-	-	2	-	1	2	1
CO-3	2	1	1	1	2	1	-	1	2	1	2	1	1	1	2	1
CO-4	-	-	1	1	1	2	1	-	1	-	-	1	2	1	1	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

The Semantic Web Vision, overview of techniques and standards, Semantic Web Architecture, XML with Document Type Definitions and Schemas, Transformation/Inference rules in XSLT, RuleML and RIF, metadata with RDF (Resource Description Framework); metadata taxonomies with RDF Schema; Ontology languages, Ontology Development using Protege editor, Ontology Querying, Ontology Reasoning and Description Logic (DL), Semantic Web Application Areas, Ontology programming with Jena API, Ontology Engineering.

Learning Resources:

Text Books:

1. Grigoris Antoniou and Frank van Harmelen, "A Semantic Web Primer", 1st Edition, MIT Press, 2004.
2. John Hebeler, Matthew Fisher, Ryan Blace and Andrew Perez-Lopez, "Semantic Web Programming", 1st Edition, Wiley, 2009.

CS1454 **3-0-0 (3)**

Social Networks

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the importance of social networks and social graphs.
CO-2	Enhance analytical skills for analyzing social networking data.
CO-3	Develop skills to leverage extended enterprise data.
CO-4	Create real-life case studies using social networks.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	-	-	1	-	1	-	-	1	-	-	1	-	1	-	-	1
CO-2	1	-	-	2	2	3	2	-	-	1	1	-	-	-	2	-
CO-3	1	2	2	-	1	-	2	-	-	2	-	2	-	2	-	2
CO-4	-	1	1	2	-	2	-	1	2	-	-	1	-	3	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to social network analysis: Graphs – nodes, edges, direct and indirect friends/neighbors, degree and degree distribution, shortest path, cycle, tree, complete graph, bipartite graphs, directed graphs, weighted graphs, adjacency matrix, social interactions and connected components. Technological networks (internet, telephone network, power grids, transportation networks), social networks (facebook, movie collaboration, paper collaboration), information networks (web), biological networks (neural networks, ecological networks).

Network Centrality Measures and Models: Properties of real-world network – degree distribution, clustering coefficient, average path length; Random Graphs – Evolution of random graphs, properties of random graphs, modeling real-world networks with random graphs, Erdos-Renyi model of random graph; Small-world Model – Properties of the Small-world model, modeling real-world networks with the small-world model; Preferential attachment model – Properties of the preferential attachment model, modeling real-world networks with the preferential attachment model.

Random walk-based proximity measures, other graph-based proximity measures. Clustering with random-walk based measures.

Influence and Homophily: Measuring Assortativity, Measuring and modeling Influence, Measuring and modeling Homophily, Distinguishing influence and homophily – shuffle test, edge-reversal test, randomization test; Spread of influence through a network, influence maximization in networks, spread of disease on networks.

Games on networks, game theory strategies, dominant strategies, dominated strategies, pure strategies and mixed strategies, Nash equilibrium, multiple equilibria-coordination games, multiple equilibria-the Hawk-Dove game, mixed strategies, Modeling social network traffic using game theory.

Learning Resources:

Text Books:

1. Reza Zafarani, Mohammad Ali Abbasi, Huan Liu, "Social Media Mining – An Introduction", Cambridge University Press, 2014.
2. David Easley and Jon Kleinberg, "Networks, Crowds, and Markets: Reasoning About a Highly Connected World", Cambridge University Press,2010.

Reference Books:

1. Mark Newman, "Networks: An Introduction", Oxford University Press,2010.
2. Hansen, Derek, Ben Sheiderman, Marc Smith, "Analyzing Social Media Networks with NodeXL: Insights from a Connected World", Morgan Kaufmann,2011.
3. Avinash Kaushik, "Web Analytics 2.0: The Art of Online Accountability", Sybex,2009.

CS1456 **3-0-0 (3)**

Software Defined Networks

Pre-Requisites: **CS1305**

Course Outcomes:

CO-1	Understand basic knowledge of the use of SDN and its implementation.
CO-2	Design and implement a network topology using P4 and analyze various packet features.
CO-3	Ability to carry out further research in SDN and P4.
CO-4	Understand the Network Function Virtualization and its importance in Advanced communication.
CO-5	Apply the knowledge of networking to design future network problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	–	1	–	–	–	–	–	–	–	–	–	2	–	–	–	–
CO-2	1	–	–	–	–	–	2	–	–	–	–	–	–	–	–	3
CO-3	–	–	–	–	1	–	–	–	–	–	–	–	2	–	–	–
CO-4	–	–	2	–	–	–	–	–	1	–	–	–	–	–	3	–
CO-5	–	–	–	–	–	2	–	–	–	–	–	–	–	–	–	–

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Network layer – Life of a Packet, Introduction and What's inside a router, Networks without challenges facing network managers, How SDN Works and SDN Architecture, SDN Controllers OpenFlow Overview-OVS, OpenFlow-Flow Tables and SDN use cases, Mininet, Existing SDN Controllers- Floodlight and Open Daylight, SDN Controller-Link discovery, Topology management, Flow manager, Decision making, Controller placement problem, Multi-controller issues, Load Balancing, Network measurement, network verification, network security, and network traffic management, Traffic engineering, Network Function Virtualization, Smart NIC, Data Plane-Motivation, Data Plane Implementation and Introduction to P4, Introduction to P4 & PISA architecture, P4 language, Software-Defined Networking for Internet of Things, Security issues on SDN architecture.

Learning Resources:

Text Books:

1. P. Goransson, C. Black, and T. Culver, Software Defined Networks: A Comprehensive Approach, Morgan Kaufmann, 2017, 2nd Edition.
2. K. Gray and T. D. Nadeau, Network Function Virtualization. Morgan Kaufmann, 2016.
3. Kreutz, D., Ramos, F. M. V., Veríssimo, P., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. , Software-defined networking: a comprehensive survey. CoRR., 2018.

