COURSE SCHEME AND SYLLABUS

FOR

B.E. (AI & DS)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

2026

SEMESTER-I

S. No.	Course Code	Course Name	CODE**	L	T	P	Cr
1.	UCB009	Chemistry	BSC	3	0	2	**4**
2.	UES103	Programming for Problem Solving (C language)	ESC	3	0	2	**4**
3.	UES013	Electrical & Electronics Engineering	ESC	3	1	2	**4.5**
4.	UEN008	Energy and Environment	OTH	2	0	0	**2**
5.	UMA022	Calculus for Engineers	BSC	3	1	0	**3.5**
6.	UAI101	Foundation of Machine Intelligence: Concepts, Techniques and Applications		2	0	0	**2**
		TOTAL					**20**

SEMESTER-II

S. No.	Course Code	Course Name	CODE**	L	T	P	Cr
1.	UPH013	Physics	BSC	3	1	2	**4.5**
2.	UES101	Engineering Drawing	ESC	2	4	0	**4**
3.	UHU003	Professional Communication	HSS	2	0	2	**3**
4.	UES102	Manufacturing Processes	ESC	2	0	2	**3**
5.	UMA023	Differential Equations and Linear Algebra	BSC	3	1	0	**3.5**
6.	UAI201	Programming Essentials for AI and ML		2	0	2	**3**
		TOTAL					**21**

SEMESTER-III

S. No.	Course Code	Course Name	CODE**	L	T	P	Cr
1.	UCS303	Operating System	PCC	3	0	2	**4**
2.	UAI301	Object Oriented Programming	PCC	3	0	2	**4**
3.	UCS301	Data Structures	PCC	3	0	2	**4**
4.	UMA021	Numerical Linear Algebra	BSC	3	0	2	**4**
5.	UTA016	Engineering Design Project I (2 self-effort hours)	ESC	1	0	2	**3**
6.	UAI302	Experiential Learning – Foundations and Practice	BSC			2	**1**
7.	UHU050	Evolutionary Psychology	HSS	1*	0	0	**1**
		TOTAL					21

Note: *Alternate week

SEMESTER-IV

S. No.	Course Code	Course Name	CODE**	L	T	P	Cr
1.	UCS415	Design and Analysis of Algorithms	PCC	3	0	2	**4**
2.	UCS310	Database Management Systems	PCC	3		2	**4**
3.	UMA401	Probability and Statistics	BSC	3	0	2	**4**
4.	UTA035	Engineering Design Project II (Edge Computing Design Project)	PCC	1	0	4	**3**
5.	UTD002	Employability Development Skills	HSS	2	0	0	**2**
6.	UCS405	Discrete Mathematical Structures	PCC	3	1	0	**3.5**
7.	UAI401	Experiential Learning – Advanced Applications and Projects				2	**1**
		TOTAL					21.5

SEMESTER-V

S. No.	Course Code	Course Name	CODE**	L	T	P	Cr
1	UAI504	Computer Networks	CP	2	0	2	**3**
2	UAI501	Image Processing	CP	3	0	2	**4**
3	UCS503	Software Engineering	CP	3	0	2	**4**
4	UAI503	Computer Architecture and Organization	CP	3	0	0	**3**
5	UAI502	Advanced Deep Learning		3		2	**4**
6	UAI402	Foundation of Data Engineering	PCC	3	0	2	**4**
7		ELECTIVE-I	PEC	2	0	2	3.0
		TOTAL					25

Note: *Alternate week

SEMESTER-VI

S. No.	Course Code	Course Name	CODE**	L	T	P	Cr
1	UCS701	Theory of Computation	CP	3	1	0	**3.5**
2	UMA071	Numerical Optimization	CF	3	0	2	**4**
3	UAI601	Accelerated Computing		3		2	**4**
4	UTA025	Innovation and Entrepreneurship	PR	1	0	2*	**3**
5	Generic Elective		GE	2	0	0	**2**
6	Elective-II		PE	2	0	2	**3**
7	Elective-III		PE	2	0	2	**3**
8	UCS797	Capstone Project*	PR	0	0	2	**-**
		TOTAL					22.5

Elective Focus

B.E. Data Science and Artificial Intelligence Program is designed to offer elective focus as soon as student clears semester IV of the program. Student has to choose EF (Elective Focus) out of the following ten choices and shall continue with this group till his study at Thapar Institute of Engineering & Technology. Choices are:

1. High Performan Computing

1.1. Cloud Computing (UCS531)
1.2. GPU Computing (UCS635)
1.3. Parallel & Distributed Computing (UCS645)
1.4. Simulation & Modelling (UCS751)

2. Computer Animation and Gaming

2.1. Computer Vision (UCS532)
2.2. 3D Modelling and Animation (UCS636)
2.3. Game Design & Development (UCS646)
2.4. Augmented and Virtual Reality (UCS752)

3. Information and Cyber Security

3.1. Computer & Network Security (UCS534)
3.2. Secure Coding (UCS638)
3.3. Cyber Forensics (UCS648)
3.4. Blockchain Technology and Applications (UCS754)

4. Mathematics and Computing

4.1. Linear Algebra for Artificial Intelligence and Machine Learning (UMC513)
4.2. Financial Mathematics (UMC632)
4.3. Mathematics for Quantum Computing (UMC633)
4.4. Cryptography and Coding Theory (UMC744)

5. Financial Derivative (Future First Collaboration)

5.1. Finance, Accounting and Valuation (UCS539)
5.2. Financial Markets and Portfolio Theory (UCS675)
5.3. Derivatives Pricing, Trading and Strategies (UCS658)
5.4. Quantitative and Statistical Methods for Finance (UMC743)

6. DevOps and Continuous Delivery (Xebia Collaboration)

6.1. Source Code Management (UCS537)
6.2. Build and Release Management (UCS659)

6.3. Continuous Integration and Continuous Deployment (UCS660)
6.4. System Provisioning and Configuration Management (UCS758)

7. Full Stack (Xebia Collaboration)

7.1. UI & UX Specialist (UCS542)
7.2. Data Engineering (UCS677)
7.3. Test Automation (UCS662)
7.4. Cloud & DevOps (UCS745)

8. Conversational AI (NVIDIA Collaboration)

8.1. Conversational AI: Accelerated Data Science (UCS551)
8.2. Conversational AI: Natural Language Processing (UCS664)
8.3. Conversational AI: Speech Processing & Synthesis (UCS749)
8.4. Generative AI (UCS748)

9. Robotics and Edge AI (NVIDIA Collaboration)

9.1. Edge AI and Robotics: Data Centre Vision (UCS668)
9.2. Edge AI and Robotics: Accelerated Data Science (UCS547)
9.3. Edge AI and Robotics: Embedded Vision (UCS671)
9.4. Edge AI and Robotics: Reinforcement Learning & Conversational AI (UCS760)

10. Cyber Forensics and Ethical Hacking (EC-Council, USA)

10.1. Network Defence (UCS550)
10.2. Ethical Hacking-1 (UCS673)
10.3. Ethical Hacking-2 (UCS674)
10.4. Computer Hacking and Forensic Investigation (UCS750)

11. Intelligent Transport Systems

11.1. Network and Communication for Connected Vehicles (UEC646)
11.2. Intelligent Transportation Systems(UCS678)
11.3. Data Analytics in Automobile Engineering(UCS679)
11.4. Cyber security for Mobility Systems(UCS762)

LIST OF PROFESSIONAL ELECTIVES

Elective I

S.No.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS531	CLOUD COMPUTING	PEC	2	0	2	3.0
2	UCS532	COMPUTER VISION	PEC	2	0	2	3.0
3	UCS534	COMPUTER & NETWORK SECURITY	PEC	2	0	2	3.0
4	UMC513	LINEAR ALGEBRA FOR ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING	PEC	2	0	2	3.0
5	UCS539	FINANCE, ACCOUNTING AND VALUATION	PEC	2	0	2	3.0
6	UCS537	SOURCE CODE MANAGEMENT	PEC	2	0	2	3.0
7	UCS542	UI & UX SPECIALIST	PEC	2	0	2	3.0
8	UCS551	CONVERSATION AL AI: ACCELERATED DATA SCIENCE	PEC	2	0	2	3.0
9	UCS668	SCIENCE EDGE AI AND ROBOTICS: DATA CENTRE VISION	PEC	2	0	2	3.0
10	UCS550	NETWORK DEFENCE	PEC	2	0	2	3.0
11.	UEC646	NETWORK AND COMMUNICATION FOR CONNECTED VEHICLES	PEC	3	0	0	3.0

Elective II

S.No.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS635	GPU COMPUTING	PEC	2	0	2	3.0
2	UCS636	3D MODELLING AND ANIMATION	PEC	2	0	2	3.0
3	UCS638	SECURE CODING	PEC	2	0	2	3.0
4	UMC632	FINANCIAL MATHEMATICS	PEC	2	0	2	3.0
5	UCS675	FINANCIAL MARKETS AND PORTFOLIO THEORY	PEC	2	0	2	3.0
6	UCS659	BUILD AND RELEASE MANAGEMENT	PEC	2	0	2	3.0
7	UCS661	DATABASE ENGINEER	PEC	2	0	2	3.0
8	UCS664	CONVERSATIONAL AI: NATURAL LANGUAGE PROCESSING	PEC	2	0	2	3.0
9	UCS547	EDGE AI AND ROBOTICS: ACCELERATED DATA	PEC	2	0	2	3.0
10	UCS673	ETHICAL HACKING-1	PEC	2	0	2	3.0
11	UCS678	INTELLIGENT TRANSPORTATION SYSTEMS	PEC	3	0	0	3.0

Elective III

S. No.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS645	PARALLEL & DISTRIBUTED COMPUTING	PEC	2	0	2	3.0
2	UCS646	GAME DESIGN & DEVELOPMENT	PEC	2	0	2	3.0
3	UCS648	CYBER FORENSICS	PEC	2	0	2	3.0
4	UMC633	MATHEMATICS FOR QUANTUM COMPUTING	PEC	2	0	2	3.0
5	UCS658	DERIVATIVES PRICING, TRADING AND STRATEGIES	PEC	2	0	2	3.0
6	UCS660	CONTINUOUS INTEGRATION AND CONTINUOUS DEPLOYMENT	PEC	2	0	2	3.0
7	UCS662	TEST AUTOMATION	PEC	2	0	2	3.0
8	UCS749	CONVERSATIONAL AI: SPEECH PROCESSING & SYNTHESIS	PEC	2	0	2	3.0
9	UCS671	EDGE AI AND ROBOTICS: EMBEDDED VISION	PEC	2	0	2	3.0
10	UCS674	ETHICAL HACKING-II	PEC	2	0	2	3.0
11	UCS679	DATA ANALYTICS IN AUTOMOBILE ENGINEERING	PEC	3	0	0	3.0

Elective IV

S. No.	COURSE NO.	TITLE	CODE	L	T	P	CR
1	UCS751	SIMULATION & MODELLING	PEC	2	0	2	3.0
2	UCS752	AUGMENTED AND VIRTUAL REALITY	PEC	2	0	2	3.0
3	UCS754	BLOCKCHAIN TECHNOLOGY AND APPLICATIONS	PEC	2	0	2	3.0
4	UMC744	CRYPTOGRAPHY AND CODING THEORY	PEC	2	0	2	3.0
5	UMC743	QUANTITATIVE AND STATISTICAL METHODS FOR FINANCE	PEC	2	0	2	3.0
6	UCS758	SYSTEM PROVISIONING AND CONFIGURATION MANAGEMENT	PEC	2	0	2	3.0
7	UCS745	CLOUD & DEVOPS	PEC	2	0	2	3.0
8	UCS748	GENERATIVE AI	PEC	2	0	2	3.0
9	UCS760	EDGE AI AND ROBOTICS: REINFORCEMENT LEARNING & CONVERSATIONAL AI	PEC	2	0	2	3.0
10	UCS750	COMPUTER HACKING AND FORENSIC INVESTIGATION	PEC	2	0	2	3.0
11	UCS762	CYBER SECURITY FOR MOBILITY SYSTEMS	PEC	3	0	0	3.0

Generic Elective

S. No.	Course No.	Course Name	CODE	L	T	P	Cr
1.	UHU016	Introductory Course in French	OEC	2	0	0	2.0
2.	UHU017	Introduction to Cognitive Science	OEC	2	0	0	2.0
3.	UHU018	Introduction to Corporate Finance	OEC	2	0	0	2.0
4.	UCS002	Introduction to Cyber Security	OEC	2	0	0	2.0
5.	UPH064	Nanoscience and Nanomaterials	OEC	2	0	0	2.0
6.	UEN006	Technologies for Sustainable Development	OEC	2	0	0	2.0
7.	UMA069	Graph Theory and Applications	OEC	2	0	0	2.0
8.	UBT510	Biology for Engineers	OEC	2	0	0	2.0
9.	UMA070	Advanced Numerical Methods	OEC	2	0	0	2.0
10.	UTD004	Campus 2 Corporate	OEC	2	0	0	2
11.	UHUXXX	Gender, Identity and Technology	OEC	2	0	0	2
12.	UHU051	Creative Writing	OEC	2	0	0	2

SEMESTER I

UCB009: Chemistry

L	T	P	Cr
3	0	2	4.0

Course Objective: The course aims at elucidating principles of applied chemistry in industrial systems, water treatment, engineering materials, computational and analytical techniques.

Syllabus

Atomic and Molecular spectroscopy: Introduction to spectroscopy, principles of atomic absorption, flame emission spectrophotometry and ICP-AES (Inductively Coupled Plasma-Atomic Emission Spectroscopy), Quantification by calibration method, Jablonski diagram, fluorescence and phosphorescence, Beer-Lambert's Law, principle and applications of UV-Vis and IR spectroscopy.

Electrochemistry: Background of electrochemistry, Ionic mobility, Conductometric titrations, Modern Batteries: Pb-acid and Li ion battery, Corrosion and its protection.

Water Treatment and Analysis: Physiochemical parameters of water quality, External and internal methods of Softening of water: carbonate, phosphate, calgon and colloidal conditioning, Zeolite process, Ion exchange process, treatment of water for domestic use, Desalination of brackish water: Reverse osmosis & Electrodialysis.

Fuels: Classification of fuels, Calorific value, Cetane and Octane number, alternative fuels: biodiesel, Power alcohol, synthetic petrol, Fuel cells: H2 production and storage, Water splitting, Rocket propellant.

Chemistry of Polymers: Classification of polymers, tacticity of polymers, molecular weight calculations, Polymers in daily life, conducting, inorganic and biodegradable polymers.

Computers in Chemistry: Introduction to SMILES (Simplified Molecular Input Line-Entry System): Methodology and encoding rules, SMILES notation-chemical structure interconversions and its applications.

Laboratory Work

Electrochemical measurements: Experiments involving use of pH meter, conductivity meter, potentiometer, Spectroscopic technique, Volumetric titrations: Determination of mixture of bases, hardness, alkalinity, chloride and iron content, Application of polymers and SMILES Language.

Course Learning Objectives (CLO)

The students will be able to:

1. Recognize principles and applications of atomic and molecular spectroscopy.
2. Explain the concepts of conductometric titrations, modern batteries and corrosion.

3. Apply and execute water quality parameter and treatment methods.
4. Discuss the concept of alternative fuels, application of polymers and SMILES.
5. Execute laboratory techniques like pH metry, potentiometry, spectrophotometry, conductometry and volumetry.

Text Books

1. Engineering Chemistry, S. Vairam and S. Ramesh, Wiley India 1st ed, 2014.
2. Engineering Chemistry, K. S. Maheswaramma, and M. Chugh. Pearson, 2016.

Reference Books

1. Engineering Chemistry, B. Sivasankar, Tata McGraw-Hill Pub. Co. Ltd, New Delhi, 2008.
2. Engineering Chemistry, M.J. Shulz, Cengage Learnings, 2007.
3. J. Chem. Inf. Comput. Sci., D. Weininger, Vol. 28, 1988, 31-36.

Evaluation Scheme

Sr. No.	**Evaluation elements**	**Weightage (%)**
1	MST	25-30
2	EST	40-45
3	Sessional: (May include the following) Assignment, Sessional (Includes Regular Lab assessment and Quizzes Project (Including report, presentation etc.)	30

UES103: Programming for Problem Solving (C language)

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course is designed to solve and explore the problems using the art of computer programming with the help of C Language. Students will be able to apply these problem solving concepts in real life applications.

Introduction to Computer Fundamentals- Computer Memory Hierarchy, Types of Software Binary number system, Algorithm, Flowchart, Formulate simple algorithms for logical and arithmetic problems.

Basics of C Programming: Structure and Life cycle of a C Program, Data types, Identifiers, Variables, Keywords, Constants, input/output statements, Operators, Type conversion and type casting. Translate the algorithms to code snippets.

Decision Making and Iterative Statements-Decision making- if, if-else, Nested if-else, Multiple if, else if, switch, Ternary Operator, **Loops-** (while, do-while, for), Nesting of Loops, break, continue and goto. Implement the switch () to solve the basic functions of scientific calculator.

Functions: Function prototype, Definition and Call, Type of Functions, Scope of variables in (Block, Function, Program, File), Storage classes (Auto, Register, Static and Extern), Recursion (with the introduction of Stack), Implementation of recursion to solve the problem of Tower of Hanoi.

Arrays and Strings- One-dimensional array its operations (Traversal, Linear Search, Insertion, Deletion, Bubble Sort), Two-dimensional and its operations (Addition, Transpose and Multiplication), Passing of array into a function (row and entire array), Input and output of a string, string inbuilt functions, 2-D Character array.

Pointers: Introduction to Pointers, Pointer arithmetic, Passing arguments to a function using pointer (understanding of call by value and call by reference), Accessing arrays using pointers Dynamic memory allocation (malloc(), calloc(), realloc() and free()), Pointer and Functions.

Structures and Union: Structure declaration, Initialization of structures, Structure variables, Accessing structure elements using (.) operator, Array of structure variables, Passing structure variable to a function (individual and entire structure), Structure pointer, Comparison of Structure and Union.

File Handling: Introduction of Files (streams in C), using File (Declaring, Opening and Closing), Operations on File (Reading, Writing and appending), and Random Access of

a file, command line argument.

Laboratory Work

To implement programs for various kinds of real life applications in C Language.

Course Learning Outcomes (CLOs) /Course Objectives (COs):

On completion of this course, the students will be able to:

1. Comprehend and analyze the concepts of number system, memory, compilation and debugging of the programs in C language.
2. Analyze the control & iterative statements to solve the problems with C language source codes.
3. Design and create programs for problem solving involving arrays, strings and pointers.
4. Evaluateandanalyzetheprogrammingconceptsbasedonuserdefinedatatypesandfilehandlingusing C language.

Text Books

1. CProgrammingLanguage, BrianW.KernighanDennisM. Ritchie,2nd ed,2012.
2. ProgramminginANSIC, BalagurusamyG., 8th ed., 2019

Reference Books

1. LetUs C, KanetkarY.,16th ed.,2017
2. Programming with C, Byron S Gottfried,McGraw Hill Education, Forth edition, 2018

UES013: Electrical and Electronics Engineering

L	T	P	Cr
3	1	2	4.5

Course Objective: To introduce the basic concepts of electrical and electronics engineering.

DC Circuits: Introduction to circuit elements; rms and average values for different wave shapes, independent and dependent current and voltage sources; Kirchhoff's laws; mesh and node analysis; source transformations; network theorems: Superposition theorem, Thevenin's and Norton's theorem, Maximum power transfer theorem; star-delta transformation; steady state and transient response of R-L and R-C and R-L-C circuits.

AC Circuits: Concept of phasor, phasor representation of circuit elements; analysis of series and parallel AC circuits; concept of real, reactive and apparent powers; resonance in RLC series and parallel circuits; balanced three phase circuits: voltage, current and power relations for star and delta arrangement; analysis of balanced and unbalanced circuits; three phase power measurement using two-wattmeter and one-wattmeter methods.

Magnetic circuits: analogy between electric and magnetic circuits; series and parallel magnetic circuits; operating principles of electrical appliances: single-phase transformer and rotating machines; tests and performance of single-phase transformer.

Digital Logic Design: Digital signals, Number systems, Positive and negative representation of numbers, Signed-number representation, Binary arithmetic, Postulates and theorems of Boolean Algebra**,** Algebraic simplification, Sum of products and product of sums formulations (SOP and POS), Gate primitives, Logic Gates and Universal Gates, Minimization of logic functions, Karnaugh Maps, Logic implementation using Gates, Decoder, MUX, Flip-Flops, Asynchronous up/down counters.

Electronic Devices: p- n junction diode: V-I characteristics of diode, Operation of Bipolar Junction Transistor, CB and CE configuration, Transistor as a switch, Operation of SCR, DIAC and TRIAC.

Operational Amplifier Circuits: The ideal operational amplifier, the inverting, non-inverting amplifiers, Op-Amp Characteristics, Applications of Op-amp: summing amplifier, differentiator and integrator.

Laboratory Work: Kirchhoff's laws, network theorems, ac series and parallel circuit, three phase power measurement, magnetic circuit, tests on transformer, resonance in AC circuit, combinational circuits, flip flops, shift register and binary counters, asynchronous and synchronous up/down counters, BJT characteristics.

Course Learning Objectives (CLO)

The students will be able to:

1. Apply various networks laws and theorems to solve dc circuits

2. Compute different ac quantities with phasor representation
3. Comprehend the operation in magnetic circuits, single phase transformer and rotating machines
4. Recognize and apply the number systems and Boolean algebra.
5. Reduce and simplify Boolean expressions and implement them with logic gates.
6. Discuss and explain the working of diode, transistor and operational amplifier, their configurations and applications.