Reference Books:

1. Qiang Duan Mehmet Toy, Virtualized Software-Defined Networks And Services, Artech House, 2016. First Edition.
2. Siamak Azodolmolky and Oswald, CokerSoftware-Defined Networking with OpenFlow, Packt Publishing, 2017, Second Edition.

CS1458 **3-0-0 (3)**

Virtual Reality and Augmented Reality

Pre-Requisites: **None**

Course Outcomes:

CO-1	Apply concepts of Virtual Reality.
CO-2	Apply concepts of Augmented Reality.
CO-3	Integrate sensors with AR/VR system.
CO-4	Design AR/VR application for a given task.
CO-5	Analyze existing AR/VR systems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	2	2	2	2	2	–	–	–	–	–	–	–	1	1	–	–
CO-2	2	2	2	2	2	–	–	–	–	–	–	–	1	1	–	–
CO-3	2	2	2	1	2	–	–	–	–	–	–	–	–	2	–	–
CO-4	3	2	2	3	3	–	–	–	–	1	–	–	2	2	2	2
CO-5	2	3	–	3	–	–	–	–	–	–	–	–	–	–	2	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Virtual Reality: Basics, History, Overview of Various Realities, Immersion.

Perception: Objective & Subjective Reality, Perceptual Modalities, Perception of Space & Time.

Content Creation: Environmental design, Affecting Behavior.

Interaction: VR Interaction Concepts, Input Devices, Interaction Patterns & Techniques.

Iterative Design of VR: Philosophy of Iterative Design, The Define stage, Make stage and Learn stage.

Software Development for Virtual Reality : Introduction to Augmented Reality: Basics, Displays, Tracking, Computer Vision for Augmented Reality, Calibration & Registration, Visualization, Interaction, Modeling & Annotation, Authoring, Navigation, Collaboration.

Software Development for Augmented Reality : Application and Future of VR/AR: Applications of VR/AR in Entertainment, Medical, Manufacturing, Education, etc.

Learning Resources:

Text Books:

1. Jason Jerald, The VR Book: Human-Centered Design for Virtual Reality, ACM and Morgan & Claypool Publishers, 2016, 1st Edition
2. Dieter Schmalstieg, Tobias Hollerer, Augmented Reality, Principles and Practice, Addison Wesley, 2016, 1st Edition
3. Ralf Doerner, Wolfgang Broll, Paul Grimm, Bernhard Jung, Virtual and Augmented Reality (VR/AR): Foundations and Methods of Extended Realities (XR), Springer, 2022

Reference Books:

1. Steven M. LaValle, Virtual Reality, Cambridge University Press, 2023
2. Jesse Glover and Jonathan Linowes, Complete Virtual Reality and Augmented Reality Development with Unity, Packt Publishers, 2019.

Open Electives

Code	Course Title	L-T-P	Credits	Sem.
CS1475	**Fundamentals of Data Structures**	2-0-0	2	VII
CS1477	**Object Oriented Programming Principles**	2-0-0	2	VII
CS1478	**Machine Learning Methods**	2-0-0	2	VIII
CS1480	**Internet of Things**	2-0-0	2	VIII

CS1475 **2-0-0 (2)**

Fundamentals of Data Structures

Pre-Requisites: **None**

Course Outcomes:

CO1	Understand the concept of ADT, identify data structures suitable to solve problems.
CO2	Develop and analyze algorithms for stacks, queues.
CO3	Develop algorithms for binary trees and graphs.
CO4	Implement symbol table using hashing techniques.
CO5	Implement Graph algorithms.

Syllabus:

Introduction to Iterative and Recursive Algorithms

Abstract Data Types (ADTs), Implementation and Applications of Stacks, Operations and Applications of Queues, Array Implementation of Circular Queues, Implementation of Stacks using Queues, Implementation Queues using Stacks, Linked Lists, Search and Update Operations on Varieties of Linked Lists, Linked List Implementation of Stacks and Queues

Introduction to Trees, Implementation of Trees, Binary Trees, Tree Traversals with an Application, Binary Search Trees (BSTs), Query and Update Operations on BSTs, AVL Trees, Rotations, Search and Update Operations on Balanced BSTs.

Hashing: Implementation of Dictionaries, Hash Function, Collisions in Hashing, Separate Chaining, Open Addressing, Analysis of Search Operations

Priority Queues: Priority Queue ADT, Binary Heap Implementation and Applications of Priority Queues, Disjoint Sets.

Graph Algorithms: Graphs and their Representations, Graph Traversal Techniques: Breadth First Search (BFS) and Depth First Search (DFS), Applications of BFS and DFS, Minimum Spanning Trees (MST), Prim's and Kruskal's algorithms for MST, Dijkstra's Algorithm for Single Source Shortest Paths.

Learning Resources:

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein,"Introductionto Algorithms", Second Edition, PHI, 2009.
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Third Edition, Pearson Education, 2006.

Reference Books:

1. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, 2011.
2. Michael T.Goodrich and Roberto Tamassia, "Algorithm Design: Foundations, Analysis and Internet Examples", Second Edition, Wiley-India, 2006.

CS1477 **2-0-0 (2)**

Object Oriented Programming Principles

Pre-Requisites: **None**

Course Outcomes:

CO1	Understand fundamental concepts in object-oriented approach.
CO2	Analyze design issues in developing OOP applications.
CO3	Write computer programs to solve real world problems in Java.
CO4	Analyze source code API documentations.
CO5	Create GUI based applications.

Syllabus:

Object- oriented thinking, overview of java, Object- oriented design, Structure of java program. Types and modifiers, Classes, declaring objects in classes, Methods, constructors, garbage collection, Method overloading, passing objects as parameters, Inheritance, various forms and types of inheritance, Multilevel hierarchy, use of super, method overriding, Applications of method overriding, abstract classes, Packages with examples

Interfaces and implementation, Exception handling, types, throwing, creating own exceptions, Multithreading and concepts, its usage and examples, Input/output streams, String operations and examples

Applets- methods, creation, designing and examples, Event handling- event classes, Event listener interfaces, AWT classes, working with frames, AWT controls-layoutmanager.

Learning Resources:

Text Books:

1. Timothy Budd, "Understanding Object-Oriented Programming with Java", Pearson, 2000.
2. Herbert Schildt, "The Complete Reference Java 2", TMH, 2017.

CS1478 **2-0-0 (2)**

Machine Learning Methods

Pre-Requisites: **None**

Course Outcomes:

CO1	Understand and Design Unsupervised Learning Methods.
CO2	Design Regression model to solve the classification and prediction problems.
CO3	Design Neural network to solve classification and function approximation problems.
CO4	Build optimal classifiers using genetic algorithms and Support vector machines.

Syllabus:

Well defined learning problems, Designing a Learning System; Introduction to Statistical Decision Theory - Regression, Linear Regression, Multivariate Regression and Logistic Regression; Unsupervised Learning Methods: Clustering, Partitional, Hierarchical and Density based clustering Methods; DECISION TREE LEARNING - Decision tree learning algorithm-Inductive bias- Issues in Decision tree learning; Bayesian Learning – Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier; ARTIFICIAL NEURAL NETWORKS – Perceptron, Gradient descent and the Delta rule, Adaline, Multilayer networks, Derivation of back-propagation rule-Back-propagation Algorithm, Convergence, Generalization; Support vector Machines; Instance-Based Learning – k-Nearest Neighbor Learning.

Learning Resources:

Text Books:

1. Tom.M.Mitchell, "Machine Learning", McGraw Hill International Edition, 1997.
2. C Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.

CS1480 **2-0-0 (2)**

Internet of Things

Pre-Requisites: **None**

Course Outcomes:

CO1	Analyze the protocol Stack for Internet of Things to address the heterogeneity in devices and networks.
CO2	To familiarize the IoT Hardware, Software and various Platforms of IoT.
CO3	Understanding the smart IoT Applications using smart sensor devices.
CO4	Usage of smart mobile apps for societal applications.

Syllabus:

Internet of Things-Concepts, Characteristics, Challenges, Industrial IoT, IoT Hardware, IoT Operating system, IoT Platforms IoT Architectures- Reference Model, Protocol Architecture, Perception Layer-Sensors, RFID, Network Layer-802.15.4, ZigBee, BLE and 6LoWPAN Application Layer- MQTT, CoAP, Mobile app developments for IoT, Introduction to Arduino Programming: Introduction to Raspberry Pi, Implementation of IoT with Raspberry Pi, IoT Smart Applications-Smart City, Health-Care, Smart Homes, Connected Vehicles, Smart Grid.

Learning Resources:

Text Books:

1. Olivier Hersent, “The Internet of Things Key Applications and Protocols”, Wiley, 2012.
2. Sudip Misra, “Introduction to IoT”, Cambridge University Press; First edition , 2021.
3. David Hanes, “IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things”, Cisco Press; 1st edition 2017.

Reference Books:

1. Arshdeeep Bahga, Vijay Madisetti, "Internet of Things: A Hands-on Approach", Universities Press, 2015.
2. Raj Kamal, "Internet of Things: Architecture and Design Principles", McGraw Hill Education private limited, 2017.
3. Kai Hwang, Min Chen, "Big Data Analytics for Cloud, IoT and Cognitive Computing", Wiley, 2018.

SYLLABI
Minor: Computer Science and Engineering

CS1M01 **2-0-2 (3)**

Data Structures

Pre-Requisites: **None**

Course Outcomes:

CO1	Understand the concept of ADT, identify data structures suitable to solve problems.
CO2	Develop and analyze algorithms for stacks, queues.
CO3	Develop algorithms for binary trees and graphs.
CO4	Implement sorting and searching algorithms.
CO5	Implement symbol table using hashing techniques.

Syllabus:

Introduction to Iterative and Recursive Algorithms

Abstract Data Types (ADTs), Implementation and Applications of Stacks, Operations and Applications of Queues, Array Implementation of Circular Queues, Implementation of Stacks using Queues, Implementation Queues using Stacks, Linked Lists, Search and Update Operations on Varieties of Linked Lists, Linked List Implementation of Stacks and Queues

Introduction to Trees, Implementation of Trees, Binary Trees, Tree Traversals with an Application, Binary Search Trees (BSTs), Query and Update Operations on BSTs, AVL Trees, Rotations, Search and Update Operations on Balanced BSTs, Splay Trees, B-trees, Trie, C-Trie

Hashing: Implementation of Dictionaries, Hash Function, Collisions in Hashing, Separate Chaining, Open Addressing, Analysis of Search Operations

Priority Queues: Priority Queue ADT, Binary Heap Implementation and Applications of Priority Queues, Disjoint Sets.

Sorting Algorithms: Stability and In Place Properties, Insertion Sort, Merge Sort, Quick Sort, Heap Sort, Lower Bound for Comparison Based Sorting Algorithms, Linear Sorting Algorithms: Counting Sort, Radix Sort, Bucket Sort

Graph Algorithms: Graphs and their Representations, Graph Traversal Techniques: Breadth First Search (BFS) and Depth First Search (DFS), Applications of BFS and DFS, Minimum Spanning Trees (MST), Prim's and Kruskal's algorithms for MST, Connected Components, Dijkstra's Algorithm for Single Source Shortest Paths, Biconnected Components.

Learning Resources:

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein,"Introductionto Algorithms", Second Edition, PHI, 2009.
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Third Edition, Pearson Education, 2006.

Reference Books:

3. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, 2011.
4. Michael T.Goodrich and Roberto Tamassia, "Algorithm Design: Foundations, Analysis and Internet Examples", Second Edition, Wiley-India, 2006.