Text Books

1. Hughes, E., Smith, I.M., Hiley, J. and Brown, K., Electrical and Electronic Technology, Prentice Hall (2008) 10th ed.
2. Nagrath, I.J. and Kothari, D.P., Basic Electrical Engineering, Tata McGraw Hill (2002).
3. Boylestad, R.L. and Nashelsky, L., Electronic Devices & Circuit Theory, Perason (2009).
4. Mano M. M. and Ciletti, M.D., Digital Design, Pearson, Prentice Hall, (2013).
5. Reference Books
6. Chakraborti, A., Basic Electrical Engineering, Tata McGraw−Hill (2008).
7. Del Toro, V., Electrical Engineering Fundamentals, Prentice−Hall of India Private Limited (2004).
8. David Bell, Electronics Devices and Circuits, Oxford Publications (2009).

Evaluation Scheme

Sr. No.	**Evaluation elements**	**Weightage (%)**
1	MST	25-30
2	EST	40-45
3	Sessional: (May include the following) Assignment, Sessional (Includes Regular Lab assessment and Quizzes Project (Including report, presentation etc.)	30

UEN008: Energy and Environment

L	T	P	Cr
2	0	0	2.0

Course Objective: The exposure to this course would facilitate the students in understanding the terms, definitions and scope of environmental and energy issues pertaining to current global scenario; understanding the need of sustainability in addressing the current environmental & energy challenges.

Introduction: Concept of sustainability and sustainable use of natural resources, Climate Change & its related aspects.

Air Pollution: Origin, Sources and effects of air pollution; Primary and secondary meteorological parameters; wind roses; Atmospheric stability; Source reduction and Air Pollution Control Devices for particulates and gaseous pollutants in stationary sources.

Water Pollution: Origin, Sources of water pollution, Category of water pollutants, Physicochemical characteristics, Components of wastewater treatment systems.

Solid waste management: Introduction to solid waste management, Sources, characteristics of municipal solid waste, Solid waste management methods: Incineration, composting, landfilling.

Energy Resources: Classification of Energy Resources; Non-conventional energy resources- Biomass energy, Thermo-chemical conversion and biochemical conversion route; Solar energy-active and passive solar energy absorption systems; Type of collectors; Thermal and photo conversion applications.

Course Learning Objectives (CLO)

The students will be able to:

1. Comprehend the interdisciplinary context of environmental issues with reference to sustainability
2. Assess the impact of anthropogenic activities on the various elements of environment and apply suitable techniques to mitigate their impact.
3. Demonstrate the application of technology in real time assessment and control of pollutants.
4. Correlate environmental concerns with the conventional energy sources associated and assess the uses and limitations of non-conventional energy technologies

Text Books

1. Moaveni, S., Energy, Environment and Sustainability, Cengage (2018)
2. Rajagopalan, R., Environmental Studies, Oxford University Press (2018)
3. O'Callagan, P.W., Energy Management, McGraw Hill Book Co. Ltd. (1993).

Reference Books

1. Peavy H.S., Rowe D.S., and Tchobanoglous, G. (2013) Environmental Engineering, McGraw Hill.
2. Rao, M.N. and Rao, H.V.N. (2014) Air Pollution, McGraw Hill.
3. Metcalf and Eddy. (2003) Wastewater Engineering: Treatment and Reuse, Fourth Edition, McGraw Hill.
4. Rai, G.D. (2014) Non-conventional Energy Resources, Khanna Publishers.

Evaluation Scheme

Sr. No.	**Evaluation elements**	**Weightage (%)**
1	MST	25-30
2	EST	40-45
3	Sessional: (May include the following) Assignment, Sessional (Includes Regular Lab assessment and Quizzes Project (Including report, presentation etc.)	30

UMA022: Calculus for Engineers

L	T	P	Cr
3	1	0	3.5

Course Objective: To provide students with skills and knowledge in sequence and series, advanced calculus, calculus of several variables and complex analysis which would enable them to devise solutions for given situations they may encounter in their engineering profession.

Syllabus

Sequences and Series: Introduction to sequences and infinite series, Tests for convergence/divergence, Limit comparison test, Ratio test, Root test, Cauchy integral test, Alternating series, Absolute convergence, and conditional convergence.

Series Expansions: Power series, Taylor series, Convergence of Taylor series, Error estimates, Term by term differentiation and integration.

Partial Differentiation: Functions of several variables, Limits and continuity, Chain rule, Change of variables, Partial differentiation of implicit functions, Directional derivatives and its properties, Maxima and minima by using second order derivatives.

Multiple Integrals: Double integral (Cartesian), Change of order of integration in double integral, Polar coordinates, Graphing of polar curves, Change of variables (Cartesian to polar), Applications of double integrals to areas and volumes, Evaluation of triple integral (Cartesian).

Complex analysis: Introduction to complex numbers, Geometrical interpretation, Functions of complex variables, Examples of elementary functions like exponential, trigonometric and hyperbolic functions, Elementary calculus on the complex plane (limits, continuity, differentiability), Cauchy – Riemann equations, Analytic functions, Harmonic functions.

Course Outcomes(COs)

The students will be able to:

Determine the convergence/divergence of infinite series, approximation of functions using power and Taylor's series expansion and error estimation.

Examine functions of several variables, define and compute partial derivatives, directional derivatives, and their use in finding maxima and minima in some engineering problems.

Evaluate multiple integrals in Cartesian and Polar coordinates, and their applications to engineering problems.

Represent complex numbers in Cartesian and Polar forms and test the analyticity of complex functions by using Cauchy – Riemann equations.

Text Books

Thomas, G.B. and Finney, R.L., Calculus and Analytic Geometry, Pearson Education (2007), 9th ed.

Stewart James, Essential Calculus; Thomson Publishers (2007), 6th ed.

Kasana, H.S., Complex Variables: Theory and Applications, Prentice Hall India, 2005 (2nd edition).

Reference Books

Wider David V, Advanced Calculus: Early Transcendentals, Cengage Learning (2007).

Apostol Tom M, Calculus, Vol I and II, John Wiley (2003).

Brown J.W and Churchill R.V, Complex variables and applications, McGraw Hill, (7th edition)

Evaluation Scheme

Sr. No.	**Evaluation elements**	**Weightage (%)**
1	MST	25-30
2	EST	40-45
3	Sessional: (May include the following) Assignment, Sessional (Includes Regular Lab assessment and Quizzes Project (Including report, presentation etc.)	30

UAI101: Foundation of Machine Intelligence: Concepts, Techniques and Applications

L	T	P	Cr
2	0	0	2.0

AI Essentials: Definition and scope of AI, History and evolution of AI, AI vs. Human Intelligence, AI in engineering and technology.

Intelligent Agents and Environment Interaction: Definition and Types of Intelligent Agents, Agents and Environments: How agents perceive and act.

Knowledge Representation and Reasoning: Introduction to Knowledge Representation, Logic-based approaches: Propositional and Predicate Logic, Rule-based Systems and Expert Systems.

Introduction to Data and its analysis: Types of data: Structured, Semi-structured, Unstructured, Data Streams, Statistical Data types, Sampling.

Data Analysis Methods: Descriptive Analysis, Exploratory Analysis, Inferential Analysis, Predictive Analysis.

Machine Learning Essentials: Introduction to machine learning, Types of learning (Supervised, Unsupervised, Semi-Supervised, and Reinforcement Learning, Self - Supervised Learning) Introduction to Neural Networks and Deep Learning.

Recent Advancements in AI: Explainable AI (XAI), Edge AI, Generative AI, AI for Social Good (Applications of AI in healthcare, education, climate change, and sustainability), AI Security (Protecting AI systems from attacks and ensuring secure AI development)

Case studies of AI-driven innovations: Smart Speakers, Self-Driven cars, Smart Agriculture.

Text Books / Reference Books:

1. M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, 2nd ed. MIT Press, 2018.
2. E. Alpaydin, *Introduction to Machine Learning*, 4th ed. MIT Press, 2020.

Course Learning Objectives

1. Understand and Describe Fundamental AI Concepts and Intelligent Agents
2. Apply Knowledge Representation and Data Analysis Techniques
3. Differentiate and Implement Machine Learning Techniques
4. Evaluate and Discuss Recent Advancements and Innovations in AI

SEMESTER II

UPH013: Physics

L	T	P	Cr
3	1	2	4.5

Course Objective: To introduce the student to the basic physical laws of oscillators, acoustics of buildings, ultrasonics, electromagnetic waves, wave optics, lasers, and quantum mechanics and demonstrate their applications in technology. To introduce the student to measurement principles and their application to investigate physical phenomena.

Oscillations and Waves: Oscillatory motion and damping, Applications - Electromagnetic damping – eddy current; ***Acoustics:*** Reverberation time, absorption coefficient, Sabine's and Eyring's formulae (Qualitative idea), Applications - Designing of hall for speech, concert, and opera; ***Ultrasonics:*** Production and Detection of Ultrasonic waves, Applications - green energy, sound signaling, dispersion of fog, remote sensing, Car's airbag sensor.

Electromagnetic Waves: Scalar and vector fields; Gradient, divergence, and curl; Stokes' and Green's theorems; Concept of Displacement current; Maxwell's equations; Electromagnetic wave equations in free space and conducting media, Application - skin depth.

Optics: ***Interference:*** Parallel and wedge-shaped thin films, Newton rings, Applications as Non-reflecting coatings, Measurement of wavelength and refractive index. ***Diffraction:*** Single and Double slit diffraction, and Diffraction grating, Applications - Dispersive and Resolving Powers. ***Polarization:*** Production, detection, Applications – Anti-glare automobile headlights, Adjustable tint windows. ***Lasers:*** Basic concepts, Laser properties, Ruby, HeNe, and Semiconductor lasers, Applications – Optical communication and Optical alignment.

Quantum Mechanics: Wave function, Steady State Schrodinger wave equation, Expectation value, Infinite potential well, Tunneling effect (Qualitative idea), Application - Quantum computing.

Laboratory Work

1. Determination of damping effect on oscillatory motion due to various media.
2. Determination of velocity of ultrasonic waves in liquids by stationary wave method.
3. Determination of wavelength of sodium light using Newton's rings method.
4. Determination of dispersive power of sodium-D lines using diffraction grating.
5. Determination of specific rotation of cane sugar solution.
6. Study and proof of Malus' law in polarization.

7. Determination of beam divergence and beam intensity of a given laser.
8. Determination of displacement and conducting currents through a dielectric.
9. Determination of Planck's constant.

Micro Project:
Students will be given physics-based projects/assignments using computer simulations, etc.

Course Learning Objectives (CLO)
The students will be able to:
1. understand damped and simple harmonic motion, the role of reverberation in designing a hall and generation and detection of ultrasonic waves.
2. use Maxwell's equations to describe propagation of EM waves in a medium.
3. demonstrate interference, diffraction and polarization of light.
4. explain the working principle of Lasers.
5. use the concept of wave function to find probability of a particle confined in a box.
6. perform an experiment, collect data, tabulate and report them and interpret the results with error analysis.

Text Books
1. Beiser, A., Concept of Modern Physics, Tata McGraw Hill (2007) 6th ed.
2. Griffiths, D.J., Introduction to Electrodynamics, Prentice Hall of India (1999) 3rd ed.
3. Jenkins, F.A. and White, H.E., Fundamentals of Optics, McGraw Hill (2001) 4th ed.

Reference Books
1. Wehr, M.R, Richards, J.A., Adair, T.W., Physics of The Atom, Narosa Publishing House (1990) 4th ed.
2. Verma, N.K., Physics for Engineers, Prentice Hall of India (2014)1st ed.
3. Pedrotti, Frank L., Pedrotti, Leno S., and Pedrotti, Leno M., Introduction to Optics, Pearson Prentice HallTM (2008) 3rd ed.

Evaluation Scheme

Sr. No.	Evaluation elements	Weightage (%)
1	MST	25-30
2	EST	40-45
3	Sessional: (May include the following) Assignment, Sessional (Includes Regular Lab assessment and Quizzes Project (Including report, presentation etc.)	30

UES101: Engineering Drawing

L	T	P	Cr
2	4	0	4.0

Course Objective: This module is dedicated to graphics and includes two sections: 2D drafting and 3D modelling of solid objects. This course is aimed at making the student understand the concepts of projection systems, learn how to create projections of solid objects using first and third angle orthographic projection as well as isometric and auxiliary projection, concept of sectioning, to interpret the meaning and intent of toleranced dimensions and to create/edit drawings using drafting software. In addition, this course shall
give an insight on the basic 3D modelling concepts like extrude, revolve, sweep, construction
of complex solids.

Syllabus

Engineering Drawing Concepts

1. Introduction to Engineering Drawing
2. Projection systems: First angle and third angle projection system
3. Orthographic Projection: Points, Lines, Solid objects
4. Isometric Projections
5. Auxiliary Projections
6. Development of surfaces
7. Section of solids
8. Limits, fits and tolerances

2D Drafting

1. Management of screen menus commands
2. Creating basic drawing entities
3. Co-ordinate systems: Cartesian, polar and relative coordinates
4. Drawing limits, units of measurement and scale
5. Layering: organizing and maintaining the integrity of drawings
6. Design of prototype drawings as templates.
7. Editing/modifying drawing entities: selection of objects, object snap modes, editing commands,
8. Dimensioning: use of annotations, dimension types, properties and placement, adding text to drawing

3D Modelling

1. Management of screen menus commands
2. Introduction to basic 3D modelling commands such as extrude, revolve, sweep etc.
3. Creation of 2D drawings from a 3D model

Micro Projects /Assignments:

1. Completing the views - Identification and drawing of missing lines and views in the projection of objects

2. Projects related to orthographic and isometric projections Using wax blocks/soap bars/any soft material to develop three dimensional object from given orthographic projections
3. a. 3D modelling of complex machine components
   b. Development of production drawings of individual components from the model

Course Learning Objectives (CLO)
The students will be able to:
1. creatively comprehend the geometrical details of common engineering objects
2. draw dimensioned orthographic and isometric projections of simple engineering objects
3. interpret the meaning and intent of limits, fits and tolerances in the drawing
4. create/edit the engineering drawings for simple engineering objects using 2D drafting software
5. create/edit 3D models of engineering components using 3D modelling software

Text Books
1. Jolhe, D.A., Engineering Drawing, Tata McGraw Hill, 2008
2. Davies, B. L., Yarwood, A., Engineering Drawing and Computer Graphics, Van Nostrand Reinhold (UK), 1986

Reference Books
1. Gill, P.S., Geometrical Drawings, S.K. Kataria & Sons, Delhi (2008).
2. Gill, P.S., Machine Drawings, S.K. Kataria & Sons, Delhi (2013).
3. Mohan, K.R., Engineering Graphics, Dhanpat Rai Publishing Company (P) Ltd, Delhi (2002).
4. French, T. E., Vierck, C. J. and Foster, R. J., Fundamental of Engineering Drawing & Graphics Technology, McGraw Hill Book Company, New Delhi (1986).
5. Rowan, J. and Sidwell , E. H., Graphics for Engineers, Edward Arnold, London (1968).
6. Mastering AutoCAD 2021 and AutoCAD LT 2021, Brian C. Benton, George Omura, Sybex - John Wiley and Sons, Indiana (2021).

Evaluation Scheme

Course Component	**Weightage**
AutoCAD tutorials/SolidWorks/Project work*	35
MST (1.5 hours-CAD based)**	20
EST (2 hours-CAD based)**	45

***Students are required to bring their personal computers for the** tutorial work.*
**Availability of institute server resources for sharing the software licences with the student community.*
***Institute computational resources in collaboration with other academic units / departments for conducting the mid semester and end semester test.*

UHU003: Professional Communication

L	T	P	Cr
2	0	2	3.0

Course Objective: The course is designed to develop the interpersonal, written, and oral as well as the non- verbal communication skills of the students. The course begins by building up on the theoretical concepts and then practicing on the applicability of the various elements. Since the course has very high applicability content, the students are advised to practice in class as well as off class. A very high level of interaction is expected of the students in the class.

Syllabus

Fundamentals of Communication: Meaning, Types and Characteristics of communication, Applicability of Transactional Analysis and Johari Window for enhancing interpersonal communication skills. Seven Cs of Effective Communication, Barriers to Effective Communication.

Effective Oral Communication: Understanding Principles of Oral communication, Formal and Informal Oral Communication, Oral Communication and Behavioral Patterns, Advantages and Disadvantages of Oral Communication.

Effective Listening: Listening vs Hearing, Active Listening techniques, Barriers to Listening.

Effective non-verbal communication: Meaning and Importance of Non-Verbal Communication, Different Types of Non-verbal Communication, Interpretation of Non-verbal Cues.

Effective written Communication: Characteristics of Good Writing, Choice of Words, Sentence Construction, Paragraph development, Forms of writing.

Business Communication: Technical Report Writing, Designing Resumes and Cover Letters for effective job application, E-mail writing and e-mail etiquette.

Organizational Communication: Directional communication: Downward, Upward and Horizontal Communication, Grapevine.

Reading: The following texts (one from each of the two categories listed below) are required to be read by the students in the semester:
Category 1: Animal Farm by George Orwell, Lord of the Flies by William Golding, Life of Pi by Yann Martel
Category 2: The Namesake by Jhumpa Lahiri, The God of Small Things by Arundhati Roy, Q&A by Vikas Swarup

Laboratory Work

1. Needs-assessment of spoken and written communication with feedback.

2. Training for Group Discussions through simulations and role plays.
3. Technical report writing on survey-based projects.
4. Project-based team presentations.

Course Learning Objectives (CLO)

The students will be able to:

1. Apply communication concepts for effective interpersonal communication.
2. Speak assertively and effectively.
3. Interpret non-verbal cues in professional communication.
4. Write objectively, purposefully and effectively.
5. Design effective resumes and reports.

Text Books

1. Mukherjee H.S..Business Communication: Connecting at Work. Oxford University Press.(2013)
2. Lesikar R.V, and Flately M.E., Basic Business Communication Skills for empowering the internet generation.(2006)
3. Raman, M.,and Singh ,P, Business Communication . Oxford . University Press (2008).

Reference Books

1. Riordan, G.R. Technical Communication. Cengage Learning India Private Ltd. (2012)
2. Butterfield, Jeff., Soft Skills for everyone, Cengage Learning New Delhi, (2013).
3. Robbins, S.P., & Hunsaker, P.L., Training in Interpersonal Skills, Prentice Hall of India, New Delhi, (2008).
4. Orwell, G., Animal Farm, Fingerprint Publishing, New Delhi, (2017).
5. Golding, W, Lord of the Flies, Faber & Faber; Export edition (1999)
6. Martel,Y., Life of Pi, RHC, New Delhi, (2012).
7. Lahiri,J., The Namesake, Harpercollins (2007)
8. Arundhati Roy,A., The God of Small Things, Penguin India, (2002).
9. Swarup,V., Q&A, Black Swan,(2009).

Evaluation Scheme

Sr. No.	Evaluation elements	Weightage (%)
1	MST	25-30
2	EST	40-45
3	Sessional: (May include the following) Assignment, Sessional (Includes Regular Lab assessment and Quizzes Project (Including report, presentation etc.)	30

UES102: Manufacturing Processes

L	T	P	Cr
2	0	2	3.0

Course Objective: This course introduces the basic concepts of manufacturing via machining, forming, casting and joining, enabling the students to develop a basic knowledge of the mechanics, operation and limitations of basic machining tools along with metrology and measurement of parts. The course also introduces the concept of smart manufacturing.

Machining Processes: Principles of metal cutting, Cutting tools, Cutting tool materials and applications, Geometry of single point cutting tool, Introduction to computerized numerical control (CNC) machines, G and M code programming for simple turning and milling operations, introduction of canned cycles.

Metal Casting: Introduction & Principles of sand casting, Requisites of a sound casting, Permanent mold casting processes, casting defects

Metal Forming: Hot & cold metal working, Forging, Rolling, Sheet Metal operations.

Joining Processes: Method of joining, type of electric arc welding processes, Methods of shielding, Power source characteristics, Resistance welding, Soldering, Brazing.

Smart Manufacturing: IoT and ML in manufacturing, Introduction to Additive Manufacturing, Robotics and Automation in manufacturing.

Laboratory Work

Relevant shop floor exercises involving practices in Sand casting, Machining, Welding, Sheet metal fabrication techniques, CNC turning and milling exercises, Experiments on basic engineering metrology and measurements to include measurements for circularity, ovality, linear dimensions, profiles, radius, angular measurements, measurement of threads, surface roughness.

Basic knowledge and derivations related to above measurements, uncertainties, statistical approaches to estimate uncertainties, Line fitting, static and dynamic characteristics of instruments will be discussed in laboratory classes.

Assignments: Assignments for this course will include the topics: Manufacturing of micro-chips used in IT and electronics industry and use of touch screens. Another assignment will be given to practice numerical exercises on topics listed in the syllabus. Case study related to smart manufacturing.

Micro Project: Fabrication of multi-operational jobs using the above processes as per requirement by teams consisting of 4 -6 members. Quality check should be using the equipment available in metrology lab.

Course Learning Objectives (CLO)

The students will be able to:

1. identify & analyse various machining processes/operations for manufacturing of industrial components
2. apply the basic principle of bulk and sheet metal forming operations
3. apply the knowledge of metal casting for different requirements.
4. identify and analyse the requirements to for achieving a sound welded joint apply the concept of smart manufacturing

Text Books

1. Degarmo, E. P., Kohser, Ronald A. and Black, J. T., Materials and Processes in Manufacturing, Prentice Hall of India (2008) 8th ed.
2. Kalpakjian, S. and Schmid, S. R., Manufacturing Processes for Engineering Materials, Dorling Kingsley (2006) 4th ed.

Reference Books

1. Martin, S.I., Chapman, W.A.J., Workshop Technology, Vol.1 & II, Viva Books (2006) 4th ed.
2. Zimmer, E.W. and Groover, M.P., CAD/CAM - Computer Aided Designing and Manufacturing, Dorling Kingsley (2008).
3. Pandey, P.C. and Shan, H. S., Modern Machining Processes, Tata McGraw Hill (2008).
4. Mishra, P. K., Non-Conventional Machining, Narosa Publications (2006).
5. Campbell, J.S., Principles of Manufacturing, Materials and Processes, Tata McGraw Hill Company (1999).
6. Lindberg, Roy A., Processes and Materials of Manufacture, Prentice Hall of India (2008) 4th ed.

Evaluation Scheme

Sr. No.	**Evaluation elements**	**Weightage (%)**
1	MST	25-30
2	EST	40-45
3	Sessional: (May include the following) Assignment, Sessional (Includes Regular Lab assessment and Quizzes Project (Including report, presentation etc.)	30

UMA023: Differential Equations and Linear Algebra

L	T	P	Cr
3	1	0	3.5

Course Objective: To introduce students the theory and concepts of differential equations, linear algebra, Laplace transformations and Fourier series which will equip them with adequate knowledge of mathematics to formulate and solve problems analytically.

Syllabus

Ordinary Differential Equations: Review of first order differential equations, Exact differential equations, Second and higher order differential equations, Solution techniques using one known solution, Cauchy - Euler equation, Method of undetermined coefficients, Variation of parameters method, Engineering applications of differential equations.

Laplace Transform: Definition and existence of Laplace transforms and its inverse, Properties of the Laplace transforms, Unit step function, Impulse function, Applications to solve initial and boundary value problems.

Fourier Series: Introduction, Fourier series on arbitrary intervals, Half range expansions, Applications of Fourier series to solve wave equation and heat equation.

Linear Algebra: Row reduced echelon form, Solution of system of linear equations, Matrix inversion, Linear spaces, Subspaces, Basis and dimension, Linear transformation and its matrix representation, Eigen-values, Eigen-vectors and Diagonalisation, Inner product spaces and Gram-Schmidt orthogonalisation process.

Course Outcomes(COs)

The students will be able to:

Solve the differential equations of first and 2nd order and basic application problems described by these equations.

Find the Laplace transformations and inverse Laplace transformations for various functions. Using the concept of Laplace transform students will be able to solve the initia value and boundary value problems.

Find the Fourier series expansions of periodic functions and subsequently will be able to solve heat and wave equations.

Solve systems of linear equations by using elementary row operations.

Identify the vector spaces/subspaces and to compute their bases/orthonormal bases. Further, students will be able to express linear transformation in terms of matrix and find the eigenvalues and eigenvectors.

Text Books

Simmons, G.F., Differential Equations (With Applications and Historical Notes), Tata McGraw

Hill (2009).
Krishnamurthy, V.K., Mainra, V.P. and Arora, J.L., An introduction to Linear Algebra, Affiliated East West Press (1976).

Reference Books

1. Kreyszig Erwin, Advanced Engineering Mathematics, John Wiley (2006), 8th edition.
Jain, R.K. and Iyenger, S.R.K., Advanced Engineering Mathematics, Narosa Publishing House (2011), 4th edition.

Evaluation Scheme

Sr. No.	**Evaluation elements**	**Weightage (%)**
1	MST	25-30
2	EST	40-45
3	Sessional: (May include the following) Assignment, Sessional (Includes Regular Lab assessment and Quizzes Project (Including report, presentation etc.)	30

UAI201: Programming Essentials for AI and ML

L	T	P	Cr
2	0	2	3.0

Introduction to Python programming: Setting up the Python environment (IDEs, Jupyter Notebooks), Writing and running Python scripts, Basic syntax, variables, and data types (integers, floats, strings)

Control Structures: `if`, `else`, `elif`, Loops: `for`, `while`, `break`, `continue`, `pass`.

Data Structures in Python: Lists and list operations (indexing, slicing, appending), Tuples and their immutability, Dictionaries: key-value pairs, adding and removing elements, Sets: basic operations and set methods.

Working with Strings and Files: String manipulation and formatting, Reading from and writing to files, Working with file paths, Handling exceptions in file operations. Using numpy, torch, mat and pickle files.

Error Handling and Debugging: Understanding exceptions and error types, Using `try`, `except`, `else`, `finally` for error handling, Debugging techniques and tools.

Libraries and APIs: Introduction to Python libraries (e.g., `requests`, `json`, `datetime`), Working with APIs to fetch and process data, Understanding JSON data format, Manipulating XML files. Introduction to Data Handling with Pandas: Introduction to the Pandas library, Working with Data Frames: loading, selecting, and filtering data, Basic data analysis and manipulation.

Text Books / Reference Books:

1. A. Sweigart, *Automate the Boring Stuff with Python: Practical Programming for Total Beginners*, 1st ed. No Starch Press, 2015.
2. E. Matthes, *Python Crash Course*, 2nd ed. No Starch Press, 2019.
3. M. Lutz, *Learning Python*, 5th ed. O'Reilly Media, 2013.

Course Learning Objectives

1. Understand and Apply Basic Programming Concepts
2. Implement Control Structures and Data Handling Techniques
3. Perform String Manipulation, File Operations, and Error Handling
4. Utilize Python Libraries for Data Handling and API Interaction

SEMESTER III

UCS303: OPERATING SYSTEMS

L	T	P	Cr
3	0	2	4.0

Course Objectives: To understand the role, responsibilities, and the algorithms involved for achieving various functionalities of an Operating System.

Introduction and System Structures: Computer-System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security, Computing Environments Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, System Programs, Operating-System Design and Implementation, Operating-System Structure, Operating-System Design & Virtualization.

Process Management: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication, Multi-threaded programming: Multi-core Programming.

Multithreading Models, Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Algorithm Evaluation.

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

Memory Management: Basic Hardware, Address Binding, Logical and Physical Address, Dynamic linking and loading, Shared Libraries, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table, Virtual Memory Management: Demand Paging Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Allocating Kernel Memory.

File Systems: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing, Protection, File-System Structure, File-System Implementation Directory Implementation, Allocation Methods, Free-Space Management.

Disk Management: Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap-Space Management, RAID Structure.

Protection and Security: Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of the Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, The Security Problem, Program Threats, System and Network Threats, User Authentication, Implementing Security Defenses, Firewalling to Protect Systems and Networks.

Concurrency: The Critical- Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors.

Laboratory work:
To explore detailed architecture and shell commands in Linux / Unix environment, Understanding of OS virtualization simulate CPU scheduling, Paging, Disk-scheduling and process synchronization algorithms and simulate multithreading processing scenarios with a focus on energy-efficient scheduling. Students has to submit a mini project.

Sample Mini Projects:

- Develop a basic task manager that monitors CPU and memory usage and suggests actions to save energy.
- Simulate efficient virtual memory management techniques for optimized performance in industrial systems.
- Modify a disk scheduling algorithm to reduce energy consumption.
- Create a user-friendly interface for visually impaired users to interact with an OS.
- Simulate an OS that distributes processes efficiently across multiple cores.

Course Outcomes (COs):
After the completion of the course, the student will be able to:

1. Explain the basic of an operating system viz. system programs, system calls, user mode and kernel mode.
2. Select a particular CPU scheduling algorithms for specific situation, and analyze the environment leading to deadlock and its rectification.
3. Explicate memory management techniques viz. caching, paging, segmentation, virtual memory, and thrashing.
4. Understand the concepts related to file systems, disk-scheduling, and security, protection.
5. Comprehend the concepts related to concurrency.

Text Books:

1. Silberschatz A., Galvin B. P. and Gagne G., Operating System Concepts, John Wiley & Sons Inc (2013) 9th ed.
2. Stallings W., Operating Systems Internals and Design Principles, Prentice Hall (2018) 9th ed.

Reference Books:

1. Bovet P. D., Cesati M., Understanding the Linux Kernel, O'Reilly Media (2006), 3rd ed.
2. Kifer M., Smolka A. S., Introduction to Operating System Design and Implementation: The OSP 2 Approach, Springer (2007).

UAI301 Object Oriented Programming

L	T	P	Cr
3	**0**	2	**4.0**

Course Overview: This course introduces the principles of Object-Oriented Programming using Python. Students will learn how to design and implement reusable, modular, and scalable code using classes, objects, inheritance, polymorphism, encapsulation, and other OOP concepts.

Introduction to OOP Concepts: What is Object-Oriented Programming? Difference between procedural and object-oriented programming, Core OOP principles: Encapsulation, Abstraction, Inheritance, Polymorphism, Classes vs. Objects: Definitions and examples, Creating your first class in Python.

Working with Classes and Objects: Defining attributes and methods, The __init__ method (constructor), Instance variables vs. class variables, Accessing and modifying object properties, Understanding self keyword, Deleting objects and attributes. Case Study on Optimizing Power Consumption in Smart City Energy Grids.

Encapsulation and Access Modifiers: Public, protected, and private members, Name mangling in Python, Getters and setters, Data hiding and abstraction.

Inheritance: Types of inheritance: single, multiple, multilevel, hierarchical, Base and derived classes, Method overriding, Use of super() function, isinstance() and issubclass() functions.

Polymorphism: Method overloading (simulation in Python), Operator overloading, Duck typing and polymorphic behavior, Polymorphism with inheritance and interfaces. Case Study on Developing an Eco-Friendly Smart Transportation System.

Advanced OOP Concepts: Class methods and static methods (@classmethod, @staticmethod), Abstract base classes using abc module, Magic methods (__str__, __repr__, __add__, etc.), Composition vs. inheritance, Managing complex projects with modules and packages.

Error Handling & Debugging in OOP: Custom exceptions in classes, Exception handling inside methods, Logging errors and debugging OOP code, PEP8 guidelines for OOP.

Laboratory Work:
To implement object-oriented constructs using the Python programming language.

Course Learning Objectives (CLO)
The students will be able to:

1. To recall the knowledge of structure and its variables to comprehend the concept of classes, objects, constructors and destructors for implementing the object oriented paradigms.
2. To apply and analyze the inheritance on real life case studies via coding competences.
3. To design and develop code snippets for polymorphism to proclaim coding potential; and management of run-time exceptions.
4. To assess and interpret the knowledge of templates to appraise the standard template libraries.

Text Books:

1. Phillips, D. (2015). Python 3 Object-Oriented Programming (2nd ed.). Packt Publishing.
2. Matthes, E. (2019). Python Crash Course: A Hands-On, Project-Based Introduction to Programming . No Starch Press.

Reference Books:

1. Zelle, J. M. (2016). Python Programming: An Introduction to Computer Science (3rd ed.). Franklin, Beedle & Associates.
2. Lutz, M. (2013). Learning Python (5th ed.). O'Reilly Media.

UCS301: **Data Structures**

L	T	P	Cr
3	0	2	4.0

Course Objective: To become familiar with different types of data structures and their applications.

Syllabus

Analysing algorithms: Basics of algorithm and its analysis, Complexity classes, order arithmetic, Time and space trade-off in algorithms. Practical implications of inefficient algorithms on energy consumption, need of minimizing computational overhead for sustainability.

Linear Data Structures: Arrays, Strings and string processing, Linked lists (Singly, Doubly, Circular), Abstract data types, their implementation and applications: Stacks (using Arrays and Linked-list), Queues (using Arrays and Linked-list), Hash tables: Hash functions, collision resolution techniques, Strategies for choosing the appropriate data structure.

Searching and Sorting: Linear Search, Binary Search. Introduction to internal and external sort, Bubble Sort, Selection Sort, Insertion Sort, Shell Sort, Quick Sort, Merge Sort, Counting Sort, Radix Sort.

Trees and their applications: Introduction to binary tree, tree traversal algorithms, Binary search tree, AVL Tree, B Tree etc. and common operations on these trees. Spatial data structures (e.g., Quadtrees, R-trees etc.) for sustainable urban planning and waste management. Heap, Heap Sort, Priority Queue using Heap.

Graphs and their applications: Graph Terminology and its representation, Depth and breadth first traversals, Shortest-path algorithms (Dijkstra and Floyd), Data Structures for Disjoint Sets, Minimum spanning tree (Prim and Kruskal). Applications of graphs in smart grid management, waste collection routes, recycling networks etc.

Laboratory Work

Implementation of various data structures such as Arrays, Stacks, Queues, Lists, Binary tree traversals, BST, AVL trees, Graphs traversals, Sorting and Searching techniques.

Course Learning Objectives (CLO)

The students will be able to:

1. Understand the fundamental data structures, their implementation and some of their standard applications.

2. Select and implement appropriate searching and sorting techniques for solving a problem based on their characteristics.

3. Apply tree and graph data structures for specific applications.

4. Design and analyse algorithms using appropriate data structures for real-world problems.

Text Books

1. Introduction to Algorithms,Cormen H. T., Leiserson E. C., Rivest L. R., and Stein C, MIT Press,3rd ed., 2009
2. Data Structures, Algorithms and Applications in C++,Sahni S., Universities Press 2nd ed. 2005.

Reference Books

1. Data Structures and Algorithms Made Easy,Karumanchi N., Career Monk Publications, 5th ed., 2017
2. Data structures and algorithms in C++, Adam Drozdek, 4th edition.

UMA021: Numerical Linear Algebra

L	T	P	Cr
3	0	2	4.0

Course Objectives: The goal of this course is to give students an introduction to numeric and algorithmic techniques used for the solution of a broad range of mathematical problems, with an emphasis on computational issues and linear algebra. In addition, students will become familiar with numeric programming environments Matlab.

Contents:

Roots of Non-Linear Equations: Mathematical preliminaries, bisection, fixed-point, Newton's method and its extension to system of equations.

Interpolation and Integration: Lagrange and Newton basis of polynomials and interpolation problems, divided difference interpolation, forward and backward differences, trapezoidal and Simpson's rules, method of undetermined coefficients.

Matrix Algebra: Gauss elimination method, pivoting strategies, matrix factorization, Jacobi and GaussSeidel methods, matrix norm and conditioning, linear least square problems.

Matrix Computations: Orthogonal and orthonormal basis, Gram-Schmidt process, orthogonal matrices and similarity transformations, power method for eigen-value and eigen-vector, QR algorithm, singular value decomposition.

Laboratory Work:

Lab experiments will be set in consonance with materials covered in the theory and the implementation of numerical methods will be done using MATLAB

Course Learning Outcomes (CLOs) /Course Objectives (COs):

On completion of this course, the students will be able to:

1. Make use of iterative methods to solve nonlinear equations.
2. Approximate the functions using interpolating polynomials and apply to definite integrals.
3. Evaluate solution of system of linear equations and least square problems.
4. Perform matrix computations and evaluate eigen-values and eigen-vectors.

Text Books

1. Richard L. Burden, J. Douglas Faires, and Annette Burden, Numerical Analysis, Cengage Learning, 10th edition, 2015.
2. Gilbert Strang, Linear Algebra and its Applications, Cengage Learning, 4th edition, 2005.
3. J. Desmond Higham and Nicholas J. Higham, MATLAB Guide, Third Edition, Society for Industrial and Applied Mathematics, 2016.

Reference Books

1. Steven C. Chapra and Raymond P. Canale, Numerical Methods for Engineers, McGraw-Hill Higher Education, 6th edition, 2010.
2. E. Ward Cheney and David R. Kincaid, Numerical Mathematics and Computing, Cengage Learning, 7th edition, 2012.
3. Endre Suli and David F. Mayers, An Introduction to Numerical Analysis, Cambridge University Press, 2003

UTA016: ENGINEERING DESIGN PROJECT – I
(including 2 self-effort hours)

L	T	P	Cr
1	0	2	3.0

Course Objectives: To develop design skills according to a Conceive-Design-Implement-Operate (CDIO) compliant methodology. To apply engineering sciences through learning-by- doing project work. To provide a framework to encourage creativity and innovation. To develop teamwork and communication skills through group-based activity. To foster self-directed learning and critical evaluation.

To provide a basis for the technical aspects of the project a small number of lectures are incorporated into the module. As the students would have received little in the way of formal engineering instruction at this early stage in the degree course, the level of the lectures is to be introductory with an emphasis on the physical aspects of the subject matter as applied to the 'Mangonel' project. The lecture series include subject areas such as Materials, Structures, Dynamics and Digital Electronics delivered by experts in the field.

This module is delivered using a combination of introductory lectures and participation by the students in 15 "activities". The activities are executed to support the syllabus of the course and might take place in specialised laboratories or on the open ground used for firing the Mangonel. Students work in groups throughout the semester to encourage teamwork, cooperation and to avail of the different skills of its members. In the end the students work in sub-groups to do the Mangonel throwing arm redesign project. They assemble and operate a Mangonel, based on the lectures and tutorials assignments of mechanical engineering they experiment with the working, critically analyse the effect of design changes and implement the final project in a competition. Presentation of the group assembly, redesign and individual reflection of the project is assessed in the end.

Breakup of lecture details to be taken up by MED:

Lec No.	Topic	Contents
Lec1	INTRODUCTION	The Mangonel Project, History, Spreadsheet.
Lec2	PROJECTILE MOTION	No DRAG, Design spreadsheet simulator for it.
Lec3	PROJECTILE MOTION	With DRAG, Design spreadsheet simulator for it.
Lec4	STRUCTURES FAILURE	STATIC LOADS
Lec5	STRUCTURES FAILURE	DYNAMIC LOADS

Lec6	REDESIGNING THE MANGONEL	Design constraints and limitations of materials for redesigning the Mangonel for competition as a group.
Lec7	MANUFACTURING	Manufacturing and assembling the Mangonel.
Lec8	SIMULATION IN ENGINEERING DESIGN	Simulation as an Analysis Tool in Engineering Design.
Lec9	ROLE OF MODELLING & PROTOTYPING	The Role of Modelling in Engineering Design.

Breakup of lecture details to be taken up by ECED:

Lec No.	Topic	Contents
Lec1-5	Digital Electronics	Prototype, Architecture, Using the Integrated Development Environment (IDE) to Prepare an Arduino Sketch, structuring an Arduino Program, Using Simple Primitive Types (Variables), Simple programming examples. Definition of a sensor and actuator.

Tutorial Assignment / Laboratory Work:

Associated Laboratory / Project Program: T – Mechanical Tutorial, L – Electronics Laboratory, W – Mechanical Workshop of "Mangonel" assembly, redesign, operation and reflection.

Title for the weekly work in 15 weeks	Code
Using a spread sheet to develop a simulator	T1
Dynamics of projectile launched by a Mangonel – No Drag	T2
Dynamics of projectile launched by a Mangonel – With Drag	T3
Design against failure under static actions	T4
Design against failure under dynamic actions	T5
Electronics hardware and Arduino controller	L1
Electronics hardware and Arduino controller	L2
Programming the Arduino Controller	L3
Programming the Arduino Controller	L4
Final project of sensors, electronics hardware and programmed Arduino controller based measurement of angular velocity of the "Mangonel" throwing arm.	L5
Assembly of the Mangonel by group	W1

Assembly of the Mangonel by group	W2
Innovative redesign of the Mangonel and its testing by group	W3
Innovative redesign of the Mangonel and its testing by group	W4
Final intergroup competition to assess best redesign and understanding of the "Mangonel".	W5

Project:
The Project will facilitate the design, construction and analysis of a "Mangonel". In addition to some introductory lectures, the content of the students' work during the semester will consist of:

1. The assembly of a Mangonel from a Bill Of Materials (BOM), detailed engineering drawings of parts, assembly instructions, and few prefabricated parts;
2. The development of a software tool to allow the trajectory of a "missile" to be studied as a function of various operating parameters in conditions of no-drag and drag due to air;
3. A structural analysis of certain key components of the Mangonel for static and dynamic stresses using values of material properties which will be experimentally determined;
4. The development of a micro-electronic system to allow the angular velocity of the throwing arm to be determined;
5. Testing the Mangonel;
6. Redesigning the throwing arm of the Mangonel to optimise for distance without compromising its structural integrity;
7. An inter-group competition at the end of the semester with evaluation of the group redesign strategies.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
Upon completion of this module, students will be able to:

1. Simulate trajectories of a mass with and without aerodynamic drag using a spreadsheet based software tool to allow trajectories be optimized;
2. Perform a test to acquire an engineering material property of strength in bending and analyze the throwing arm of the "Mangonel" under conditions of static and dynamic loading;
3. Develop and test software code to process sensor data;
4. Designn, construct and test an electronic hardware solution to process sensor data;
5. Construct and operate a Roman catapult "Mangonel" using tools, materials and assembly instructions, in a group, for a competition;
6. Operate and evaluate the innovative redesign of elements of the "Mangonel" for functional and structural performance;

Text Books:

1. Michael Mc Roberts, Beginning Arduino, Technology in action publications.
2. Alan G. Smith, Introduction to Arduino: A piece of cake, Create Space Independent Publishing Platform (2011).

Reference Book:

1. John Box all, Arduino Workshop – A Hands-On Introduction with 65 Projects, No Starch Press (2013).

SEMESTER IV

UCS415: DESIGN AND ANALYSIS OF ALGORITHMS

L	T	P	Cr
3	0	2	4.0

Course Objective: To provide students with the knowledge and skills necessary to design and analyse algorithms for solving computational problems.

Introduction and Complexity Analysis: Analysing algorithms, Complexity classes, Time and space trade-offs in algorithms, Recurrence relations, Analysis of iterative and recursive algorithms, Amortized Analysis.

Algorithm Design Techniques and Analysis

Divide and Conquer: Fundamentals of divide and conquer strategy, Applications such as The maximum subarray problem, Strassen's algorithm for matrix multiplication, merge sort, quick sort etc.

Greedy Algorithms: Elements of greedy strategy, Sustainable resource allocation using greedy algorithms, Applications such as activity selection, Huffman Coding for energy efficiency in data transmission, job sequencing, fractional knapsack problem, waste bin placement, load balancing in smart grid etc.