CS1M02 **2-0-2 (3)**

Principles of Database Systems

Pre-Requisites: **CS1M01**

Course Outcomes:

CO-1	Understand functional components of the DBMS.
CO-2	Devise queries using Relational Algebra, Relational Calculus and SQL.
CO-3	Design database schema.
CO-4	Develop E-R model.
CO-5	Evaluate and optimize queries.
CO-6	Analyze transaction processing, concurrency control and recovery techniques.

Syllabus:

Introduction to DBMS: Historical perspective, File Versus a DBMS, Advantages of DBMS, Describing and storing data in DBMS, Architecture of a DBMS, Different Data Models;
Entity Relationship (ER) model: Features of ER model, conceptual design using ER model, design for large enterprises; Relational model–structure and operations, Integrity constraints over relations;
Query languages: Relational Algebra, Relational Calculus and SQL– Queries, Constraints, Form of SQL query, UNION, INTERSECT and EXCEPT, Nested queries, Aggregate Operators, Null values, Complex Integrity constraints in SQL, triggers and Embedded SQL;
Database Design: Mapping ER model to Relational form; Functional Dependency–Closer of functional dependencies, closer of attributes, canonical cover and Properties of Decompositions; Normalization process – 1NF, 2NF, 3NF and BCNF; Multivalued dependency– Closer properties of Multivalued dependency and 4NF; Join dependency– PJNF, Decomposition Algorithms;
Transaction Management: ACID properties, transactions, schedules and concurrent execution of transactions; Concurrency control – lock-based protocol, Serializability, recoverability, dealing with deadlocks and Concurrency control without locking;
Query Processing: Overview of Query Evaluation, operator evaluation; Algorithms for relational operations– Selection operation, General selection condition, Projection operation, Join operation, set operation and aggregate operation, Evaluation of relational operations; Query optimization: Alternative plans, functions of query optimizer, translating SQL queries into relational algebra, estimating the cost of a plan, relational algebra equivalences, and other approaches to query optimization;
Database Recovery: Failure classification, Recovery and atomicity, Log-based recovery shadow paging and Advanced Recovery Techniques:
Security and Authorization: Access control, direct access control and Mandatory access control, Role of DBA, Application development.

Learning Resources:

Text Books:

1. Silberschatz, Korth and Sudharshan, "Database System Concepts", McGraw Hill, 2010, 6th Edition.
2. Elamsri, Navathe, Somayajulu and Gupta, "Fundamentals of Database Systems", Pearson Education, 2011 6th Edition.

Reference Books:

1. Raghu Ramakrishnan, Johannes Gehrke, "Database Management Systems", McGraw Hill, 2003, 3nd Edition.
2. Authors, Title of the Text Book, Name of the Publisher, Year of publication, Edition

Other Suggested Readings:

1. MIT Open Course Ware: https://ocw.mit.edu/courses/6-830-database-systems-fall-2010/.

CS1M03 **3-0-0 (3)**

Operating System Principles

Pre-Requisites: **CS1M01**

Course Outcomes:

CO-1	Distinguish functional architectures of operating systems and file systems.
CO-2	Develop algorithms for subsystem components.
CO-3	Design device drivers and multi-threading libraries for an OS.
CO-4	Develop application programs using UNIX system calls.
CO-5	Design and solve synchronization problems.

Syllabus:

Introduction: Batch, iterative, time sharing, multiprocessor, distributed, cluster and real-time systems, UNIX system introduction and commands
Operating system structures: Computer system structure, Network structure, I/O Structure, Storage Structure, Dual mode operation, System components, Operating-System Services, System Calls, System Programs, System structure, Virtual Machines, System Design and Implementation, System Generation
Processes and Threads : Process Concept, Process Scheduling, Operations on Processes, Cooperating Processes, Interprocess Communication, Communication in Client – Server Systems, Multithreading Models, Threading Issues, Pthreads Basic Concepts,
CPU Scheduling: Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Real- Time Scheduling, Algorithm Evaluation, Process Scheduling Models
Process Synchronization: Synchronization Background, the Critical-Section Problem, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Critical Regions, Monitors, OS Synchronization
Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock
Memory Management: Memory Management Background, Swapping, Contiguous Memory Allocation, Paging, Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Process Creation, Page Replacement, Allocation of Frames, Thrashing, Operating-System Examples, Other Considerations
File System: File Concept, Access Methods, Directory Structure, File-System Mounting, File Sharing, Protection File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, Log-Structured File System, NFS
I/O Systems: Hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O to Hardware Operations, STREAMS, Performance, Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management, RAID Structure, Disk Attachment, Stable-Storage Implementation, Tertiary-Storage Structure

Learning Resources:

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Principles", Wiley, 10th Edition, 2019.
2. Richard Stevens, Stephen Rago, "Advanced Programming in the UNIX Environment", Pearson Education, 2nd Edition, 2013.

CS1M04 **3-0-0 (3)**

Computer and Communication Networks

Pre-Requisites: **CS1M03**

Course Outcomes:

CO1	Understand OSI and TCP/IP models.
CO2	Analyze MAC layer protocols and LAN technologies.
CO3	Design applications using internet protocols.
CO4	Implement routing and congestion control algorithms.
CO5	Develop application layer protocols.

Syllabus:

Introduction – network architecture - protocol implementation issues - network design. Reference models- The OSI Reference Model- the TCP/IP Model - A Comparison of the OSI and TCP/IP Models
Datalink Layer-Ethernet, Token ring, wireless LANs-Issues with data link Protocols-Encoding framing and error detection and correction-sliding window Protocol-Medium access control
Network layer – network layer design issues - Routing algorithms - Congestion control algorithms – Internetworking - The network layer in the internet - Internet Protocol (IP) - Unicast, multicast, and inter domain routing
Transport layer - Elements of transport protocol - Congestion control – The Internet's Transmission Control Protocol (TCP) - Remote Procedure Call (RPC) – Implementation semantics of RPC – BSD sockets - client-server applications
Application layer - Domain name server – Simple Mail Transfer Protocol – File Transfer Protocol - World wide web - Hypertext transfer protocol -Presentation formatting and data compression- Introduction to Network security - Web Services architectures for developing new application protocols.