Dynamic Programming: Elements of dynamic programming, Memorization and tabulation approaches, Applications such as matrix multiplication, 0/1 knapsack (Optimizing Energy Storage), Longest common subsequence, Optimal binary search tree, etc.

Backtracking: Introduction, Applications such as N queen problem, sum of subsets, graph coloring, etc.

Branch and Bound Algorithm: General method, Applications such as0/1 knapsack problem, Traveling salesperson problem etc.

Graphs & Algorithms: Introduction to graphs, Paths and Circuits, Euler Graphs, Hamiltonian graphs, Applications of Eulerian/Hamiltonian Graph in urban planning, Cut-sets, Strongly connected component, Topological sort, Max flow: Ford Fulkerson algorithm, max flow- min cut, Application of Ford Fulkerson in smart grid energy distribution.

String Matching Algorithms: Suffix arrays, Rabin-Karp, Knuth-Morris Pratt (KMP), Boyer Moore algorithm.

Problem Classes: P, NP, NP-Hard and NP-complete, deterministic and non-deterministic polynomial time algorithm approximation, Randomized algorithms.

Laboratory Work (if applicable): Implementation of various algorithmic techniques for solving common computational/engineering problems.

Course Learning Objectives (CLO)

The students will be able to:

1. Analyse the complexity of algorithms and implement it in a specific scenario.
2. Apply common algorithmic techniques such as greedy, dynamic programming etc. to standard computational problems.
3. Design solutions using appropriate data structures and algorithms, such as string matching, randomized, approximation and graph algorithms.
4. Develop efficient algorithms to solve various computationally complex problems in computing.

Text Books

1. Cormen H. T., Leiserson E. C., Rivest L. R., and Stein C., Introduction to Algorithms, MIT Press (2009) 3rd ed.
2. Horwitz E., Sahni S., Rajasekaran S., Fundamentals of Computers Algorithms, Universities Press (2008) 2nd ed.

Reference Books

1. Levitin A., Introduction to the design and analysis of algorithms, Pearson Education (2008) 2nd ed.
2. Aho A.V., Hopcraft J. E., Dulman J. D., The Design and Analysis of Computer Algorithms, Addsion Wesley (1974) 1st ed.
3. Sedgewick R. and Wayne K., Algorithms, Addison-Wesley Professional (2011), 4th ed.

UCS310:Database Management Systems				
	L	T	P	Cr
	3	0	2	4.0

Course Objective:Emphasis is on the need of database systems. Main focus is on E-R diagrams, relational database, concepts of normalization and de-normalization and SQL commands.

Syllabus

Introduction: Data, data processing requirement, desirable characteristics of an ideal data processing system, traditional file-based system, its drawback, concept of data dependency, Definition of database, types of database, database management system, 3-schema architecture, database terminology, benefits of DBMS.

Relational Database: Relational data model: Introduction to relational database theory: definition of relation, keys, relational model integrity rules, introduction to Relational Algebra.

Database Analysis: Conceptual data modeling using E-R data model -entities, attributes, relationships, generalization, specialization, specifying constraints, Conversion of ER Models to Tables, Practical problems based on E-R data model.

Database Design: Functional Dependency, Canonical Covers, Candidate Key Identification, Normalization- 1NF, 2NF, 3NF, BCNF, 4NF and 5NF. Concept of De-normalization and practical problems based on these forms.

Transaction Management and Concurrency control: Concept of Transaction, States of Transaction and ACID properties, Need of Concurrency control, concept of Lock, Two phase locking protocol.

Recovery Management: Need of Recovery Management, Concept of Stable Storage, Log Based Recovery Mechanism, Checkpoint.

Database Implementation: Introduction to SQL, DDL aspect of SQL, DML aspect of SQL – update, insert, delete & various form of SELECT- simple, using special operators, aggregate functions, group by clause, sub query, joins, co-related sub query, union clause, View, exist operator. PL/SQL - cursor, stored function, stored procedure, triggers, error handling, and package.

Laboratory Work

Students will perform SQL commands to demonstrate the usage of DDL and DML, joining of tables, grouping of data and will implement PL/SQL constructs. They will also implement one project.

Project: It will contain database designing & implementation, should be given to group of 2-4 students. While doing projects emphasis should be more on back-end programming like use of SQL, concept of stored procedure, function, triggers, cursors, package etc. Project should have continuous evaluation and should be spread over different components.

Healthcare data: including patient records, treatment statistics, and disease surveillance, can be managed in databases to improve healthcare services and outcomes.[3]

Construct a Databases that can manage educational resources, student performance data, and learning materials, helping ensure quality education is accessible and measurable.[4]

Data management systems can support urban planning, resource allocation, and sustainability efforts, helping to build smarter, more resilient cities.[11]

Course Learning Objectives (CLO)

The students will be able to:

1. Analyze the Information Systems as socio-technical systems, its need and advantages as compared to traditional file-based systems.
2. Analyze and design database using E-R data model by identifying entities, attributes and relationships.
3. Apply and create Relational Database Design process with Normalization and Denormalization of data.
4. Comprehend the concepts of transaction management, concurrence control and recovery management.
5. Demonstrate use of SQL and PL/SQL to implementation database applications.

Text Books

1. Database System Concepts, Silverschatz A., Korth F. H. and Sudarshan S., Tata McGraw Hill, 6th ed, 2010
2. Fundamentals of Database Systems,Elmasri R. and Navathe B. S., Pearson, 7th ed, 2016

Reference Books

1. SQL, PL/SQL the Programming Language of Oracle, Bayross I., BPB Publications, 4th ed, 2009
2. Modern Database Management,Hoffer J., Venkataraman, R. and Topi, H., Pearson, 12th ed2016
3. Simplified Approach to DBMS, Parteek Bhatia and Gurvinder Singh,
4. Database management systems. Vol. 3. Raghu Ramakrishnan and Johannes Gehrke
5. FOR SQL/RA, New York: McGraw-Hill,

UMA401 PROBABILITY AND STATISTICS

L	T	P	Cr
3	0	2	4

Course Objectives: This course shall make the students familiar with the concepts of Probability and Statistics useful in implementing various computer science models. One will also be able to associate distributions with real-life variables and make decisions based on statistical methods.

Introduction to Statistics and Data Analysis: Introduction to Statistical Inference, Samples, Populations and Experimental Design, Collection of Data, Measures of location and variability, Graphical representation of data.

Probability: Sample space, Events, Classical, relative frequency and axiomatic definitions of probability, addition rule and conditional probability, multiplication rule, total probability, Baye's Theorem.

Random Variables: Discrete, continuous and mixed random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, moments, probability and moment generating function, median and quantiles, Markov inequality, Chebyshev's inequality, Function of a random variable.

Special Distributions: Discrete uniform, binomial, geometric, negative binomial, Poisson, continuous uniform, exponential, gamma, normal, lognormal, inverse Gaussian, Cauchy, double exponential distributions, reliability of series and parallel systems.

Joint Distributions: Joint, marginal and conditional distributions, product moments, correlation and regression, independence of random variables, bivariate normal distribution.

Sampling Distributions: The Central Limit Theorem, distributions of the sample mean and the sample variance for a normal population, Chi-Square, t and F distributions.

Estimation: Unbiasedness, consistency, the method of moments and the method of maximum likelihood estimation, confidence intervals for parameters in one sample and two sample problems of normal populations, confidence intervals for proportions.

Testing of Hypotheses: Null and alternative hypotheses, the critical and acceptance regions, two types of error, power of the test, the most powerful test and Neyman-Pearson Fundamental Lemma, tests for one sample and two sample problems for normal populations, tests for proportions, Chi-square goodness of fit test and its applications.

Laboratory Work:
Implementation of statistical techniques using statistical packages viz. SPSS/R including evaluation of statistical parameters and data interpretation, regression analysis, covariance, hypothesis testing and analysis of variance.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After completion of this course, the students will be able to:

1. Analyze the data using different descriptive measures and present graphically.
2. Compute the probabilities of events along with an understanding of the random variables.
3. Comprehend the concept of statistical distributions, their properties and relevance to real-life data.
4. Understand the estimation of mean and variance and their respective hypothesis tests.

Text Books:

1. Probability & Statistics for Engineers & Scientists by R.E. Walpole, R.H. Myers, S.L. Myers & Keying Ye, Prentice Hall, (2016), 9th edition.
2. An Introduction to Probability and Statistics by V.K. Rohatgi & A.K. Md. E. Saleh, Wiley, (2008), 2nd edition

Reference Books:

1. Miller and Freund's – Probability and Statistics for Engineers by R. A. Johnson, Person Education, (2017), 9th edition.
2. Introduction to Probability and Statistics for Engineers and Scientists by S.M. Ross, Elsevier, (2014), 4th edition.

UTA035: ENGINEERING DESIGN PROJECT – II

L	T	P	Cr
1	0	4	3.0

Course Objectives: The project will introduce students to the challenge of electronic systems design & integration. The project is an example of 'hardware and software co-design' and the scale of the task is such that it will require teamwork as a co-ordinated effort.

Hardware overview of Arduino:

- Introduction to Arduino Board: Technical specifications, accessories and applications.
- Introduction to Eagle (PCB layout tool) software.

Sensors and selection criterion:

- Concepts of sensors, their technical specifications, selection criterion, working principle and applications such as IR sensors, ultrasonic sensors.

Active and passive components:

- Familiarization with hardware components, input and output devices, their technical specifications, selection criterion, working principle and applications such as-
  - Active and passive components: Transistor (MOSFET), diode (LED), LCD, potentiometer, capacitors, DC motor, Breadboard, general PCB etc.
  - Instruments: CRO, multimeter, Logic probe, solder iron, desolder iron
  - Serial communication: Concept of RS232 communication, Xbee
- Introduction of ATtiny microcontroller based PWM circuit programming.

Programming of Arduino:

- Introduction to Arduino: Setting up the programming environment and basic introduction to the Arduino micro-controller
- Programming Concepts: Understanding and Using Variables, If-Else Statement, Comparison Operators and Conditions, For Loop Iteration, Arrays, Switch Case Statement and Using a Keyboard for Data Collection, While Statement, Using Buttons, Reading Analog and Digital Pins, Serial Port Communication, Introduction programming of different type of sensors and communication modules, DC Motors controlling.

Basics of C#:

- Introduction: MS.NET Framework Introduction, Visual Studio Overview and Installation
- Programming Basics: Console programming, Variables and Expressions, Arithmetic Operators, Relational Operators, Logical Operators, Bitwise Operators, Assignment Operators, Expressions, Control Structures, Characters, Strings, String Input, serial

port communication: Read and write data using serial port.

- Software code optimization, software version control

Laboratory Work:
Schematic circuit drawing and PCB layout design on CAD tools, implementing hardware module of IR sensor, Transmitter and Receiver circuit on PCB.

Bronze Challenge: Single buggy around track twice in clockwise direction, under full supervisory control. Able to detect an obstacle.Parks safely. Able to communicate state of the track and buggy at each gantry stop to the console.

Silver Challenge: Two buggies, both one loop around, track in opposite directions under full supervisory, control. Able to detect an obstacle. Both park safely. Able to communicate state of the track and buggy at each gantry stop with console.

Gold Challenge: Same as silver but user must be able to enter the number of loops around the track beforehand to make the code generalized.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
The student will be able to:

1. Recognize issues to be addressed in a combined hardware and software system design.
2. Draw the schematic diagram of an electronic circuit and design its PCB layout using CAD Tools.
3. Apply hands-on experience in electronic circuit implementation and its testing.
4. Demonstrate programming skills by integrating coding, optimization and debugging for different challenges.
5. Develop group working, including task sub-division and integration of individual contributions from the team.

Text Books:

1. Michael McRoberts, Beginning Arduino, Technology in action publications, 2nd Edition.
2. Alan G. Smith, Introduction to Arduino: A piece of cake, CreateSpace Independent Publishing Platform (2011).

Reference Book:

1. John Boxall, Arduino Workshop - a Hands-On Introduction with 65 Projects, No Starch Press; 1st edition (2013).

UTD002: EMPLOYABILITY DEVELOPMENT SKILLS

L	T	P	Cr
2	0	0	2.0

Course Objectives:
This course aims to sensitize students with the gamut of skills which facilitate them to enhance their employability quotient and do well in the professional space. These skills are imperative for students to establish a stronger connect with the environment in which they operate. An understanding of these skills will enable students to manage the placement challenges more effectively.

Emotional Intelligence: Understanding Emotional Intelligence (EI); Daniel Goleman's EI Model: Self Awareness, Self-Regulation, Internal Motivation, Empathy, Social Skills; Application of EI during Group Discussions & Personal Interview; Application of EI in personal life, student life and at the workplace.

Team Dynamics & Leadership: Understanding the challenges of working within a team format in today's complex organizational environments; Stages of team formation; Appreciating forces that influence the direction of a team's behaviour and performance; Cross-functional teams; Conflict in Teams- leveraging differences to create opportunity Leadership in the team setting & energizing team efforts; Situational leadership; Application of team dynamics & collaboration in Group Discussions; Application of team dynamics at the workplace.

Complex Problem Solving: Identifying complex problems and reviewing related information to develop and evaluate options and implement solutions; Understanding a working model for complex problem solving - framing the problem, diagnosing the problem, identifying solutions & executing the solutions; Appreciation of complex problem solving at the workplace through case studies.

Lateral Thinking: Understanding lateral thinking & appreciating the difference between vertical & lateral thinking, and between convergent & divergent thinking; Understanding brain storming & mind-maps; Solving of problems by an indirect and creative approach, typically through viewing the problem in a new and unusual light; Application of lateral thinking during Group Discussions & Personal Interviews; Application of lateral thinking at the workplace.

Persuasion: Role of persuasion in communication; Application of ethos-pathos-logos; Using persuasive strategies to connect with individuals & teams to create competitive advantage.

Quantitative Reasoning: Thinking critically and applying basic mathematics skills to interpret data, draw conclusions, and solve problems; developing proficiency in numerical reasoning; Application of quantitative reasoning in aptitude tests.

Verbal Reasoning: Understanding and reasoning using concepts framed in words; Critical verbal reasoning; Reading Comprehension; Application of verbal reasoning in aptitude tests.

Group Discussion (GD): Illustrating the do's and don'ts in Group Discussions; Specific thrust on types of GD topics; GD evaluation parameters; Understanding the challenge in a case discussion; SPACER model .

Personal Interview (PI): Interview do's and don'ts; PI evaluation parameters; The art of introduction; Managing bouncer questions; Leading the panel in a PI

Course Learning Outcomes (CLOs):
The students will be able to

1. appreciate the various skills required for professional & personal success.
2. bridge the gap between current and expected performance benchmarks.
3. competently manage the challenges related to campus placements and perform to their utmost potential.

Recommended Books:

1. *Harvard Business Essentials; Creating Teams with an Edge; Harvard Business School Press (2004)*
2. *Edward de B., Six Thinking Hats; Penguin Life (2016)*
3. *Daniel, G., Working with Emotional Intelligence; Bantam Books (2000)*
4. *Aggarwal, R.S., Quantitative Aptitude for Competitive Examinations; S Chand (2017)*
5. *Agarwal, A., An expert guide to problem solving: with practical examples; CreateSpace Independent Publishing Platform (2016)*
6. *William, D., The Logical Thinking process; American Society for Quality (2007)*

Evaluation Scheme:

UCS405 Discrete Mathematical Structures

L	T	P	Cr
3	1	0	3.5

Course Objective: The course objective is to provide students with an overview of Discrete Mathematical Structures. Students will learn about topics such as logic and proofs, sets and functions, graph theory, boolean algebra, number theory and other important discrete math concepts.

Syllabus

Sets, Relations, and Functions: Sets: Operations on set, Inclusion-exclusion principle, Representation of Discrete Structures, Fuzzy set, Multi-set, bijective function, Inverse and Composition of functions, Floor and Ceiling functions, Growth of functions: Big-O notation, Big-Omega and Big-Theta Notations, Determining complexity of a program, Hash functions.

Relations: Different types of relation and their representation, Equivalence and partial-ordered relations, Partition and Covering of a set, N-ary relations and database, Closure of relations, Warshall's algorithm, Lexicographic ordering, Hasse diagram, Lattices, Boolean algebra.

Graphs Theory: Representation, Type of Graphs, Paths and Circuits: Euler Graphs, Hamiltonian Paths & Circuits; Cut-sets, Connectivity and Separability, Planar Graphs, Isomorphism, Graph Coloring, Covering and Partitioning, Application of Graph theory in real-life applications.

Basic Logic: Propositional logic, Logical connectives, Truth tables, Normal forms (conjunctive and disjunctive), Validity of well-formed formula, Propositional inference rules (concepts of modus ponens and modus tollens), Predicate logic, Universal and existential quantification, Proof Techniques.

Recurrence Relation: Solving linear recurrence relations, divide and conquer algorithms and recurrence relations.

Algebraic Structures: Group, Semi group, Monoids, Ring, Field, Homomorphism.

Number Theory: Divisibility and Modular Arithmetic, Solving Congruences, Applications of Congruences, Cryptographic applications

Laboratory Work

NA

Course Learning Objectives (CLO)

The students will be able to:

1. Perform operations on various discrete structures such as set, function, and relation.
2. Apply basic concepts of asymptotic notation in the analysis of the algorithm.
3. Illustrate the basic properties and algorithms of graphs and apply them in modelling and solving real-world problems.
4. Comprehend formal logical arguments and translate statements from a natural language into their symbolic structures in logic.
5. Identify and prove various properties of rings, fields, and groups.
6. Illustrate and apply the division algorithm, mod function, and Congruence.

Text Books

1. Discrete Mathematics and its Applications,Rosen H. K., McGraw Hill, 7thed., 2011
2. Discrete Mathematical Structures with Applications to Computer Science, Tremblay P. J. and Manohar, R., Tata McGraw Hill, 2008.

Reference Books

1. Contemporary Abstract Algebra, Gallian A. J., Cengage Learning, 9th ed., 2017
2. Discrete Mathematics, Lipschutz S., Lipson M., McGraw-Hill, 3rd ed.,2007

SEMESTER V

UAI504: COMPUTER NETWORKS

L	T	P	Cr
3	0	2	4.0

Course Objectives: The subject will introduce the basics of AI Networking and its' applications.

Introduction: Networks and Energy Efficient Networking, Networking Topologies, Latency, throughput, jitter, packet loss, TCP/IP Protocol Suite, Bandwidth-Delay and AI workloads, Ethernet vs InfiniBand fundamentals, Data Centers and Network Topologies: Leaf-Spine, Fat-tree, and Torus for HPC.

Remote Direct Memory Access (RDMA): Concept and Usage of RDMA in AI workloads, RDMA operations (Read/Write/Send/Recv), Kernel bypass, zero-copy, low-latency communication, RDMA over InfiniBand, RNIC architecture, Memory registration, QP, CQ, PD, GPU Direct RDMA.

AI Networking Hardware and Software: NICs: Mellanox ConnectX (5/6/7/8), Switches: Spectrum-3/4, Quantum-2 (InfiniBand), NIC Routing: Adaptive routing, Packet pacing, SHARP (in-network reduction), Software: OFED, RDMA-core, ibverbs (Creating Queue Pairs (QP), Registering memory for RDMA, Posting Send/Receive operations, Managing Completion Queues (CQ)), rdmacm (Resolving IP, Establishing RDMA connections, Handling events (connect, disconnect), Synchronizing QP setup), NCCL(AllReduce, AllGather, ReduceScatter, Broadcast).

Data Center Networking for AI: Switching Architectures (Arista, NVIDIA, Cisco), Routing vs Switching in AI clusters, Priority Flow Control (PFC), Explicit Congestion Notification (ECN), DCBX, RoCE configuration, QoS for AI traffic, Telemetry: sFlow, In-band telemetry.

Distributed AI, Communication Libraries, and Training: Distributed AI, Types of parallelism: Data, Tensor Model, Pipeline, Expert (MoE) parallelism, Bottleneck in LLM training, Communication Libraries: NCCL (NVIDIA), RCCL (AMD), PyTorch Distributed (Gloo, NCCL), TensorFlow Collective Ops, MPI with RDMA.

Laboratory work:
Students can perform RDMA programming in (C/C++ or Python), Creating QP, CQ, MR, PD, RDMA Read/Write, RDMA Send/Receive, GPUDirect RDMA, Monitoring RDMA traffic, Debugging QP states, and connection issues.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:
1. Conceptualize and explain the AI network architecture
2. Understand the RDMA concept of data communication, traffic detection, access and flow control.
3. Demonstrate the operation of various routing protocols their performance analysis.
4. Illustrate design and implementation of distributed AI and Data Center Networking within a simulated/real networking environment.

Text Books:

1. Forouzan A. B., Data communication and Networking, McGraw Hill (2012) 5th ed.
2. Shanley, T., InfiniBand Network Architecture, Addison-Wesley (2003).

Reference Books:

1. Kurose J. and Ross K., Computer Networking: A Top Down Approach, Pearson (2017) 7 th ed.
2.Abts D. & Kim J., High Performance Datacenter Networks: Architectures, Algorithms and Opportunities, Morgan & Claypool (2022).

UAI501: IMAGE PROCESSING

L	T	P	Cr
3	0	2	4.0

Course Objectives: To understand the basic concepts of Image Processing. The student must be able to apply the various concepts of Image Processing in other application areas.

Foundations & Advanced Image Representations: Image types: Hyperspectral, Multispectral, Medical and Satellite images. Image Metadata Standards: EXIF Metadata, IPTC & XMP Metadata, DICOM Metadata for Medical Imaging, Geo-Spatial Metadata. Bit-Depth & Radiometric Resolution: RAW Sensor Data, Quantization & Dynamic Range. Gamma Correction & Radiometric Transformations, Color spaces in AI applications (YCbCr, LAB, LUV). HDR Imaging Fundamentals, HDR Encoding & Standard, Tone Mapping & HDR to SDR Conversion.

Image Enhancement, Restoration in Spatial/Frequency Domain: Mean, median, and adaptive median filters, Fourier Transform, DFT, FFT, Wavelet transforms , Wiener filter (spatial/frequency domain implementation),Non-local means (NLM), Bilateral filtering, Anisotropic diffusion (Perona–Malik),Total Variation (TV) denoising, Guided filtering and rolling guidance filter, Motion blur estimation in the frequency spectrum, Gabor and wavelet-domain denoising, Multi-resolution frequency restoration (wavelet shrinkage, curvelets), Edge-preserving smoothing for deblurring, MAP (Maximum A Posteriori) frameworks, Sparse representation and dictionary learning (K-SVD), Low-light image restoration, Haze, fog, and atmospheric turbulence restoration, Smartphone image restoration pipelines, HDR reconstruction and raw sensor denoising, Decompression artifact removal (JPEG/AV1 artifact cleanup).

Image Compression: Lossless & lossy compression methods, Arithmetic Coding, JPEG, JPEG2000, CABAC/CAVLC (in H.264/HEVC), WebP (lossy + lossless), HEIF / HEIC, MPEG-2 (FFmpeg).

Feature Detection, and Image Matching: SIFT, SURF, ORB, AKAZE, RANSAC for robust matching, Deep feature extraction (CNN embeddings).

Video Processing and Real-Time Image Analytics: Frame rates and temporal sampling, motion estimation, optical flow, background subtraction (MOG2, KNN), object tracking methodologies (KCF, GOTURN, SORT, DeepSORT), real-time object detection pipelines, temporal filtering, frame interpolation, video stabilization.

Image Processing in Cutting-Edge Domains: Healthcare: MRI, CT, ultrasound image analysis, Segmentation for disease detection, Agriculture: Crop monitoring using drones, Leaf disease detection, Geospatial, Land cover classification, Disaster detection & assessment, Finance: Cheque image processing, Signature verification.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Understand and analyze foundational and advanced image representations.
2. Apply image enhancement, denoising, and restoration techniques in spatial and frequency domains.
3. Analyze and compare various image compression and encoding standards to determine their suitability for different imaging applications.
4. Analyze feature detection, image matching, and tracking techniques.
5. Develop and evaluate real-time image and video analytics systems for practical applications

Text Books:

1. 1. Digital Image Processing (4e) – Rafael C. Gonzalez & Richard Woods
2. Computer Vision: Algorithms and Applications – Richard Szeliski

Reference Books:

3. Pattern Recognition and Machine Learning – Christopher Bishop
4. Handbook of Image and Video Processing (2nd ed) - AI Bovik

UCS503: SOFTWARE ENGINEERING

L	T	P	Cr
3	0	2	4.0

Course Objectives: To plan and manage large-scale software and learn emerging trends in software engineering.

Software Engineering and Processes: Introduction to Software Engineering, Software Evolution, Software Characteristics, , Software Crisis: Problems and Causes, Software process models -Waterfall, Iterative, Incremental and Evolutionary process models, Role of Software in Sustainable Development,Green Coding Guidelines (Optimizing Memory & CPU Utilization), Regulatory Aspects & Global Policies on Green IT. ISO Standards for Green IT

Requirements Engineering: Problem Analysis, Requirement Elicitation and Validation, Requirement Analysis Approaches- Structured Analysis Vs Object Oriented Analysis, Flow modeling through Data Flow Diagram and Data Dictionary, Data Modeling through E-R Diagram, Requirements modeling through UML, based on Scenario, Behavioral and Class modeling, documenting Software Requirement Specification (SRS)

Software Design and construction: System design principles like levels of abstraction, separation of concerns, information hiding, coupling and cohesion, Structured design (top-down or functional decomposition), object-oriented design, event driven design, component-level design, test driven design, data design at various levels, architecture design like Model View Controller, Client – Server architecture. Coding Practices: Techniques, Refactoring, Integration Strategies, Internal Documentation, Green Software Architecture (Microservices vs. Monolith, Edge Computing)

Software Verification and Validation: Levels of Testing, Functional Testing, Structural Testing, Test Plan, Test Case Specification, Software Testing Strategies, Verification & Validation, Unit and Integration Testing, Alpha & Beta Testing, White box and black box testing techniques, System Testing and Overview of Debugging.

Agile Software Development: Agile Manifesto, Twelve Practices of eXtreme Programming (XP), XP values, XP practices, velocity, spikes, working of Scrum, product backlog, sprint backlog, Adaptive Software Development(ASD), Feature Driven Development (FDD), Test Driven Development, Dynamic System Development Method(DSDM), and Crystal Methodology,Agile Requirement and Design: User Stories, Story Boards, UI Sketching and Story Cards. Energy-Aware UI/UX Design (Dark Mode, Reduced Animations, Low Power UX Patterns)

Software Project Management: Overview of Project Management: Scope, Time and Cost estimations. Case Studies: Energy-Efficient Software Practices in Google, Microsoft, Meta

Laboratory work:
Implementation of Software Engineering concepts and exposure to CASE tools like Rational Software Suit through projects.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
On completion of this course, the students will be able to

1. Analyze software development process models for software development life cycle.
2. Elicit, describe, and evaluate a system's requirements and analyze them using various UML models.
3. Demonstrate the use of design principles in designing data, architecture, user and component level design.
4. Test the system by planning appropriate test cases and applying relevant test strategies.
5. Comprehend the use of agile development methodologies including UI sketching, user stories, story cards and backlog management.

Text Books:

1. Pressman R., Software Engineering, A Practitioner's Approach, McGraw Hill International, 7th ed. (2010).
2. Sommerville I., Software Engineering, Addison-Wesley Publishing Company, 9th ed. (2011).

Reference Books:

1. Jalote P.,An integrated Approach to Software Engineering, Narosa, 3rd ed. (2005).
2. Booch G.,Rambaugh J.,Jacobson I.,The Unified Modeling Language User Guide, 2nd ed. (2005).

UAI503: COMPUTER ARCHITECTURE AND ORGANIZATION

L	T	P	Cr.
3	0	0	3.0

Course Objectives: Focus is on the architecture and organization of the basic computer modules viz. controls unit, central processing unit, input-output organization and memory unit.

Basics of Computer Architecture: Number System and code conversion, Logic gates, Multiplexer, De-multiplexer, Decoder, Encoder etc. IEEE 754 Floating point representation. 32bit/64bit

Register Transfer and Micro operations: Register transfer Language, Register transfer, Bus & memory transfer, Arithmetic micro operations, Logic micro operations, Shift micro operations, Design of ALU Three state buffer, Binary Adder, Binary Incrementor.

Basic Computer Organization: Instruction codes, Computer instructions, Timing &control, Instruction Cycles, Memory, register, and input-output reference instructions, Interrupts, Complete computer description & design of basic computer. Direct and Indirect Address.

Central Processing Unit: General register organization, Stack organization, Instruction format, Addressing modes, Data transfer & manipulation, Program control, RISC, CISC. Register and memory stack, software and hardware interrupt.

Evolution of GPU Computing and Pipeline: GPU vs CPU Architecture, Streaming Multiprocessors (SMs): Structure and Function, CUDA Architecture and Programming Model, Tensor Cores: Matrix Operations, Mixed Precision, Addition &Subtraction, Multiplication Algorithms. Data Pipeline, Risk Pipeline Dependencies in a pipeline processor, pipeline hazards.

Memory Unit: Memory hierarchy, Processor vs. memory speed, High-speed memories, Main Memory, Cache memory, Associative memory, Interleaving, Virtual memory, Memory management techniques. Direct Mapping, Set Associative Mapping, Thread Scheduling and Memory Hierarchy in GPUs; GPU Interconnects (NVLink, PCIe, Multi-GPU Scaling); Applications in AI, ML, Scientific Computing & HPC.

Course Learning Outcomes (CLOs) / Course Objectives (COs): On completion of this course, the students will be able to

1. Illustrate various elementary concepts of computer architecture including, syntax of register transfer language, micro operations, instruction cycle, and control unit.

2. Describe the design of basic computer with instruction formats & addressing modes, Concepts of register and memory stack. Interpret Concepts of pipelining and related hazards.

3. Explore GPU architectures including SMs, CUDA/Tensor/Transformer cores, and their role in parallel processing and HPC.

4. Explore concepts of memory hierarchy, various memory management techniques, cache management and virtual memory and algorithms for performing addition, subtraction and division etc.

Text Books:

1. Mano, Morris M., Computer System Architecture, Prentice Hall (1991) 3rd ed. Page 77 of 226

2. Hayes, J.P., Computer Architecture and Organization, McGraw Hill (1998) 3rd ed.

3. William Stallings, Computer Organization and Architecture, Pearson (2018), 11th ed.

Reference Books:

1. Hennessy, J.L., Patterson, D.A, and Goldberg, D., Computer Architecture A Quantitative Approach, Pearson Education Asia (2006) 4th ed.

2. Leigh, W.E. and Ali, D.L., System Architecture: software and hardware concepts, South Wester Publishing Co. (2000) 2nd ed.

UAI502: ADVANCED DEEP LEARNING

L	T	P	Cr
3	0	2	4.0

Course Objective: The course aims to provide in-depth understanding of advanced deep learning methods and architectures. It builds on basic deep learning foundations and covers modern convolutional, sequence, transformer-based, and generative models, enabling students to analyze, design, and implement state-of-the-art systems.

Machine Learning and Deep Learning Preliminaries: Supervised Learning Basics, Underfitting and Overfitting, Bias–Variance Trade-off, Introduction to Deep Learning: Representation Learning, Deep Feedforward Networks, Backpropagation, Gradient Descent and Stochastic Gradient Descent. Activation Functions, Parameter Initialization, Vanishing and Exploding Gradients.

Deep Learning Foundations and Optimization Techniques: Stochastic Gradient Descent Variants: Mini-batch SGD, Momentum, Nesterov Gradient. Adaptive Optimization Methods: Adagrad, RMSProp, Adam. Normalization Layers: Batch Normalization, Layer Normalization, Weight Normalization. Regularization Techniques: L1/L2 Penalties, Dropout, Early Stopping.

Advanced Convolutional Architectures: Convolution Operation, Depthwise Separable and Dilated Convolution, Pooling, Stride, Popular CNN Architectures: VGG, Inception, ResNet, DenseNet, MobileNet, EfficientNet, ConvNeXt.

Word Representations and Sequence Modelling: Word2Vec (CBOW, Skip-Gram), GloVe Representations. Recurrent Neural Networks (RNNs), Bidirectional RNNs, Encoder–Decoder Architectures for Sequence Learning: Many-to-One, One-to-Many, Many-to-Many Networks.

Advanced Recurrent and Attention Models: Vanishing and Exploding Gradients in Sequence Models, Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM) Networks, Attention Mechanism, Transformers, BERT Architecture, Vision Transformers (ViT) and Transformer Variants.

Deep Generative Models: Autoencoders (Basic, Denoising, Sparse), Variational Autoencoders (VAE), Boltzmann Machines, Restricted Boltzmann Machines (RBM), Deep Belief Networks (DBN), Generative Adversarial Networks (GAN): Objective, Training Challenges

Course Learning Outcomes (CLOs):

1. Demonstrate understanding of deep learning foundations, and optimization techniques.
2. Compare and explain various deep learning architectures and algorithms.
3. Demonstrate the applications of deep learning in various fields.

4. Apply deep learning specific open-source libraries for solving real life problems.

Textbooks:

1. Goodfellow I., Bengio Y., Courville A., Deep Learning, MIT Press (2016).
2. Patterson J. and Gibson A., Deep Learning: A Practitioner's Approach, O'Reilly (2017).

Reference Books:

1. Haykin S., Neural Networks and Learning Machines, Pearson (2009), 3rd Ed.
2. Géron A., Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow,O'Reilly (2019), 2nd Ed.

UAI402: Foundation of Data Engineering

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course aims to explore the fundamentals and applications of data engineering, with a consideration of sustainability. Students will acquire the ability to minimize their environmental impact by designing, constructing, and maintaining data systems. The subjects covered include data lifecycle management, green computation, data pipelines, and storage optimization.

Foundations of Data Engineering and Data Modeling: Introduction to Data Engineering: Roles, workflows, and modern data stack, Data types, file formats (CSV, JSON, Avro, Parquet), Batch vs. streaming data, Data modeling: ER diagrams, normalization, star/snowflake schema, Introduction to data warehousing and OLAP, Use case: data modelling for 1. student performance, 2. school access in rural areas [4]

ETL/ELT Pipelines and Workflow Orchestration: Designing ETL/ELT pipelines, Data ingestion from APIs, sensors, and flat files, Data transformation and cleansing techniques, Workflow orchestration: Apache Airflow, Prefect, Error handling, retries, and monitoring, Use case: Ingesting and processing sensor data from water quality monitoring stations for real-time alerts and long-term reporting [6]

Distributed Data Processing and Big Data Frameworks: Introduction to distributed systems and Apache Hadoop, Apache Spark: RDDs, DataFrames, Spark SQL, Partitioning, shuffling, and performance optimization, Batch vs. stream processing fundamentals, Use case: Analyzing large-scale satellite and weather data to track deforestation and CO_2 emissions over time using Spark[13]

Real-Time Data Streaming and Messaging Systems: Introduction to message brokers and stream processing, Apache Kafka: architecture and core concepts, Stream processing with Spark Streaming / Apache Flink, Event time, windows, watermarks, Use case: Real-time energy consumption tracking from IoT meters to optimize grid load and promote renewable energy usage [7]

Data Governance, Cloud Deployment, and DevOps: Data quality and validation: expectations and checks, Metadata, lineage, and cataloguing, Security, encryption, and compliance (GDPR, HIPAA), Cloud platforms: AWS/GCP/Azure basics, Docker, Terraform, CI/CD pipelines for data workflows, Use case: Deploying a data platform in the cloud to monitor and visualize real-time traffic and pollution levels in urban areas [11]

Laboratory work:

- Creating and monitoring ETL pipelines
- Ingesting real-time sensor data using Kafka

- Running distributed processing jobs in Apache Spark
- Using Airflow for orchestration and scheduling
- Validating data quality and deploying pipelines on cloud infrastructure

Course Outcomes (COs)/Course Learning Outcomes (CLOs):

After completion of this course, the students will be able to:

1. Understand and design scalable data models and workflows for structured and unstructured data.
2. Build and manage ETL/ELT pipelines using modern tools and automation frameworks.
3. Apply distributed and real-time processing techniques to solve large-scale data problems.
4. Develop data engineering solutions for real-world challenges, including sustainability, governance, and cloud deployment.

Text Books:

1. Andreas Kretz, The Data Engineering Cookbook, 2022
2. Ananthakrishnan G., Data Engineering with Python, Packt Publishing, 2020

Reference Books:

1. Jesse Anderson, Data Engineering Teams, Manning, 2020
2. Jeffrey Aven, Data Lake Architecture, O'Reilly Media, 2018
3. Bill Inmon, Building the Data Warehouse, Wiley, 4th ed.
4. Valliappa Lakshmanan et al., Data Engineering on Google Cloud Platform, O'Reilly, 2022

SEMESTER VI

UCS701: THEORY OF COMPUTATION

L	T	P	Cr
3	1	0	3.5

Course Objectives: This course introduces basic theory of computer science and formal methods of computation. The course exposes students to the computability theory, as well as to the complexity theory.

Regular Languages: Alphabets, Language, Regular Expression, Definitions of Finite State Machine, Transition Graphs, Deterministic & Non-deterministic Finite State Machines, Regular Grammar, Thompson‘s Construction to Convert Regular Expression to NDFA & Subset Algorithm to convert NDFA to DFA, Various recent development in the Conversion of Regular Expression to NFA, Minimization of DFA, Finite State Machine with output- Moore machine and Melay Machine, Conversion of Moore machine to Melay Machine & Vice-Versa.

Properties of Regular languages: Conversion of DFA to Regular Expression, Pumping Lemma, Properties and Limitations of Finite state machine, Decision properties of Regular Languages, Application of Finite Automata.

Context Free Grammar and Push Down Automata: Context Free Grammar, Derivation tree and Ambiguity, Application of Context free Grammars, Chomsky and Greibach Normal form, Properties of context free grammar, CKY Algorithm, Decidable properties of Context free Grammar, Pumping Lemma for Context free grammar, Push down Stack Machine, Design of Deterministic and Non-deterministic Push-down stack.

Turing Machine: Turing machine definition and design of Turing Machine, Variations of Turing Machines, combining Turing machine, Universal Turing Machine, Post Machine, Chomsky Hierarchy, Post correspondence problem, Halting problem, Turing decidability.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Comprehend regular languages and finite automata and develop ability to provide the equivalence between regular expressions, NFAs, and DFAs.
2. Disambiguate context-free grammars and understand the concepts of context-free languages and pushdown automata.
3. Analyse and design efficient Turing Machines.
4. Distinguish different computing languages and classify their respective types.

Text Books:

1. Hopcroft E. J., Ullman D. J. and Motwani R., Introduction to Automata Theory, Languages and Computation, Pearson Education (2007) 3rd ed.
2. Martin C. J., Introduction to Languages and the Theory of Computation, McGraw-Hill Higher Education (2011) 4th ed.

3. Lewis R. H., Papadimitriou H. C., Elements of the Theory of Computation, Prentice Hall (1998) 2nd ed.

Reference Books:

1. Cohen A. I. D., Introduction to Computer Theory, Wiley (1997) 2nd ed.
2. Sipser M., Introduction to the Theory of Computation, Cengage Learning (2013) 3rd ed.

UAI601: Accelerated Computing

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course will provide students with fundamental knowledge of GPU Computing and Machine Learning and Deep Learning Primer.

Introduction: GPU Computing, Parallel Programming, Machine Learning with RAPIDS, Model Compression

GPU Acceleration: Introduction to GPU Architecture, CUDA Programming Basics, Accelerating Applications with Parallel Algorithms using C++, Harnessing Asynchrony with CUDA Streams**,** Implementing New Algorithms with CUDA Kernels**,** Introduction to CUDA Python with Numba, Custom CUDA Kernels in Python with Numba, Multidimensional Grids, and Shared Memory for CUDA Python with Numba, Implementing pipelines with NVIDIA libraries (cuDNN, NPP)

Image Processing and Parallel Programming: OpenCL for Image Processing, Parallel image filtering, GPU-accelerated convolution & FFT, nvJPEG, Halide-High-Performance Image Pipelines, High-Speed Video Processing.

Introduction to Accelerated Machine Learning: Introduction to Supervised and Unsupervised Learning with Rapids Acceleration.

Deep Learning Model Compression: Introduction to model pruning, quantization and distillation

Optimization Framework: Using TensorRT optimization, Deploying model on Triton Inference server

Laboratory Work:

- Practical on Traditional Data Science packages (Numpy, Pandas, Scipy, Scikit-Learn).
- CUDA C/C++ for Accelerated Computing.
  {DLI Online Course Section: Fundamentals of Accelerated Computing with CUDA C/C++}
- Numba to compile CUDA kernels for Numpy Acceleration in Python.
  {DLI Online Course Section: Fundamentals of Accelerated Computing with CUDA Python}
- Getting started with Accelerated Data Science with RAPIDS AI (cuPy, cuDF, cuSignal, cuML).
- Decision Tree Classification Clustering in RAPIDS.
- Random Forest Classification in RAPIDS.
  {DLI Online Course Section: Fundamentals of Accelerated Data Science with RAPIDS, Section 2: GPU-accelerated Machine Learning}
  Model Pruning, Post Training Quantization, Quantization Aware Training, Distillation

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course the student will be able to:

1. Demonstrate ability to deploy GPU accelerated code for image processing applications .
2. Apply model compression techniques while deploying deep learning architectures. .
3. Implement Rapids AI framework for different machine learning tasks. .
4. Analyze and evaluate performance of deep learning based inference models using TensorRT optimization and Triton Server .

Text Books:

1. Vaidya, Bhaumik. *Hands-On GPU-Accelerated Computer Vision with OpenCV and CUDA.* Packt Publishing, 2018.
2. Comer, M., et al. *Real-Time Image and Video Processing: From Research to Reality.* Morgan & Claypool Publishers, 2008.
3. Paulo Motta. *GPU Programming with C++ and CUDA: Uncover effective techniques for writing efficient GPU-parallel C++ applications*. Packt Publishing, 29 August 2025. ISBN 978-1805124542

Reference Books:

1. William Smith. *CUDA Programming with Python: From Basics to Expert Proficiency.* HiTeX Press, August 4, 2024.

UTA025: Innovation and Entrepreneurship (2 Self-Efforts Hours)

L	T	P	Cr
1	0	2	3.0

Course Objectives: This course aims to provide the students with a basic understanding in the field of entrepreneurship, entrepreneurial perspectives, concepts and frameworks useful for analyzing entrepreneurial opportunities, understanding eco-system stakeholders and comprehending entrepreneurial decision making. It also intends to build competence with respect business model canvas and build understanding with respect to the domain of start-up venture finance.

Introduction to Entrepreneurship: Entrepreneurs; entrepreneurial personality and intentions - characteristics, traits and behavioral; entrepreneurial challenges.

Entrepreneurial Opportunities: Opportunities- discovery/ creation, Pattern identification and recognition for venture creation: prototype and exemplar model, reverse engineering.

Entrepreneurial Process and Decision Making: Entrepreneurial ecosystem, Ideation, development and exploitation of opportunities; Negotiation, decision making process and approaches, - Effectuation and Causation.

Crafting business models and Lean Start-ups: Introduction to business models; Creating value propositions - conventional industry logic, value innovation logic; customer focused innovation; building and analyzing business models; Business model canvas, Introduction to lean startups, Business Pitching.