Learning Resources:

Text Books:

1. Larry L Peterson, Bruce S Davis, "Computer Networks", 5th Edition, Elsevier, 2012.
2. Andrew S. Tanenbaum, David J Wetherall, "Computer Networks", 5th Edition, Pearson Edu, 2010.

CS1M05 **3-0-0 (3)**

Software Engineering Principles

Pre-Requisites: None

Course Outcomes:

CO1	Comprehend software development life cycle.
CO2	Prepare SRS document for a project.
CO3	Apply software design and development techniques.
CO4	Identify verification and validation methods in a software engineering project.
CO5	Implement testing methods for software.
CO6	Analyze and Apply project management techniques for a case study.

Syllabus:

The Software Problem - Cost, Schedule, and Quality, Scale and Change ; Software Processes- Processand Project , Component Software Processes; Software Development Process Models - Waterfall Model, Prototyping , Iterative Development , Rational Unified Process , Time boxing Model , ExtremeProgramming and Agile Processes , Using Process Models in a Project. Software Requirements Analysis and Specification - Value of a Good SRS , Requirement Process , Requirements Specification;Formal Specification- Formal Specification in the Software process, Sub-system interface specification,Behavioural Specification; Desirable Characteristics of an SRS - Components of an SRS, Structure of a Requirements Document; Functional Specification with Use Cases - Basics , Examples , Extensions,Developing Use Cases; Other Approaches for Analysis - Data Flow Diagrams , ER Diagrams , Validation; Software Architecture - Role of Software Architecture, Architecture Views - Component and Connector View - Components, Connectors, An Example. Architecture Styles for C&C View - Pipe and Filter, Shared-Data Style , Client-Server Style, Some Other Styles, Documenting ArchitectureDesign - Evaluating Architectures; Design - Design Concepts - Coupling , Cohesion , The Open-ClosedPrinciple . Function-Oriented Design (from Pressman) - Structure Charts, Structured Design Methodology, An Example. Object-Oriented Design (from Jalote)- OO Concepts, Unified Modeling Language (UML) , A Design Methodology , Examples; Detailed Design - Logic/Algorithm Design, State Modeling of Classes; Verification - Metrics - Complexity Metrics for Function-Oriented Design,Complexity Metrics for OO Design; Coding and Unit Testing -Programming Principles and Guidelines
- Structured Programming , Information Hiding, Some Programming Practices, Coding Standards; Incrementally Developing Code - An Incremental Coding Process ,Test-Driven Development, Pair Programming; Managing Evolving Code - Source Code Control and Build, Refactoring; Unit Testing -Testing Procedural Units, Unit Testing of Classes; Code Inspection - Planning, Self-Review, Group Review Meeting; Metrics - Size Measures, Complexity Metrics; Testing - Testing Concepts - Error, Fault, and Failure, Test Case, Test Suite, and Test Harness , Psychology of Testing , Levels of Testing;Testing Process - Test Plan, Test Case Design, Test Case Execution; Black-Box Testing - EquivalenceClass Partitioning, Boundary Value Analysis , Pairwise Testing, Special Cases, State-Based Testing; White-Box Testing - Control Flow-Based Criteria, Test Case Generation and Tool Support; Metrics - Coverage Analysis, Reliability, Defect Removal Efficiency.

Learning Resources:

Text Books:

1. Pankaj Jalote, "Software Engineering Precise Approach" , Wiley Publishers, 2012.
2. Ian Sommerville, "Software Engineering", 8/e Pearson Publishers, 2012.
3. Roger Pressman, "Software Engineering", 5th edition, MCgrawHill, 2002.

CS1M06 **3-0-0 (3)**

Pattern Recognition and Machine Learning

Pre-Requisites: **None**

Course Outcomes:

CO1	Understand and Design Unsupervised Learning Methods.
CO2	Design Regression model to solve the classification and prediction problems.
CO3	Design Neural network to solve classification and function approximation problems.
CO4	Build optimal classifiers using genetic algorithms and Support vector machines.

Syllabus:

Well defined learning problems, Designing a Learning System, Dimensionality Reduction, PCA, LDA; Introduction to Statistical Decision Theory - Regression, Linear Regression, Multivariate Regression and Logistic Regression; Unsupervised Learning Methods: Clustering, Partitional, Hierarchical and Density based clustering Methods; DECISION TREE LEARNING - Decision tree learning algorithm-Inductive bias-Issues in Decision tree learning; Bootstrapping & Cross Validation, Class Evaluation Measures, ROC curve, MDL, Ensemble Methods - Bagging, Committee Machines and Stacking, Boosting; Bayesian Learning – Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier, Bayesian belief networks, EM algorithm; ARTIFICIAL NEURAL NETWORKS – Perceptron, Gradient descent and the Delta rule, Adaline, Multilayer networks, Derivation of back-propagation rule-Back-propagation Algorithm, Convergence, Generalization; Support vector Machines; Instance-Based Learning – k-Nearest Neighbor Learning, Locally Weighted Regression.

Learning Resources:

Text Books:

1. Tom.M.Mitchell, "Machine Learning", McGraw Hill International Edition, 1997.
2. C Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.

SYLLABI

Honors: Computer Science and Engineering

CS1H01 **3-0-0 (3)**

Advanced Compiler Design

Pre-Requisites: **None**

Course Outcomes:

CO1	Understand code generation methods.
CO2	Apply scalar variable optimizations and procedural optimizations on intermediate code.
CO3	Apply machine level optimizations on the low-level intermediate code.
CO4	Perform loop restructuring transformations.