Organizing Business and Entrepreneurial Finance: Forms of business organizations; organizational structures; Evolution of organization, sources and selection of venture finance options and its managerial implications. Policy Initiatives and focus; role of institutions in promoting entrepreneurship.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

Upon successful completion of the course, the students should be able to:

1. Explain the fundamentals behind the entrepreneurial personality and their intentions
2. Discover/create and evaluate opportunities.
3. Identify various stakeholders for the idea and develop value proposition for the same.
4. Describe various Business Models and design a business model canvas.
5. Analyse and select suitable finance and revenue models for start-up venture.

Text Books:

Ries, Eric (2011), The lean Start-up: How constant innovation creates radically successful businesses, Penguin Books Limited.

Blank, Steve (2013), The Startup Owner‘s Manual: The Step by Step Guide for Building a Great Company, K&S Ranch.

S. Carter and D. Jones-Evans, Enterprise and small business- Principal Practice and Policy, Pearson Education (2006)

Reference Books:

1. T. H. Byers, R. C. Dorf, A. Nelson, Technology Ventures: From Idea to Enterprise, McGraw Hill (2013)
2. Osterwalder, Alex and Pigneur, Yves (2010) Business Model Generation.
3. Kachru, Upendra, India Land of a Billion Entrepreneurs, Pearson
4. Bagchi, Subroto, (2008), Go Kiss the World: Life Lessons For the Young Professional, Portfolio Penguin
5. Bagchi, Subroto, (2012). MBA At 16: A Teenager‘s Guide to Business, Penguin Books
6. Bansal, Rashmi, Stay Hungry Stay Foolish, CIIE, IIM Ahmedabad
7. Bansal, Rashmi, (2013). Follow Every Rainbow, Westland.
8. Mitra, Sramana (2008), Entrepreneur Journeys (Volume 1), Booksurge Publishing
9. Abrams, R. (2006). Six-week Start-up, Prentice-Hall of India.
10. Verstraete, T. and Laffitte, E.J. (2011). A Business Model of Entrepreneurship, Edward Elgar Publishing.
11. Johnson, Steven (2011). Where Good Ideas comes from, Penguin Books Limited.
12. Gabor, Michael E. (2013), Awakening the Entrepreneur Within, Primento.
13. Guillebeau, Chris (2012), The $100 startup: Fire your Boss, Do what you love and work better to live more, Pan Macmillan
14. Kelley, Tom (2011), The ten faces of innovation, Currency Doubleday
15. Prasad, Rohit (2013), Start-up sutra: what the angels won‘t tell you about business and life, Hachette India.

UMA071: NUMERICAL OPTIMIZATION

L	T	P	Cr
3	0	2	4.0

Course Objectives: The objective of this course is to provide students with a comprehensive understanding of optimization techniques, both linear and nonlinear, and their applications in various fields. Students will gain knowledge of fundamental concepts, methods, and algorithms in optimization and develop the skills necessary to formulate and solve optimization problems. Through this course, students will learn to apply optimization models and techniques to realworld problems, enhancing their analytical and problem-solving abilities.

Unit 1: Introduction: Definition of optimization, types of optimization models (linear vs nonlinear, constraint vs unconstraint optimization models), geometry of linear programming

Unit 2: Linear optimization: Basic Feasible Solutions and Algebraic Method, Simplex method, its variants, and special cases, duality for Linear Programming Problems (LPP), branch and bound method for Integer Linear Programming Problems (ILPP)

Unit 3: Unconstrained Optimization: Definiteness of matrices, optimality conditions for unconstrained optimization, special case of quadratic functions, convex functions and their properties, applications of quadratic functions to least square approximation

Unit 4: Search Techniques: Steepest descent method, conjugate gradient method, Newton and Quasi-Newton methods, Fletcher and Reeve's method, David-Fletcher-Powell method

Unit 5: Constrained Optimization, Convex optimization, KKT conditions for general inequality and equality constrained problems, Frank and Wolfe's method, Rosen gradient projection method, Penalty and barrier functions, Convex quadratic problem and its application to classification by linear separators, convex optimization in support vector machines(SVM).

Course Outcomes(COs):

Upon Completion of this course, the students would be able to:

1. Formulate classify optimization problems.
2. Apply the Simplex Method and its variants for solving the Linear programming problems, and analyse the Integer Programming Problems.
3. Implement Search Techniques in Unconstrained Optimization.
4. Evaluate Constrained Optimization Problems.
5. Demonstrate Proficiency in Using Software Tools such as MATLAB or other relevant software to implement optimization algorithms and solve problems computationally.

Text Books:

1. Chandra, S., Jyadeva, Mehra, A. Numerical Optimization. Narosa, 2009.
2. Bazaraa, M.S., Shefali, H.D., Shetty, C.M. Nonlinear Programming: Theory and Algorithms. New York: John Wiley & Sons, 1993. ISBN: 0471557935
3. Beck,A.Introduction to Nonlinear Optimization Theory, Algorithms, and Applications with MATLAB. SIAM Series on Optimization.

Reference Books:

1. Cottle, R.W., Thapa, M.N. Linear and Nonlinear Optimization. Springer, 2017.
2. 2. Luenberger, D.G., Ye, Y. Linear and Nonlinear Programming, 5th Edition. Springer, 2021.
3. Bertsekas, D. Nonlinear Programming. Athena Scientific, 2016.
4. Boyd, S., Vandenberghe, L. Convex Optimization. Cambridge: Cambridge University Press, 2004.

EFB

High Performance Computing

UCS531: CLOUD COMPUTING

L	T	P	Cr
2	0	2	3.0

Introduction and Evolution of Computing Paradigms: General Benefits and Architecture, Business Drivers, Main players in the Field, Overview of Existing Hosting Platforms and its architecture, Cluster Computing, Grid Computing, XaaS Cloud Based Service Offerings, Overview of Security Issues, The impact of cloud computing on conventional and distance education systems.

Classification of Cloud Implementations: Key Amazon offerings-Amazon Web Services, The Elastic Compute Cloud (EC2), Simple Storage Service (S3), Simple Queuing Services (SQS), Bundling Amazon instances, AWS Identity Management and Security in the Cloud, Messaging in the Cloud, RESTFul Web Services.

Virtualization: Virtualization, Advantages and disadvantages of Virtualization, Types of Virtualization: Resource Virtualization i.e. Server, Storage and Network virtualization, Migration of processes, Classic Data Center, Virtualized Data Center (Compute, Storage, Networking and Application), Business Continuity in VDC. VMware vCloud – IaaS, Network virtualization through Software Defined Networks, data centre energy consumption; Renewable energy in powering data centres; Green data centres design, cooling, and energy management; Virtualization and server consolidation;

Cloud based Data Storage: Introduction to Hadoop, Hadoop Ecosystem (Pig, Hive, Cassandra and Spark), Introduction No-SQL databases, Map- Reduce framework for Simplified data processing on Large clusters using Hadoop, Data Replication, Shared access to data stores.

Related Technologies: Introduction to Fog Computing and Edge Computing, Usage of Cloud for IoT and Big data analytics, Overview of Google AppEngine - PaaS, Windows Azure, Role of Cloud, edge and fog computing in smart healthcare.

Sustainability: cloud sustainability and its importance, Global regulations and standards related to cloud sustainability, Carbon footprint measurement and reduction in cloud computing, green software principles to develop energy-efficient applications

Self-learning Content:

Cloud Issues and Challenges: Cloud models, Cloud computing issues and challenges like Security, Elasticity, Resource management and Scheduling, QoS (Quality of Service) and Resource Allocation, Cost Management and Cloud bursting.

Laboratory work:

To implement Cloud, Apache and basics of Hadoop framework, an open source implementation of MapReduce, and its Java API, Hadoop Distributed File System (HDFS). Implementation of RESTFul Web Services. To understand various concepts about

virtualization and data storage. To implement few algorithms with the help of MapReduce and some high-level language.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Comprehend the basic concepts and architecture of Cloud computing.
2. Implement Cloud Services through AWS offerings and Restful web services.
3. Apply the knowledge of virtualization through different virtualization technologies.
4. Perform operations on data sets using Map Reduce framework, SQL and NO SQL databases.

Text Books:

1. Buyya K, R., Broberg J. and Goscinski M. A., Cloud Computing: Principles and paradigms, MIT Press (2011) 4th ed.
2. Kai Hwang, Geoffrey Fox and Jack Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann (2012) 2nd ed.
3. Miller M., Cloud Computing, Que Publishing (2008) 1st ed.
4. Puttini R. and Mahmood Z., Cloud Computing: Concepts, Technology & Architecture, Service Tech press (2013) 1st ed.

Reference Books:

1. Velte A., Velte T., and Elsenpeter R., Cloud Computing: A practical Approach, Tata McGrawHill (2009) 1st ed.
2. Hurwitz J., Bllor R., Kaufman M. and Halper F., Cloud Computing for dummies (2009) 1st ed.

UCS635: GPU COMPUTING

L	T	P	Cr
2	0	2	3.0

Introduction: Heterogeneous Parallel Computing, Architecture of a Modern GPU, Power-efficient GPU architectures: low power states, dynamic voltage and frequency scaling, Speeding Up Real Applications, Parallel Programming Languages and Models, Energy consumption in data centers and the role of GPUs.

History of GPU Computing: Evolution of Graphics Pipelines, The Era of Fixed-Function Graphics Pipelines, Evolution of Programmable Real-Time Graphics, Unified Graphics and Computing Processors, GPGPU, Scalable GPUs, Recent Developments, Future Trends.

Introduction to Data Parallelism and CUDA C: Data Parallelism, CUDA Program Structure, A Vector Addition Kernel, Device Global Memory and Data Transfer, Kernel Functions and Threading, Parallelization techniques for optimal GPU utilization.

Data-Parallel Execution Model: CUDA Thread Organization, Mapping Threads to Multidimensional Data, Matrix-Matrix Multiplication—A More Complex Kernel, Synchronization and Transparent Scalability, Assigning Resources to Blocks, Thread Scheduling and Latency Tolerance.

CUDA Memories: Importance of Memory Access Efficiency, CUDA Device Memory Types, A Tiled Matrix – À Matrix Multiplication Kernel, Memory as a Limiting Factor to Parallelism.

An Introduction to OpenCL: Data Parallelism Model, Device Architecture, Kernel Functions, Device Management and Kernel Launch, Electrostatic Potential Map in OpenCL.

Parallel Programming with OpenACC: OpenACC Versus CUDA C, Execution Model, Memory Model, Basic OpenACC Programs, Parallel Construct, Loop Construct, Kernels Construct, Data Management, Asynchronous Computation and Data Transfer.

Self-Learning Content:
Basics of Parallel and distributed Computing, CUDA programming model

Laboratory work:
Practice programs using CUDA, OpenCL and OpenACC.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

On completion of this course, the students will be able to

1. Comprehend commonly used terms in parallel computing.
2. Understand common GPU architectures and Programming Models.
3. Implement algorithms efficiently for common application kernels.
4. Develop efficient parallel algorithms to solve given problems.

Text Books:

1. Sanders, J. and Kandrot, E., CUDA by Example: An Introduction to General-PurposeGPU Programming, Addison-Wesley Professional (2012) 4th Edition.
2. Kirk, D. and Hwu, M., W., Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann (2016) 3rd Edition.
3. Grama, A., Gupta, Karypis, G., Kumar, V., Introduction to Parallel Computing, Addison Wesley, (2003) 2nd Edition.

Reference Book:

1. Hwu, M., W., A GPU Computing Gems Emerald Edition (Applications of GPU Computing Series), Morgan Kaufmann (2011) 1st Edition.

UCS645: PARALLEL & DISTRIBUTED COMPUTING

L	T	P	Cr
2	0	2	3.0

Parallelism Fundamentals: Scope and issues of parallel and distributed computing, Parallelism, Goals of parallelism, Parallelism and concurrency, Multiple simultaneous computations.

Parallel Architecture: Implicit Parallelism, Array Processor, Vector Processor, Dichotomy of Parallel Computing Platforms (Flynn's Taxonomy, UMA, NUMA, Cache Coherence), Fengs Classification, Handler Classification, Limitations of Memory System Performance, Interconnection Networks, Communication Costs in Parallel Machines , Routing Mechanisms for Interconnection Networks , Impact of Process-Processor Mapping and Mapping Techniques, GPU, energy-efficient hardware and cooling techniques for parallel systems.

Parallel Decomposition and Parallel Performance: Principles of Parallel Algorithm Design: Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing. Energy-efficient parallel algorithms, Critical Paths, Sources of Overhead in Parallel Programs, Performance metrics for parallel algorithm implementations, Performance measurement, The Effect of Granularity on Performance.

Distributed Computing: Introduction: Definition, Relation to parallel systems, synchronous vs. asynchronous execution, design issues and challenges, A Model of Distributed Computations, A Model of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication.

Programming Message Passing and Shared Address Space Platforms: Send and Receive Operations, MPI: the Message Passing Interface, Topologies and Embedding, Overlapping Communication with Computation, Groups and Communicators.

CUDA programming model: Overview of CUDA, Isolating data to be used by parallelized code, API function to allocate memory on the parallel computing device. Launching the execution of kernel function by parallel threads, transferring data back to host processor with API function call.

Parallel Algorithms design, Analysis, and Programming: Parallel Algorithms, Parallel Graph Algorithms, Green scheduling algorithms, Parallel Matrix Computations, Critical paths, work and span and relation to Amdahl's law, Speed-up and scalability, Naturally parallel algorithms, Parallel algorithmic patterns like divide and conquer, map and reduce, Specific algorithms like parallel Merge Sort, Study sustainability practices in large-scale distributed systems like Google Cloud or AWS.

Self-Learning Content:

Programming Message Passing and Shared Address Space Platforms: Thread Basics, Synchronization Primitives in Pthreads, Controlling Thread and Synchronization Attributes, Composite Synchronization Constructs, Tips for Designing Asynchronous Programs.

CUDA programming model: API function to transfer data to parallel computing device, Concepts of Threads, Blocks, Grids, developing kernel function that will be executed by threads in the parallelized part.

Parallel Algorithms design, Analysis, and Programming: Parallel graph algorithms, parallel shortest path, parallel spanning tree, Producer-consumer and pipelined algorithms.

Laboratory work:

To implement parallel programming using CUDA with emphasis on developing applications for processors with many computation cores, mapping computations to parallel hardware, efficient data structures, paradigms for efficient parallel algorithms.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

On completion of this course, the students will be able to

1. Apply the fundamentals of parallel and distributed computing including parallel architectures and paradigms.
2. Apply parallel algorithms and key technologies.
3. Develop and execute basic parallel applications using basic programming models and tools.
4. Apply shared address space and message passing in programming platforms
5. Analyze the performance issues in parallel computing and trade-offs.

Text Books:

1. C Lin, L Snyder. Principles of Parallel Programming. USA: Addison-Wesley (2008).
2. A Grama, A Gupta, G Karypis, V Kumar. Introduction to Parallel Computing, Addison Wesley (2003).

Reference Books:

1. B Gaster, L Howes, D Kaeli, P Mistry, and D Schaa. Heterogeneous Computing With Opencl. Morgan Kaufmann and Elsevier (2011).
2. T Mattson, B Sanders, B Massingill. Patterns for Parallel Programming. Addison-Wesley (2004).
3. Quinn, M. J.,Parallel Programming in C with MPI and OpenMP, McGraw-Hill(2004).

UCS751: SIMULATION & MODELLING

L	T	P	Cr
2	0	2	3.0

Introduction to Modeling and Simulation: Basic concept of Simulation, Advantages, Disadvantages, Applications of simulation, limitation of simulation, Model and types of models, modeling and simulation, Continuous and discrete simulation, analog and digital simulation, System environment, components of a system, steps in a simulation study, Simulation of Queuing and Inventory System.

Random Numbers generation: Pseudo-random generators, Testing of Pseudo-random number generators, Generation of non-uniformly distributed random numbers.

Parallel process modeling: Using Petri nets and finite automata in simulation, Cellular automata and simulation.

Simulation Experiments: Run length of Static and Dynamic Stochastic Simulation Experiments, Minimizing variability in simulators without increasing Number of simulation Runs.

Design of Simulators: Design of Application Simulators for Multi-server Queuing System, PERT, Optimizing Inventory Policy and Cost in Business environment.

Input Modeling: Data collection, Identification and distribution with data, parameter estimation, Goodness of fit tests, Selection of input models without data, Multivariate and time series analysis. Verification and Validation of Model: Model Building, Verification, Calibration and Validation of Models.

Output Analysis: Types of Simulations with Respect to Output Analysis, Stochastic Nature of output data, Measures of Performance and their estimation, Output analysis of terminating simulation, Output analysis of steady state simulations.

Sustainability: Role of simulation in environmental sustainability (e.g., climate modeling, resource optimization), Sustainable practices in education, healthcare, and urban planning through simulation. Ethical considerations and long-term impact analysis, Energy-efficient simulation techniques, Reducing carbon footprint through cloud-based simulation, Eco-friendly computational models and their significance.

Laboratory Work:
To carry out work on any simulation tools, Implementation of various techniques to generate random numbers. Apply any simulation model in real life applications.

Self-Learning Content:
Different Simulation Softwares and their applications for different analysis, Trends in Simulation Software.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Describe the role of various elements of discrete event simulation and modeling paradigm.
2. Conceptualize real world situations related to systems development decisions, originating from source requirements and goals.
3. Generate and test random number variates and apply them to develop simulation models.
4. Interpret the model and apply the results to resolve critical issues in a real-world environment.
5. Classify various simulation models and their usage in real-life applications.

Text Books:

1. Payne A. J., Introduction to Simulation: Programming Techniques and Methods of Analysis, McGraw Hill (1982).
2. Gorden G., System Simulation, Prentice Hall publication (1978), 2nd ed.

Reference Books:

1. Narsingh D., Systems Simulation with Digital Computer, PHI Publication (EEE) (2004), 3rd ed.
2. Banks J., Carson J. S., Nelson L. B., Nicol M. D, Discrete Event system Simulation, Pearson Education, Asia (2010), 5th ed.

EFB

Computer Animation and Gaming

UCS532: COMPUTER VISION

L	T	P	Cr
2	0	2	3.0

Course Objectives: To understand the basic concepts of Computer Vision. The student must be able to apply the various concepts of Computer Vision in other application areas.

Digital Image Formation and Sustainable Image Preprocessing Techniques: Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.

Image Representation & Description: Optimizing algorithms for low-energy consumption in edge devices - Canny, LOG, DOG; Lightweight Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, LBP and its variants, Gabor Filters and DWT.

Image Segmentation: Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Lightweight and energy-efficient object detection models (e.g., YOLO Nano, MobileNet SSD) .

Pattern Analysis: Clustering: K-Means, Fuzzy C-means; Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Reducing data dimensionality to lower computational costs and storage requirements : PCA, LDA, ICA.

Motion Analysis: Background Subtraction and Modeling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

Self-Learning Content:
Miscellaneous: Applications: CBIR, CBVR, Activity Recognition, computational photography, Biometrics, stitching and document processing; Modern trends - super-resolution; GPU, Augmented Reality; cognitive models, fusion and SR&CS.

Laboratory Work:
To implement various techniques and algorithms studied during course.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:

1. Understand the fundamental problems of computer vision.
2. Implement various techniques and algorithms used in computer vision.
3. Analyse and evaluate critically the building and integration of computer vision algorithms and systems.
4. Demonstrate awareness of the current key research issues in computer vision.

Text Books:

1. Szeliski, R., Computer Vision: Algorithms and Applications, Springer-Verlag London Limited (2011), 1st Edition.
2. Forsyth, A., D. and Ponce, J., Computer Vision: A Modern Approach, Pearson Education (2012) 2nd Edition.

Reference Books:

1. Hartley, R. and Zisserman, A., Multiple View Geometry in Computer Vision Cambridge University Press (2003) 2nd Edition.
2. Fukunaga, K., Introduction to Statistical Pattern Recognition, Academic Press, Morgan Kaufmann (1990) 2nd Edition.
3. Gonzalez, C., R. and Woods, E., R. Digital Image Processing, Addison- Wesley (2018) 4th Edition.

UCS636: 3D MODELLING AND ANIMATION

L	T	P	Cr
2	0	2	3.0

Course Objectives: To develop the skill & knowledge in 3D Modeling & Animation. Students will understand the know-how and can function either as an entrepreneur or can take up jobs in the multimedia and animation industry, video studios, edit set-up and other special effects sectors.

3D Object Modelling: Basic modelling concepts, vertices, edges, and faces, basic transformations, pivot points, duplication and merging, extrusion, insetting, modifiers, loop cuts and face loops, subdivision methods, coordinate system and exporting, Energy-Efficient rendering model .

Low Poly Models: Triangular meshes, objects and mesh data, cursor and origins hidden geometry, Boolean modifiers, geometry from curve, curve resolution, non-planner geometry.

3D Character Modelling: Introduction, character modelling, unwrapping UVs & mapping texture, Low-Impact texturing painting, armatures, Eco-Themed Characters rigging , constrained movements, forward and inverse kinematics, time-line, keyframes, character animation, Optimized Rendering Pipelines .

Physically Based Modelling and Animation: Introduction, Energy-Efficient Simulations Foundation , Particle based Models, Collision detection and response, Particle System, Particle Simulation, Particle Rendering, Numerical Integration in Particle System, Deformable Meshes, Rigid Bodies and Constrained Dynamics, Fluid Simulation.

Self-Learning Content: Real Time Animation: Splines and curves, Key-frame techniques, Quaternions for rotations / orientations, Blending and interpolation, Kinematics, Motion capture systems, Motion graphs and character control, Animation data representations, Behavioural Animation, Facial Animation, Perception in animation.

Laboratory Work

This course covers beginner to intermediate 3D Modeling and Animation. In this Lab the students will be able to model the 3D character and objects, its UV Mapping, Texture Painting, Rigging, and Animation. Evaluation will be mainly via projects and assignments taking a creative approach to expressive 3D modelling and Animation.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Apply modelling concepts in order to implement 3D objects. (Blender / Max).
2. Understand the basic geometry and triangulation techniques behind low poly models.
3. Implement 3D humanoid characters and to apply the concept of rigging for animating the character using key frames.
4. Illustrate the theoretical and practical aspects of 3D Modelling, Key Frame Animation, Simulation & effects.
5. Demonstrate different types of animation and its effects in the real world.
6. Analyse the different processes, post processes involved in computer animation field.

Text Books:

1. House, H., D. and Keyser, C., J., Foundations of Physically Based Modeling and Animation, CRC Press (2017) 1st Edition.
2. Chopine, A., 3D Art Essentials: The Fundamentals of 3D Modeling, Texturing, and Animation, Focal Press (2011) 1st Edition.
3. Zeman, B., N., Essential Skills for 3D Modeling, Rendering, and Animation, A K Peters / CRC Press (2017) 1st Edition.

Reference Books:

1. Villar, O., Learning Blender: A Hands-On Guide to Creating 3D Animated Characters, Addison Wesley (2017) 2nd Edition.
2. Kerlow, I., The Art of 3D Computer Animation and Effects, Wiley, (2009) 4th Edition.
3. Flavell, L., Beginning Blender: Open Source 3D Modelling, Animation, and Game Design, Apress, (2010) 1st Edition.
4. Boardman, T., 3dsmax 7 Fundamentals, New Riders, (2005) 1st Edition.

UCS646: GAME DESIGN & DEVELOPMENT

L	T	P	Cr
2	0	2	3.0

Course Objectives: To become familiar with various fundamental and advanced gaming concepts including basic maths and physics used behind the game engine.

Introduction: Types of games, History, Impact of Games on Society , Game life cycle, Game loop, Components of game**,** Model and scene rendering, State Management, Scene management, Texture compression, Level of details, Frustum culling, Occlusion culling, Game as a software, Steps for Game Design, Data Structure for Game, CPU vs.GPU, Game Engine, Components of game engine, Linear Transformation. Composite transformation.

Fundamental Gaming concepts: Static and Dynamic Game objects, Vectors, Concept of Time, Lighting , Particle System, Collider, Collision handles, Materials**,** Texture mapping, Input Process, Object replication, Instantiation, Special Effects, Terrain, Audio design and production, Ray Casting.

Maths behind Game Engines: Introduction to Vectors- Addition & Subtraction, Vector length, Scaling, Unit length vectors, Dot & Cross product, Linear Interpolation, Euler Angles, Intersection, Matrices, Coordinate systems, Projections, Triangle Meshes, Optimizations, Quaternion.

Advanced Games: Augmented Reality, Virtual Reality, Mixed Reality, AR & VR based Games, Artificial Intelligence based Game, Networking based game, Android based games, Debugging mode, Understanding of Screen and World Coordinate system, Raycasting, Touch & Swipe Input: Touch in Orthographic view, Touch in Perspective view, Accelerometer input, Scaling of Game screen, AR/VR/Android/iOS/Windows Game Deployment methods, Integrating game mechanics into e-learning tools and fitness-focused games that encourage physical activity

Self-Learning Content: Game Physics: Mathematical concepts, Basic transformations, Collision Detection and response, Newton's law of motion, Modeling gravity, Air resistance, Unstable rotation, Inertia tensor, Moment of Inertia, Applying torque to rigid body, The Magnus effect, Overview of friction, Critical angle, Dynamic Friction.

Laboratory work: 2D and 3D game development for windows and android platform using Unity 3D Game Engine and C# language.

Course learning outcomes (CLOs):

On completion of this course, the students will be able to

1. Illustrate the basic concepts, requirements and processes of game design and development
2. Implement the fundamental gaming concepts to create a game.
3. Understand the physics and mathematics behind the game engine.
4. Demonstrate the advanced gaming concepts such as AR, VR, Android etc.
5. Develop a 2D/3D game using C# and Unity 3D Game engine.

Text Books:

1. *Eberly H. D., Game Physics ,Morgan Kaufmann Publisher (2010), 2nd ed.*
2. *Bond G. J., Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C# , Addison-Wesley (2015), 2nd ed.*

Reference Books:

1. *House H. D., Keyser C. J, Foundations of Physically Based Modeling and Animation, CRC Press (2017), 1st ed.*
2. *Okita. A., Learning C# Programming with Unity 3D, CRC Press (2014), 1st Ed.*

UCS752: AUGMENTED AND VIRTUAL REALITY

L	T	P	Cr
2	0	2	3.0

Course Objectives: To become familiar with the concept and applications of augmented & virtual reality and learn different types of algorithmic techniques and strategies.

Introduction of Augmented Reality (AR): Definition and Applications, History, Types of AR, Suitable devices, Holograms, Mixed reality, Ubiquitous computing, AR Displays: Method of Augmentation, Spatial Display Model.

Tracking in AR: Basic steps of AR, Tracking, Occlusion, Calibration, Registration, Co-ordinate Systems: Model-View-Projective Transformation, Frame of reference, Characteristics of Tracking Technology: Physical Phenomenon, Triangulation, Trilateration, Measurement Principles, Degree of Freedom, Stationary Tracking System, Mobile Tracking, Optical Tracking, Sensor Fusion.

Computer Vision for AR: Marker Tracking, Thresholding, Contour detection, Hough Transformation, Quadrilateral fitting, SIFT, Pose Estimation, Homography, Incremental Tracking, SLAM: Bundle Adjustment, Parallel Tracking and Mapping, Outdoor Tracking, STML.

Virtual Reality: Definition, History, Application, Types of VR, Components of VR,VR- HMDs and their working, Geometric modeling, Modeling Transformation, Viewing transformation Chain and Rendering Pipeline, Light and Optical System, Rendering Problems in VR, Shading Models, Rasterization, Depth, Motion and Auditory Perception, Rendering, Post Rendering Image Warping, **AR/VR for interactive and immersive applications in STEM**

Self-Learning Content: Calibration and Registration: Camera Representation , Camera Calibration, Display Calibration, Registration, Visual Coherence, Photometric Registration, Common Illumination, Diminished Reality, Camera Simulation, Stylized Augmented Reality.

Laboratory work: To implement various techniques studied during course.

Course learning outcomes (CLOs):
On completion of this course, the students will be able to

1. Analyze the components of AR systems, its current and upcoming trends, types, platforms, and devices.
2. Understand the basic steps and technologies required to achieve AR system.
3. Apply various well-known computer vision algorithms in order to implement the AR.
4. Understand the various components, applications, latest devices and working model of VR systems.
5. Develop interactive augmented and virtual reality applications for PC and Mobile based devices using a variety of input devices.

Text Books:

1. *Dieter Schmalstieg, Tobias Höllerer, Augmented-Reality-Principles-and-Practice-Usability- , Addison-Wesley (2016) 1st ed.*
2. *Parisi T., Learning Virtual Reality, O'Reilly (2016)1st ed.*
3. *Gerard Jounghyun Kim, Designing Virtual Reality Systems :The Structured Approach, Springer (2005) 1st ed.*

Reference Books:

1. *Whyte J., Virtual Reality and the Built Environment, Architectural Press (2002).*
2. *Aukstakalnis S., Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR, Addison-Wesley (2016).*

EFB

Informatio n and Cyber Security

UCS534: COMPUTER & NETWORK SECURITY

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course is designed to impart a critical theoretical and detailed practical knowledge of a range of computer network security technologies as well as network security tools.

Introduction: Security Attacks, Security Services, Security Mechanisms and Principles, Security goals, Malicious software, Worms, Viruses, Trojans, Spyware, Botnets, Life cycle of a vulnerability: CAN and CVE, Environmental impact of network security infrastructure, e-waste, and lifecycle analysis.

Computer Security: Set-UID programs, privileged programs, environment variables: hidden inputs, capability leaking, invoking other programs, principle of least privileges. Environment variables and attacks, attacks via dynamic linker, external program and library. Shellshock attack, exploiting shellshock vulnerability. Buffer overflow attacks: program memory layout, stack and function invocation. Writing a shell code, injecting code into buffer, address space layout randomization, Stack Guard.

Network Security: Packet sniffing and spoofing, Attacks on TCP protocol, SYN flood, TCP reset attack, session hijacking attack, Firewalls: Packet filter, Stateful firewall, Application firewall. IP tables, DNS poisoning, Authoritative replies, ARP poisoning, Heartbleed Bug and Attack, Public key infrastructure and Transport Layer Security, Optimizing network security for energy-efficient data transmission.

Laboratory work:
Demonstrate use of Environment variables and privileged programs, Implementing secure network protocols with resource optimization, Demonstrate Buffer Overflow and showcase EIP and other register status, insert malicious shell code into a program file and check its malicious or benign status, perform ARP poisoning, implement stateful firewall using IPTables.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:

1. Identify software vulnerabilities and apply various security mechanisms to protect against security attacks.
2. Demonstrate shellshock attack and its countermeasure.
3. Demonstrate buffer-overflow attack, locate and fix security leaks in a computer software.
4. Implement firewall and its variants.
5. Implement PKI and TLS.

Text Books:

1. Stallings, W., Network Security Essentials, Prentice Hall (2017) 6th Edition.
2. Cheswick, R., W., Bellovin, M., S., and Rubin, D., A., Firewalls and Internet Security, Addison-Wesley Professional (2003) 2nd Edition.
3. Wenliang Du, Computer Security: A hands-on approach, CreateSpace (2017).

Reference Books:

1. Graves, K., Certified Ethical Hacking Study Guide,Sybex (2010) 1st Edition.

2. Stallings, W., Cryptography and Network Security, Prentice Hall (2013), 6th Edition.

UCS638: SECURE CODING

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course aims to provide an understanding of the various security attacks and knowledge to recognize and remove common coding errors that lead to vulnerabilities. It gives an outline of the techniques for developing a secure application.

Introduction: Security, CIA Triad, Viruses, Trojans, and Worms, Security Concepts- exploit, threat, vulnerability, risk, attack, Rootkits, Trapdoors, Botnets, Key loggers, Honeypots. Active and Passive Security Attacks.

Need for secure systems: Proactive Security development process, Secure Software Development Cycle (SSDLC), Security issues while writing SRS, Design phase security, Development Phase, Test Phase, Maintenance Phase, Writing Secure Code – Best Practices SD3 (Secure by design, default and deployment), Security principles and Secure Product Development Timeline, Case studies on secure coding practices for environmental monitoring systems.

Threat modelling process and its benefits: Identifying the Threats by Using Attack Trees and rating threats using DREAD, Risk Mitigation Techniques and Security Best Practices. Security techniques, authentication, authorization. Defense in Depth and Principle of Least Privilege, Threat modeling in smart healthcare and education systems.

Software & Web Security: Return-to-libc attack, format string vulnerability. Race condition vulnerability, Dirty COW, PE Code injection. Cross site request forgery: CSRF attacks on HTTP GET and POST services & countermeasures. XSS attack: self-propagating XSS worm, preventing XSS attacks, SQL injection attack & countermeasures. Client-side attacks, web practices to ensure safe e-commerce platforms

Laboratory Work:
In this Lab, student shall learn to recognize and remove common coding errors that lead to vulnerabilities. This lab also gives an outline of the techniques for developing a secure application code, implementing different types of attacks and protection schemes for both software and web security. Evaluation will be mainly based on projects and assignments.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:

1. Demonstrate skills needed to deal with common programming errors and develop secure applications.
2. Implement PE Code injection and demonstrate control hijacking via EIP manipulation
3. Demonstrate client-side attacks and identify nature of threats to software and incorporate secure coding practices throughout the planning and development of software product.
4. Demonstrate SQL injection, XSS attack and suggest countermeasures for the same.

Text Books:

1. Howard, M. and LeBlanc, D., Writing Secure Code, Howard, Microsoft Press (2002) 2nd Edition.

2. Deckard, J., Buffer Overflow Attacks: Detect, Exploit, Syngress (2005) 1st Edition.
3. Wenliang Du, Computer Security: A hands-on approach, CreateSpace (2017).

Reference Books:

1. Swiderski, F. and Snyder, W., Threat Modeling, Microsoft Professional, (2004) 1st Edition.
2. Salt, C., J., SQL Injection Attacks and Defence, Elsevier (2012), 2nd Edition.

UCS648: CYBER FORENSICS

L	T	P	Cr
2	0	2	3.0

Course Objectives: To maintain an appropriate level of awareness, knowledge and skill required to understand and recreate the criminal terminology and Cyber Forensics investigation process.

Introduction to Cybercrime: Defining Cybercrime, Understanding the Importance of Jurisdictional Issues, Quantifying Cybercrime, Differentiating Crimes That Use the Net from Crimes That Depend on the Net, working toward a Standard Definition of Cybercrime, Categorizing Cybercrime, Developing Categories of Cybercrimes, Prioritizing Cybercrime Enforcement, Reasons for Cybercrimes, Role of Cybercrime in Hindering Sustainable Development.

Understanding the People on the Scene: Understanding Cybercriminals, Profiling Cybercriminals, Categorizing Cybercriminals, Understanding Cyber victims, Categorizing Victims of Cybercrime, Making the Victim Part of the Crime-Fighting Team, Understanding Cyber investigators, Recognizing the Characteristics of a Good Cyber investigator, Categorizing Cyber investigators by Skill Set, Analyzing Gendered Aspects of Cyber Victimization, Digital Literacy and Awareness Campaigns.

Computer Investigation Process: Demystifying Computer/Cybercrime, Investigating Computer Crime, How an Investigation Starts, Investigation Methodology, Securing Evidence, Before the Investigation, Professional Conduct, Investigating Company Policy Violations, Policy and Procedure Development, Policy Violations, Warning Banners, Conducting a Computer Forensic Investigation, The Investigation Process, Assessing Evidence, Acquiring Evidence, Examining Evidence, Documenting and Reporting Evidence, Closing the Case.

Acquiring, Duplicating and Recovering Deleted Files: Recovering Deleted Files and Deleted Partitions, recovering "Deleted" and "Erased" Data, Data Recovery in Linux, Recovering Deleted Files, Recovering Deleted Partitions, Data Acquisition and Duplication, Data Acquisition Tools, Recovering Data from Backups, Finding Hidden Data, Locating Forgotten Evidence, Defeating Data Recovery Techniques, Responsible Consumption of Digital Resources during Forensic Investigations.

Collecting and Preserving Evidence: Understanding the Role of Evidence in a Criminal Case, Defining Evidence, Admissibility of Evidence, Forensic Examination Standards, Collecting Digital Evidence, Evidence Collection, Preserving Digital Evidence, Preserving Volatile Data, Special Considerations, Recovering Digital Evidence, Deleted Files, Computer Forensic Information, Understanding Legal Issues, Searching and Seizing Digital Evidence

Building the Cybercrime Case: Major Factors Complicating Prosecution, Difficulty of Defining the Crime, Jurisdictional Issues, The Nature of the Evidence, Human Factors, Overcoming Obstacles to Effective Prosecution, The Investigative Process, Investigative Tools, Steps in an Investigation, Defining Areas of Responsibility.

Self-Learning Contents:

Acquiring, Duplicating and Recovering Deleted Files: Deleted Partition Recovery Tools, Deleted File Recovery Tools, Data Acquisition and Duplication Tools, Defeating Data Recovery Techniques.

Collecting and Preserving Evidence: Data Recovery Software and Documentation, Computer Forensic Resources, Computer Forensic Training and Certification, Computer Forensic Equipment and Software, Computer Forensic Services.

Laboratory Work:
Hands with open source tools for forensic investigation process models (from Item confiscated to submitting evidence for lawful action), such as FTK, Sleuth Toolkit (TSK), Autopsy, etc.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:

1. Familiarize with cybercrime & forensics ontology
2. Investigation of crime scene and acquisition of digital evidence.
3. Recovery and analysis of digital data of evidential value.
4. Investigate computer based crime and create document for judicial proceedings.

Text Books:

1. Shinder L. D., Cross M., Scene of the Cybercrime, Syngress (2008) 2nd ed.
2. Marcella J. A. and Guillossou F., Cyber Forensics: From Data to Digital Evidence, Wiley (2012).
3. Nina Godbole, Sunit Belapure, Cyber Security, Wiley (2011).

Reference Books:

1. Marcella J. A. and Menendez D., Cyber Forensics: A Field Manual for Collection, Examining and preserving Evidence of computer crimes. Auerbach Publication (2010), 2nd ed.

Dejey, Murugan, Cyber Forensics, Oxford (2018).

UCS754: BLOCKCHAIN TECHNOLOGY AND APPLICATIONS

L	T	P	Cr
2	0	2	3.0

Course Objectives: The objective of this course is to provide exposure on blockchain technology and its real-time applications.

Basic Cryptography: Introduction to cryptography and cryptanalysis, Cryptographic issues, cryptographic components, cryptographic techniques, cryptographic categories: symmetric key and asymmetric key cryptography, traditional ciphers, modern ciphers, message integrity, message authentication, key management, digital signatures, entity authentication, ECDSA, ECC, Ring, One time signature, Hashing: SHA-356, SHA-512, TLS and SSL, Timestamp, Public and Private keys, Merkle root hash.

Bitcoin Cryptocurrencies: What is Bitcoin, Brief history of Bitcoin, Bitcoin mining and supply, Bitcoin cryptocurrency (BTC), Traditional centralized vs. decentralized, Bitcoin's blockchain: evolution of blockchain, block header, genesis block, hash generation, Bitcoin address: formats, hash generation, address structure, transactions: multi-signatures, generating transactions, storing data, block verification and validation, block mining.

Smart Contracts: Introduction to smart contracts, smart contracts used in a centralized and decentralized systems, Blockchain platforms using smart contracts: Ethereum, architecture of Ethereum virtual machine, token- ETH, Mining process, ERC- standards, transactions in Ethereum, Hyperledger fabric, Sidechains, NXT, Stellar, R3Conda, Litecoin, Quorum, IBM, Openchain, Eris:db, Case studies related to smart contracts for transparent aid distribution .

Consensus Mechanisms: Double spending problem, BFT, PBFT, PoW, PoS, DPoS, PoA, PoB, PoR, PoET, PoI, PoO, PoSp, PoC, Ripple, Tendermint, energy consumption of traditional consensus mechanisms and alternative eco-friendly mechanisms.
.

Applications of Blockchain: Financial system, smart grid, healthcare , smart transportation system, e-Governance, education, exchange and trading, online market place, commercial supply chain, food production, drug manufacturing, safety and security.

Laboratory Work:
Experiments on creating of blockchain, implementation of smart contract on Python, Conda and Ethereum, Solidity, Implementing **carbon credit systems** using blockchain.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course, the student will be able to:

1. Create their own blockchain using Block creation and verification
2. Create the smart contracts for transaction execution
3. Evaluate the performance of blockchain in presence of various attacks
4. Develop and validate various security models for real-life applications.

Text Book:

1. Melanie Swan, “Blockchain: Blueprint for a new economy”, Oreilly publications.

Reference books:

1. Bellaj Badr, Rcihcard Horrocks and Xun Brian Wu, “Blokchain by example”, Packt Publications.
2. Fatima Castiglione Maldonado, “Introduction to Blockchain and Ethereum”, Packt Publications.

EFB

Mathematics and Computing

UMC513:Linear Algebra for Artificial Intelligence and Machine Learning

L	T	P	Cr
2	0	2	3.0

Course Objective: This course aims to build a strong mathematical foundation in linear algebra for AI/ML applications. It focuses on geometric intuition behind matrix operations and transformations while connecting theoretical concepts to practical machine learning models. MATLAB-based implementation of matrix algorithms will be incorporated to strengthen computational insight and practical skills. Students are expected to have prior knowledge of basic linear algebra.

Norms and Matrix Calculus: Review of vector spaces and matrices, Introduction to norms and inner product spaces, Orthogonality projections, Orthonormal basis of vector space, orthogonal matrices, Inner product space, Kronecker product and block matrices.

Eigenvalues and Matrix Decompositions: Existence of eigenvalues on complex vector spaces, Eigenvalues and minimal polynomials, Eigenvectors, Algebraic and geometric multiplicity, Diagonalization, Conditions for diagonalizability, Gershgorin disk theorem, spectral theorem, Positive definite matrices, Singular Value Decomposition (SVD) and Geometric properties, recommendation models, Low-rank approximation, Principal Component Analysis (PCA), noise reduction techniques, dimensionality reduction, covariance matrices.

Matrix Transformations: Householder matrices and their applications, Householder transformation.

Applications in AI/ML: Tensors, Google Page rank algorithm, Face recognition, computer vision, spectral clustering

Laboratory Work Lab work based on programming in MATLAB of various linear algebra and machine learning concepts

Text Books

1. Introduction to Applied Linear Algebra- Vectors, Matrices, and Least Squares, Stephen Boyd and Lieven Vandenberghe, Cambridge University Press, 2018.
2. Matrix computations, Gene Golub, C. F. Van Loan, Hindustan Book Agency, 2015.
3. Linear Algebra and Learning from Data, Gilbert Strang, Wellesley-Cambridge Press, 2019.

Course Learning Objectives (CLO)

The students will be able to:
1. Analyze and solve systems of linear equations and understand vector space structures used in AI models.
2. Apply eigenvalue decomposition and SVD for dimensionality reduction and data representation.
3. Analyze the use of householder transformations
4. Implement linear algebra-based machine learning algorithms such as PCA, regression, spectral clustering and PageRank

UMC632: FINANCIAL MATHEMATICS

L	T	P	Cr
2	0	2	3.0

Course Objectives: This is an introductory course in finance to equip with a framework and basic techniques necessary for financial engineering. The main focus is on valuation of financial assets and more specifically derivative products. The course will introduce the concept of risk and relation between risk and return. The knowledge of risk and valuation will be integrated in optimal decision-making. The models will be studied in discrete-time scenario.