Syllabus:

Introduction: Interpreters - Recursive and iterative interpreters
Code generation: Arithmetic statement translation, acyclic graph representation, pattern matching in the acyclic graph, graph rewriting, linearization following the dependencies, code generation for purely register machine, purely stack machine and with memory addressing. Code generation for a basic block.
Simple optimizations: Constant folding, Scalar replacement of aggregates and Algebraic simplifications, Value numbering, Loop invariants identification, loop invariant code motion, partial redundancy elimination, procedure optimizations - call and return optimizations, Code hoisting, Induction variable identification and optimizations, Unnecessary bounds checking elimination Register Allocation and Code scheduling : Local methods and global methods, representation of the register allocation problem as graph colouring problem, heuristics, register tracking and spilling, Pipeline and Code scheduling effect on performance of the program, Software pipelining, speculative scheduling, boosting
Inter-procedural data flow analysis - static and dynamic, optimizations, Optimizing for memory hierarchies
High performance systems – Scalar, vector, multiprocessor, SIMD, Message Passing Architectures. Sequential and parallel loops. Data dependence, Use-Def chains. Dependence system, GCD test, Banerjee's Inequality, Exact algorithm, exact algorithm, Vectorization, Concurrentization, Array region analysis, Loop restructuring transformations.

Learning Resources:
Text Books:

1. Steven S. Muchnick, "Advanced Compiler Design & Implementation", Morgan Kaufmann, Elsevier Science, 2003.
2. Michael Wolfe, "High Performance Compilers for Parallel Computing", Addison Wesley, 1995.

CS1H02 **3-0-0 (3)**

Advanced Software Engineering

Pre-Requisites: **None**

Course Outcomes:

CO1	Apply the Object-Oriented Software-Development Process to design software.
CO2	Design large-scale, reusable and complex software systems with Design and Architectural patterns.
CO3	Develop and apply testing strategies for software applications.
CO4	Analyze different Software Reliability parameters using Markovian Models, Finite Failure Category Models and Infinite Failure category Models.
CO5	Design and Plan software solutions to security problems using various paradigms.

Syllabus:

Introduction and System Engineering: Introduction, Software Process and Methodology, System Engineering. Analysis and Architectural Design: Software Requirement Elicitation, Domain Modeling, Architectural Design. Modeling and Design of Interactive Systems and Other Types of Systems: Deriving Use Cases from Requirements, Actor-System Interaction Modeling, Object Interaction Modeling, Applying Responsibility-Assignment Patterns, Deriving a design class diagram, User Interface Design. Object State modeling of Event-Driven Systems, Activity Modeling for Transformational Systems. Implementation and Quality Assurance: Implementation Considerations, Software Quality Assurance, Software Testing.
Software Reliability Modeling: Markovian Models, Finite Failure Category Models, Infinite Failure Category Models. Comparison of Software Reliability Models.
Project Management and Software Security: Software Project Management, Software Security.

Learning Resources:

Text Books:

1. Kung, David. Object-oriented software engineering: an agile unified methodology. McGraw-Hill Higher Education, 2013.
2. Gamma, Erich. Design patterns: elements of reusable object-oriented software, Pearson Education India, 1995.

Reference Books:

1. M. Xie, Software Reliability Modelling, World Scientific; 1991.
2. John D. Musa, Anthony Iannino, Kazuhira Okumoto, Software Reliability Measurement, Prediction, Application.McGraw-Hill Book Company; 1987.

CS1H04 **3-0-0 (3)**

Advanced Computer Networks

Pre-Requisites: **None**

Course Outcomes:

CO1	Differentiate between traditional networks and software defined networks and understand the key benefits and use cases of SDN.
CO2	Interpret the SDN data plane devices and OpenFlow Protocols.
CO3	Implement the operation of SDN control plane with different controllers.
CO4	Apply techniques that enable applications to control the underlying network using SDN.
CO5	Evaluate Network Functions Virtualization components and their roles in SDN.

Syllabus:

Networking Basics: Switching, Addressing, Routing
SDN Background and Motivation: Evolving network requirements-The SDN Approach: Requirements, SDN Architecture, and Characteristics of Software-Defined Networking.
SDN Data plane and OpenFlow: Data plane Functions, Data plane protocols, OpenFlow: Switch-Controller Interaction, Flow Table, Packet Matching, Actions and Packet Forwarding Flow Table Structure, Flow Table Pipeline, The Use of Multiple Tables, Group Table, Extensions and Limitations, Data plane scalability.
SDN Control Plane: SDN Control Plane Architecture: Control Plane Functions, Southbound Interface, Northbound Interface, Routing, Cooperation and Coordination among Controllers, Controller placement problem, SDN controllers: OpenDaylight, Ryu, ONOS, Floodlight, Control plane scalability, fault tolerance.
SDN Application Plane: SDN Application Plane Architecture: Northbound Interface, Network Applications, User Interface- Network Services Abstraction Layer: Abstractions in SDN, Frenetic- Traffic Engineering Measurement and Monitoring, Security, network updates, SDN usecases: Traffic engineering, network management, network virtualization.
Network Functions Virtualization: Background and Motivation for NFV- NFV Principles, High-Level NFV Framework, NFV Benefits and Requirements- SDN vs. NFV, Network Functions, Service Creation and Chaining, NFV Orchestration, VNF deployment, Service function Chain Deployment. NFV Reference Architecture: NFV Management and Orchestration.
Emerging SDN Models: Protocol Models: NETCONF, BGP, MPLS, Controller Models, Application Models: Proactive, Declarative, External, SDN in Datacenters: Multitenancy, Failure Recovery, SDN in Internet eXchange Points (IXPs).

Learning Resources:

Text Books:

1. Paul Goransson Chuck Black Timothy Culver: Software Defined Networks: A Comprehensive Approach, Morgan Kaufmann, 2016.
2. Ken Gray Thomas Nadeau: Network Function Virtualization, Morgan Kaufmann, 2016.
3. Larry Peterson , Carmelo Cascone , Bruce Davie: Software-Defined Networks: A Systems Approach, Systems Approach, 2021.

CS1H05 **3-0-0 (3)**

Digital Video Processing

Pre-Requisites: **None**

Course Outcomes:

CO1	Use the suitable image/video acquisition approach, representation and quality assessment for a given scenario.
CO2	Apply image/video processing operations for enhancement and restoration.
CO3	Understand image/video compression standards.
CO4	Solve a video indexing, summarization, browsing and retrieval problem.
CO5	Implement an approach to solve a video analytics problem.

Syllabus:

Introduction – Introduction to digital image and video processing.
Image/Video processing – Intensity transformations, Applications of linear filtering and non-linear filtering. Morphological operations. Colour image processing. Features for Image and Video processing. Geometric transforms and Image registrations, Image segmentation and analysis. Spatiotemporal noise filtering, coding artifact reduction, Blotch detection and removal, video stabilization, and recent image/video operations and features in contemporary literature.
Image Compression – lossless coding, block transform coding, fundamentals of vector quantization, wavelet image compression, JPEG image compression.
Video Compression – basic concepts and techniques of video coding and the H.261 Standard, spatiotemporal subband/Wavelet Video Compression, Object-based video coding, introduction to video coding standards and formats: H.261, MPEG-2/DVB, MPEG-4, H.264/AVC, H.265/HEVC, H.266/VVC.
Image and Video Acquisition – Image scanning, sampling and interpolation, video sampling and interpolation. Recent standards and practices in image and video acquisition.
Video Quality Assessment – Introduction, HVS Modelling based methods, feature based methods, motion modelling-based methods, and approaches based on contemporary research.
Indexing, Summarization, Browsing and Retrieval – Introduction, Image and Video features, Video analysis, video representation, video browsing and video retrieval. Video features in contemporary research.

Video Analytics – Review of video analytics algorithms for: motion and change detection, object detection, object tracking, behaviour analysis, face recognition, Image and Video classification and recent visual analytics approaches in contemporary research.

Learning Resources:

Text Books:

1. Alan C Bovik, Handbook of Image and Video Processing, 2nd Edition, Academic Press, 2005.
2. Alan C. Bovik, The Essential Guide to Video Processing, 1st Edition, Academic Press, 2009.
3. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, 4th Edition, Pearson, 2018.

Other Suggested Readings:

1. Recent articles in Research and Industry.

CS1H07 **3-0-0 (3)**

Data Privacy

Pre-Requisites: **None**

Course Outcomes:

CO1	Define differential privacy.
CO2	Design techniques to achieve differential privacy for linear queries.
CO3	Design mechanisms for query release problem using online learning algorithms.
CO4	Analyze computational complexity of differentially private mechanisms.

Syllabus:

The Promise of Differential Privacy: Privacy-preserving data analysis; Basic Terms: The model of computation, Towards defining private data analysis, Formalizing differential privacy; Basic Techniques and Composition Theorems: Useful probabilistic tools, Randomized response, The laplace mechanism, The exponential mechanism, Composition theorems, The sparse vector technique; Releasing Linear Queries with Correlated Error: An offline algorithm: SmallDB, An online mechanism: private multiplicative weights; Generalizations: Mechanisms via α-nets, The iterative construction mechanism, Connections; Boosting for Queries: The boosting for queries algorithm, Base synopsis generators; When Worst-Case Sensitivity is Atypical: Subsample and aggregate, Propose- test-Release, Stability and privacy; Lower Bounds and Separation Results: Reconstruction attacks, Lower bounds for differential privacy; Differential Privacy and Computational Complexity: Polynomial time curators, Some hard-to-Syntheticize distributions, Polynomial time adversaries; Differential Privacy and Mechanism Design: Differential privacy as a solution concept, Differential privacy as a tool in mechanism design, Mechanism design for privacy aware agents; Differential Privacy and Machine Learning: The sample complexity of differentially private machine learning, Differentially private online learning, Empirical risk minimization; Additional Models: The local model, Pan-private streaming model, Continual observation, Average case error for query release.

Learning Resources:

Text Books:

1. C. Dwork and A. Roth, The Algorithmic Foundations of Differential Privacy, now Publishers, 2014.
2. Charu C. Aggarwal, Privacy-Preserving Data Mining: Models and Algorithms, 1st Edition, Springer, 2008.

Other Suggested Readings:

1. Relevant Research Papers

CS1H09 **3-0-0 (3)**

Responsible and Explainable AI

Pre-requisites: **CS1304**

Course Outcomes:

CO1	Understand the fairness of the AI model and explore bias reduction strategy.
CO2	Enhance analytical skills in identifying data factors that affect AI algorithms' performance.
CO3	Develop skills to leverage trust and fairness in building AI systems.
CO4	Create real-life case studies for Responsible AI frameworks in different scenarios.

Syllabus:

Introduction to Explainable and Responsible AI, Robustness, Need for Ethics in AI. AI for Society and Humanity, Stages of AI model development and how XRAI is relevant to these stages, Responsible AI Frameworks; Bias and Fairness of AI Model - Fairness and Bias - Sources of Biases, Exploratory data analysis, limitation of a dataset, Preprocessing, in-processing and postprocessing to remove bias, Group fairness and Individual fairness, Counterfactual fairness; Explainable AI- Interpretability through simplification and visualization, Intrinsic interpretable methods, Post Hoc interpretability, Explainability through causality, Model agnostic Interpretation; Ethical Considerations in AI, Ethics, and Accountability - Auditing AI models, fairness assessment, Principles for ethical practices; Vulnerability of AI Model, Privacy preservation - Attack models, Privacy-preserving Learning, Differential privacy, Federated learning; Case studies - Recommendation systems, Computer Vision, Natural Language Processing, etc.; Responsible Generative AI and Large Language Models.

Learning Resources:

Text Books:

1. Virginia Dignum, “Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way” Springer Nature, 2019.
2. Christoph Molnar “Interpretable Machine Learning”.Lulu, 1st Edition, 2019.

CS1H10 **3-0-0 (3)**

Algorithmic Coding Theory

Pre-Requisites: **None**

Course Outcomes:

CO1	Understand Shannon's noisy coding theorem, Shannon capacity and entropy.
CO2	Design of error correcting codes and decoding algorithms.
CO3	Design and Analysis of light weight and code-based cryptosystems.
CO4	Design of network coding algorithms for communication networks.

Syllabus:

Shannon Theorem, Shannon capacity, Hamming's Theory, Error correcting codes, Linear codes, Impossibility results for codes, Mac Williams Identities, Linear programming bound, The asymptotic perspective, Encoding, Decoding from erasures, Decoding RS codes, List decoding, linear time decoding, LDPC codes, Sipser-Spielman codes, Linear time encoding and decoding, Linear time and near optimal error decoding, Expander based constructions of efficiently, decodable codes, Some NP hard coding theoretic problems, Applications in complexity theory, Cryptography with error correcting codes, Lossless Multicast Network Coding, Network coding in Lossy Networks, Security against adversarial errors, Error correction bounds for centralized network coding.

Learning Resources:

Text Books:

1. Tom Richardson, RudigerUrbanke, Modern Coding Theory, Cambridge University Press, 2008.
2. John b. Anderson and Seshadri Mohan, Source and Channel Coding: An Algorithm Approach, Springer, 1991.
3. G. Kabatiansky, E. Krouk and S. Semenov, Error Correcting Coding and Security for Data Networks, John Wiley & Sons Ltd., 2005.
4. Jiri Adamek, Foundations of Coding, Wiley Interscience Publication, John Wiley & Sons, 1991.

CS1H12 **3-0-0 (3)**

Soft Computing Techniques

Pre-Requisites: **None**

Course Outcomes:

CO1	Understanding of optimizations problems, comprehend the fuzzy logic and the concept of fuzziness involved in various systems and fuzzy set theory.
CO2	Understand the fundamental theory and concepts of neural networks and Identify different neural network architectures, algorithms, applications and their limitations.
CO3	Apply genetic algorithms and neural networks to solve real world problems.
CO4	Apply soft computing techniques to solve engineering and other societal problems.

Syllabus:

Overview of course and Basic of Soft Computing, Introduction of Neural Networks, Learning Process and Learning Task, Supervised Learning – Single and Multi – Layer Network, Associative Memory, Self-organizing Maps, Neuro-Dynamics, Hopfield Network, Fuzzy Logic and Systems-Fuzzy Sets and Membership Functions, Operations on Fuzzy Sets, Fuzzification. Fuzzy Numbers- Uncertain Fuzzy Values, Fuzzy Numbers and its L-R representation, Operations on Fuzzy Numbers. Fuzzy Relations, Fuzzy Inference Systems- Architecture of Fuzzy Inference System, Fuzzy Inference Rules and Reasoning, Defuzzification. Applications of Fuzzy Logic, Genetic algorithms and evolutionary computation. Applications of Genetic Algorithms & Hybrid Systems.

Learning Resources:

Text Books:

1. R.A. Aliev, R.R. Aliev, Soft Computing and Its Applications, World Scientific Publications, 2001.
2. Roger Jang, Tsai Sun, Eiji Mizutani, Neuro-Fuzzy and Soft Computing: A computational Approach to Learning & Machine Intelligence, PHI, 2008.
3. Simon Haykin, Neural Network: A Comprehensive Foundation, PHI, 1999.
4. Kishan Mehtrotra, S. Ranka, Elements of artificial Neural Networks, Penram International Publishing (India), 2009
5. Timothy Ross, Fuzzy Logic with Engineering Applications, 3rd Edition, McGraw-Hill, 2010.
6. Bart Kosko, Neural Networks and Fuzzy Systems, PHI, 1994.

CS1H14 **3-0-0 (3)**

Reinforcement Learning

Pre-Requisites: **None**

Course Outcomes:

CO1	Formulate Reinforcement Learning problems.
CO2	Apply various Tabular Solution Methods to Markov Reward Process Problems.
CO3	Apply various Iterative Solution methods to Markov Decision Process Problems.
CO4	Comprehend Function approximation methods.

Syllabus:

Introduction: Introduction to Reinforcement Learning (RL) – Difference between RL and Supervised Learning, RL and Unsupervised Learning. Elements of RL, Markov property, Markov chains, Markov reward process (MRP).
Evaluative Feedback - Multi-Arm Bandit Problem: An n-Armed Bandit Problem, Exploration vs Exploitation principles, Action value methods, Incremental Implementation, tracking a non-stationary problem, optimistic initial values, upper-confidence-bound action selection, Gradient Bandits.
Introduction to and proof of Bellman equations for MRPs
Introduction to Markov decision process (MDP), state and action value functions, Bellman expectation equations, optimality of value functions and policies, Bellman optimality equations.

Dynamic Programming (DP): Overview of dynamic programming for MDP, principle of optimality, Policy Evaluation, Policy Improvement, policy iteration, value iteration, asynchronous DP, Generalized Policy Iteration.
Monte Carlo Methods for Prediction and Control: Overview of Monte Carlo methods for model free RL, Monte Carlo Prediction, Monte Carlo estimation of action values, Monto Carlo Control, On policy and off policy learning, Importance sampling.

Temporal Difference Methods: TD Prediction, Optimality of TD(0), TD Control methods - SARSA, Q-Learning and their variants.

Eligibility traces: n-Step TD Prediction, Forward and Backward view of TD(λ), Equivalence of forward and backward view, Sarsa(λ),, Watkins's Q(λ), Off policy eligibility traces using importance of sampling.

Function Approximation Methods: Value prediction with function approximation, gradient descent methods, Linear methods, control with function approximation.

Learning Resources:

Text Books:

1. Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction, 2nd Edition, The MIT Press, 2018.
2. Csaba Szepesvari, Algorithms for Reinforcement Learning, Morgan & Claypool, 2010