Basics of Financial Mathematics: Financial markets, terminologies, basic definitions and assumptions, Interest rate, present value, future value, NPV, annuity and perpetuity, no arbitrage principle, bonds and bonds pricing.

Portfolio Optimization: Portfolio optimization: introduction, risk, return, two-assets portfolio, Markowitz curve, efficient frontier, multi-assets all risky portfolio, mean-variance Markowitz model, two fund theorem, one fund theorem, the capital asset pricing model, mean absolute deviation model.

Futures and Forwards: Introduction to derivatives, forward and futures contracts and their pricing without dividends, hedging strategies using futures.

Theory of Option Pricing: Introduction to call and put options, pay-off curves of option combinations, profit diagrams, hedging and speculation properties of options, valuation of options using pricing and replication strategies, mathematical properties of their value functions, put-call parity. Binomial lattice model, single and multi-period binomial formula for pricing European style and American style options, dividend and non-divided cases, CRR model, Black-Scholes model for option pricing.

Laboratory activities: Extraction of data from various online resources like NSE, moneyconrol.com etc. Implementation and validation of various models studied in the course for option and portfolio valuation using Matlab/R/Excel/Python.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After completion of the course, students will be able to:

1. Understand basic quantities that are reported in everyday life such as interest rates, periodic payments of money, dividends, shares, bonds, forwards, futures etc.
2. Construct a portfolio which is optimal in a given market scenario.
3. Evaluate call and put option prices using various models.

Text Books / Reference Books:

1. D.G. Luenberger, Investment Science, Oxford University Press, 1999 (new edn. 2013).
2. S. Chandra, S. Dharmaraja, A. Mehra, R. Khemchandani, Financial Mathematics: An Introduction, Narosa, 2012.
3. M. Capinsky and T. Zastawniak, Mathematics for Finance: An Introduction to Financial Engineering, Springer, 2004 (new edn, 2011).
4. J C Hull, Options, Futures and other Derivatives, Prentice Hall, 8th edn, 2011.
5. J H Cochrane, Asset Pricing, Princeton University, 2000 (new edn 2005).

UMC633: MATHEMATICS FOR QUANTUM COMPUTING

L	T	P	Cr
2	0	2	3.0

Course Objective: Course Objectives: The objective of this course is to provide the students with the necessary mathematical foundations of quantum computation

Unit 1: Hilbert spaces, inner-product structure and Dirac bra-ket notation. Qubit as a vector, Bloch-sphere picture. Matrix and outer-product representations, Pauli matrices. Hermitian operators (observables) and unitary operators (time-evolution), eigenvalues & eigenvectors. Measurement & expectation, Trace and its properties; expectation values via the trace rule; core postulates of quantum mechanics. Composite systems, Tensor product for multi-partite states, no-cloning theorem. Density-operator formalism, Pure, mixed, and completely mixed states, density-operator properties, partial trace and reduced density operators, Bloch representation of density matrices.

Unit-II: Measurement, Entanglement, Bell's Theorem, Bipartite Systems and Bell Basis, Separable State, Entangled state, Schmidt Decomposition and Schmidt number.

Unit-III: Single qubit gates, multiple qubit gates, universal quantum gates, Quantum circuits, design of quantum circuits.

Unit-IV: QuantumInterference, Quantum Teleportation, Entanglement Swapping, Superdense Coding, Deutsch-Josza algorithm, Grover's algorithm, Simon's Algorithm, Quantum Fourier Transform, Phase estimation, Shor's algorithm

Laboratory Work Implementation of Quantum concepts in Qiski / IBMQuantum Platform /MATLAB / Pennylane.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After completion of the course, students will be able to:

1. Understand basic quantities that are reported in everyday life such as interest rates, periodic payments of money, dividends, shares, bonds, forwards, futures etc.
2. Construct a portfolio which is optimal in a given market scenario.
3. Evaluate call and put option prices using various models.

Text Books / Reference Books:

1. D.G. Luenberger, Investment Science, Oxford University Press, 1999 (new edn. 2013).
2. S. Chandra, S. Dharmaraja, A. Mehra, R. Khemchandani, Financial Mathematics: An Introduction, Narosa, 2012.
3. M. Capinsky and T. Zastawniak, Mathematics for Finance: An Introduction to Financial Engineering, Springer, 2004 (new edn, 2011).
4. J C Hull, Options, Futures and other Derivatives, Prentice Hall, 8th edn, 2011.
5. J H Cochrane, Asset Pricing, Princeton University, 2000 (new edn 2005).

UMC744: CRYPTOGRAPHY AND CODING THEORY

L	T	P	Cr
2	0	2	3.0

Course Objective: The course intends to provide an introduction to theory of congruences, prime numbers, and quadratic residues. The focus then shifts to computational aspects and applications in cryptography (for secure encryption and decryption methods) and coding theory (for reliable communication and error correction) used in communication systems.

Congruences and their applications: Primes, Fermat and Mersenne primes, Primality testing and factorization, Linear congruences, Chinese Remainder Theorem, congruences with aprime power modulus, Fermat‘s little theorem, Wilson‘s Theorem, Euler function, Quadratic Residues, Legendre Symbol, Euler‘s criterion.

CryptographyBasics: Symmetric and asymmetric key cryptography, Pseudo-primes, Pseudoprimality Testing, Randomized Primality test, AKS primality test, Randomized Primality test &Deterministic Polynomial Time Algorithm, Pollard-Rho Method.

Public key Cryptosystems: RSA, Diffie Hellmann key exchange, different attacks and Remedies, Digital Signature, Elliptic curve cryptography. Coding Theory: Error-correcting codes, Bounds on general codes, Linear codes, Dual codes, Hamming codes, Cyclic codes.

Laboratory Work: Implementation of traditional ciphers and coding theory techniques, including error correcting codes.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
On successful completion of this course, students will have the knowledge and skills to:

1. . Find the greatest common factor using the Euclidean Algorithm, investigate different factorization methods and solve linear congruences.
2. ApplyWilson'sandFermat's Little Theorem as the basis for primality tests and factoring algorithms.
3. Apply and analyse elementary number theory concepts to symmetric and asymmetric key cryptography for encrypting and decrypting a message.
4. Demonstrate the application of coding theory techniques for reliable data transmission and error correction.

Text Books:

1. Neal Koblitz, A course in Number Theory and Cryptography, Springer, 2006
2. Niven, H.S. Zukermann, H.L. Montgomery, An introduction to theory of numbers, Willey, 2015
3. D. Burton, Elementary Number Theory, McGraw-Hill, 2012
4. J. S. Kraft and L. C. Washington, An Introduction to Number Theory with Cryptography, 2018
5. W.Trappe,L.C.Washington,Introduction to Cryptography with CodingTheory, 2020.

Reference Books:

1. Behrouz A. Forouzan, D. Mukhopadhyay, Cryptography and Network Security, McGraw Hill, 2015.
2. J. Pipher, J. Hoffstein and J.H. Silverman, An introduction to Mathematical Cryptography, Springer-verlag 2014.

UCS539: Finance, Accounting and Valuation

L	T	P	Cr
2	0	2	3.0

Course Objectives: Understanding relationship of finance, accounting and valuation of securities.

Introduction to Accounting: Meaning of accounting, the accounting process, fundamental equation, types of accounts, accounting statements, recording of transactions, conceptual framework

Summary Statements: Types of summary statements, preparation of the statements, relationship between the statements, introduction to financial statement analysis

Basics of Finance: Meaning of finance, process of financial decision making, types of financial decisions, capital structure decisions.

Time Value of Money: Meaning, principle, calculations, interest rates, importance of interest rates, importance of different types of interest rates and returns.

Valuation: Introduction to valuation, valuation of stocks, valuation of bonds, methods and techniques

Practical sessions: To gain an understanding of proprietary software for international derivatives

1. Introduction to proprietary software.
2. Detailed understanding of basic features of proprietary software.
3. Understanding order types and their implementation
4. Regular practice to understand execution of basic strategies.
5. Introduction to trading and investment.
6. Introduction to analytical methods

Recommended Prerequisites: Basics of Microsoft Excel

Course learning outcomes (CLOs): After the completion of the course, the student will be able to:

1. Explain the basic accounting concepts and apply the fundamental equation in basic business transactions
2. Explicate and apply the techniques learnt for doing financial statement analysis
3. Explain various financial decisions and evaluate some of them
4. Explicate the principle of time value of money (TVP) and importance of interest rates in TVP
5. Apply the methods learnt for valuation of securities

Reference Books:

1. Jamie Pratt. (8th Edition). Financial Accounting in an Economic Context.
2. Ross, Westerfield, Jaffe & Jordan. Corporate Finance: Core Principles and Applications.

UCS675: Financial Markets and Portfolio Theory

L	T	P	Cr
2	0	2	3.0

Course Objectives: Understanding various financial markets and their interrelationships

Banks and Financial Institutions: Types of financial institutions, evolution of financial system, flow of money, creation of money

Monetary System: Monetary authority, monetary policy framework, policy tools, comparison of different countries

Risk and Return: Meaning of risk, meaning of return, estimation of risk and return

Capital and Money Markets: Meaning, types, capital and money market instruments

Portfolio Theory: Meaning of portfolio, theoretical principles, choices, optimal weights, optimal portfolio choice, introduction to pricing models

Practical sessions:

To use the proprietary software in live international derivatives

1. Introduction to an international derivatives product
2. Introduction to technical analysis
3. Practice of analytical methods on the proprietary software
4. Introduction to evaluation methods

Recommended Prerequisites: Course – Finance, Accounting and Valuation;

Course learning outcomes (CLOs):

After the completion of the course, the student will be able to:

1. Explain the role of the financial system and the process of creation of money
2. Explicate various monetary policy tools
3. Explain relationship between risk and return
4. Explicate the types of capital markets and money markets including the market instruments
5. Apply the portfolio theory to choose an optimal portfolio

Reference Books:

1. Financial Markets and Institutions – Anthony Saunders & Marcia Millon Cornett
2. Ross, Westerfield, Jaffe & Jordan. Corporate Finance: Core Principles and Applications.

UCS658: Derivatives Pricing, Trading and Strategies

L	T	P	Cr
2	0	2	3.0

Course Objectives: Understanding methods of valuation and strategies of trading derivative instruments

Law of One Price: Meaning, implication of the law of one price, no arbitrage model, usage in pricing of securities and derivative instruments

Pricing and Valuation: Basic principles, building blocks, assumptions, difference between price and value, pricing and valuation of basic derivative instruments

Basics of Option Pricing: Meaning of options, types of options, difference between options and basic derivative instruments

Practical sessions: To use the proprietary software in live international derivatives

1. Introduction to derivatives strategies
2. Introduction to an additional international derivatives product
3. Learning and creating trading strategies
4. Practicing the strategies on the proprietary software
5. Understanding the role of derivatives in risk reduction

Recommended Prerequisites: Courses Finance, Accounting and Valuation + Financial Markets and Portfolio Theory

Course learning outcomes (CLOs):
After the completion of the course, the student will be able to:

1. Explain and apply the law of one price in pricing models
2. Explicate and apply formulae to calculate price and value of a generic forward contract
3. Explain and apply formulae to calculate price and value of basic derivative instruments
4. Explicate the meaning and types of options
5. Apply the basic methods to calculate option prices

Reference Books:

1. Fundamentals of Futures and Options Markets - John C. Hull
2. Ross, Westerfield, Jaffe & Jordan. Corporate Finance: Core Principles and applications.

UMC743: Quantitative and Statistical Methods for Finance

L	T	P	Cr
2	0	2	3.0

Course Objectives: Understanding quantitative and statistical methods used for finance and derivatives:

Refresher on Statistics: Correlation, OLS regression, probability distributions and moments, using Microsoft Excel for statistical calculations and interpretations

Option Pricing Introduction to option pricing models, formulae and derivation, option Greeks, risk management using options

Financial Time Series: Introduction to time series, types, univariate and multivariate time series models, autocorrelation, AR models, MA models, ARMA models, ARIMA models, stationary series, unit-root

Volatility: Meaning of volatility, types, methods of calculation, volatility models, estimation

Practical sessions: To use the proprietary software in live international derivatives

1. Introduction to other derivative products
2. Refining trading strategies created in previous courses.
3. Practicing the strategies on the proprietary software.
4. Application of the quantitative and statistical methods
5. Introduction to algorithmic trading on the proprietary software

Recommended Prerequisites: Courses – Finance, Accounting and Valuation + Financial Markets and Portfolio Theory + Derivatives Pricing, Trading and Strategies

Course learning outcomes (CLOs): After the completion of the course, the student will be able to:

1. Explain and apply the time series models learnt for financial data
2. Explicate and apply PCA for financial data
3. Explain and apply option pricing models to calculate prices of options
4. Explicate types of volatility and volatility models
5. Apply the basic methods learnt to estimate volatility

Reference Books:

1. Introduction to Time Series Analysis and Forecasting - Douglas C. Montgomery, Cheryl L. Jennings
2. Fundamentals of Futures and Options Markets - John C. Hull

EFB

DevOps and Continuous Delivery

UCS537: SOURCE CODE MANAGEMENT

L	T	P	Cr
2	0	2	3.0

Course Objectives: The objective of the course is to teach techniques to combine software development and IT operations using DevOps. It helps to understand faster software development practices with higher quality.

Traditional Software Development: The Advent of Software Engineering, Waterfall method, Developers vs IT Operations conflict.
Rise of Agile methodologies: Agile Vs Waterfall Method, Iterative Agile Software Development, Individual and team interactions over processes and tools, working software over comprehensive documentation, Customer collaboration over contract negotiation, responding to change over following a plan
Definition and Purpose of DevOps: Introduction to DevOps, DevOps and Agile, Minimum Viable Product, Application Deployment, Continuous Integration, Continuous Delivery
CAMS (Culture, Automation, Measurement and Sharing): CAMS – Culture, Automation, Measurement, Sharing, Test-Driven Development, Configuration Management, Infrastructure Automation, Root Cause Analysis, Blamelessness, Organizational Learning.
Cloud Sustainability: Choosing eco-friendly cloud services (e.g., AWS, Azure, or GCP with carbon-neutral data centers) aligns with DevOps goals and sustainability.
Typical Toolkit for DevOps: Introduction to continuous integration and deployment, Version control system
Source Code Management History and Overview: Examples - SVN, Mercury and Git, History - Linux and Git by Linus Torvalds,
Version Control System: Version control system vs Distributed version control system: Local repository, Advantages of distributed version control system, The Multiple Repositories Models, completely resetting local environment, Revert - cancelling out changes.

Laboratory work:
Basic structure and Implementation of various distributed version control systems for source code management.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After completion of this course, the students will be able to:

5. Identify the need for migrating from traditional software development to Agile model and then to DevOps.
6. Define and understand the basic principles and need of DevOps and Continuous Delivery.
7. Understand the history and overview of Source Code Management, along with real-time examples.
8. Differentiate between centralized and distributed version control systems and basic operations in version control systems and Demonstrate the use of various version control systems.

Text Books:

3. The DevOps Handbook - Book by Gene Kim, Jez Humble, Patrick Debois, and Willis Willis.
4. Pro Git – Book by Scott Chacon and Ben Straub (available at https://git-scm.com/book/).

Reference Books:

4. What is DevOps? - by Mike Loukides.

UCS659: BUILD AND RELEASE MANAGEMENT

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course includes theory and lab. The course comprises four modules. The main objective of this course to help participants understand the process of build and release management.

Introduction to Build and Release Management: Introduction to build, understanding different phases of build and release management, Measuring and minimizing carbon footprints of software builds, introduction to release management, best practices for build and release management, Intelligent scheduling for release deployments to reduce energy consumption, concept of build abstraction and dependency abstraction.

Dependency Management: Introduction to dependency management, Best practices for versioning to reduce duplication and waste, how to use source code repositories, managing transitive dependencies, dependency scope and discussion of various tools like Ant, Maven and Gradle, Tools for measuring energy usage in software delivery pipelines.

Document and Reporting: Introduction to build document and reporting, incorporating sustainability principles in project documentation, different types of documentation, understanding site life cycle, advance site configurations and reports, generation of unit test reports, generation of code coverage reports, code coverage tools, code coverage pros and cons.

Release Cycle: To understand project release life cycle, different stages of release lifecycle, source code repositories, using renewable-energy-based data centers for redundancy, how to install and configure source code repositories and deploying build to production goals- prepare, perform, clean and rollback.

Laboratory work: Setting up Maven environment and understanding POM hierarchy, creation of a project using Maven and its configurations.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After completion of this course, the students will be able to:

1. Explain the basics of build and release management by learning build abstraction and declarative dependency management.
2. Describe dependency management and the associated concepts like repositories, dependency identification and scope, transitive dependencies, and the examples for build tools.
3. Discuss the process of documentation and reporting, using site life cycle, site configuration and generation of unit testing and code coverage reports
4. Define release cycle and the phases of release, preparing, cleaning and performing goals.

UCS660: Continuous Integration and Continuous Deployment

L	T	P	Cr
2	0	2	3.0

Course Objectives: The objective of the course is to teach techniques to automate the process of integration and deployment software product. It covers prerequisites, anatomy and framework/tools used for the automated process of continuous integration and continuous deployment.

DevOps Automation: Phases in software development-delivery pipeline, components of automated software delivery, RAD model and model driven architecture.
Automation Benefits: advantages of automation, Time and efforts saving scenarios, error preventing scenarios, energy efficiency and resource optimization: scenarios/case study.
Continuous Integration and Continuous Deployment Introduction: Overview and practices of continuous integration, working mechanism and benefits of continuous integration; continuous delivery's introduction and pipeline. Prerequisites and benefits in efficient delivery cycles, reducing energy consumption, introduction and business drivers of continuous deployment, benefits of continuous deployment.
Stages and Anatomy of CI CD: Core continuous integration process and advanced continuous integration process, release process, continuous delivery engineering practices, continuous testing & promotion of builds, continuous monitoring of delivery pipeline for optimized resources usage, understanding continuous feedback process.
Testing, Debugging and Refactoring: Understanding test-driven development (TDD), categories of TDD, Junit framework, need for code refactoring, its process and strategies in reducing defects and rework
Understanding Framework and Tools: Common frameworks and code architectures, third party code, IDEs (Eclipse, Netbeans and IntelliJ), common mistakes and avoiding them, issues with making code IDE dependent.

Laboratory work:
Setting up Jenkins, Jenkins job, parameters, build, post-build actions and pipeline; Jenkins plugins, using Jenkins as a continuous integration server; Configuring Jenkins with git plugin; Jenkins pipeline to poll the feature branch.
.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After completion of this course, the students will be able to:

1. Understand the phases of software development-delivery pipeline and automation benefits.
2. Identify and apply continuous integration and deployment prerequisites, process and benefits.
3. Understand and implement the continuous delivery engineering practices and release process.
4. Identify & use the test-driven deployment and various tools/frameworks used for continuous integration and delivery in DevOps

Text Books:

1. Gene Kim, Jez Humble, Patrick Debois, John Willis, "The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations", IT revolution Press (2016) 1st ed.

Reference Books:

1. Sander Rossel, "Continuous Integration, Delivery, and Deployment: Reliable and Faster Software Releases with Automating Builds, Tests, and Deployment", Packt Publishing (2017) 1st ed.

Online material available at:https://digitallearn.xebiaacademyglobal.com/

UCS758: System Provisioning and Configuration Management

L	T	P	Cr
2	0	2	3.0

Understanding Containers: Transporting goods analogy and its problems, Containerization platform, images and runtime, comparison with virtual machine, chroot system call, FreeBSD Jails, LinuX containers (LXC), Docker.

Introduction to Containerization: Docker architecture, different environments (Dev, QA and Prod), overcoming issued with different environments, virtual machine for dev/deployments, containers for dev/deployments, advantages and drawbacks of containerization.

Orchestration Tools: Orchestration: its definition and need, Docker swarm and Kubernetes, AWS (ECS and EKS), Kubernetes on cloud, monitoring containers and its process.

Introduction to Provisioning: Basic and software definition, provisioning concepts, reason for exclusive provisioning, configuration management definition and tools, difference between provisioning and configuration management, provisioning tools, test machines for provisioning, deployment and its relationship with provisioning.

On Premise Provisioning: Understanding and Defining On Premise, On Premise provisioning infrastructure, Templating, server templating and its challenges.

Provisioning on Cloud: defining cloud provisioning, types of cloud provisioning, life-cycle of provisioning on cloud, On Premise cloud mitigation strategies, network security enablement from On Premises to cloud, micro-services management in cloud, Environmental impact of cloud technologies, Benefits of Sustainable System Provisioning.

Provisioning and Configuration Management: State of tools in provisioning and configuration, definition and need for configuration management, its benefits and drawbacks in DevOps, need for monitoring in DevOps, reasons for using provisioning and configuration tools, automation, preventing errors and tracking changes, examples of tools and their capabilities.

Laboratory Work:
System Provisioning: Automation of infrastructure, AWS configuration for Terraform, create IAM User, security group, spinning up with EC2 instance, variables, resources, modules, state management, VPC, IAM policy, S3 bucket and its variables.

Containers Lab: Playing with Vagrant and understanding its file, Docker machine, Dockerfile, Docker extras, DTR, Docker compose and swarm, Kubernetes -Minikube, deploying Pods and services on Minikube.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course the student will be able:

1. Understand the concept of Virtualization and Containerization.
2. Familiarize with Orchestration and System provisioning.
3. Analyze and demonstrate the infrastructure automation and state management in the cloud

environment.

4. Understand and demonstrate the need for configuration management.

Text Book:

Course material by Xebia, *Xebia Academy: Log in to the site (*learnhub.xebia.com*)*.

Gerardus Blokdyk, *Server Provisioning and Configuration Management Complete Self-Assessment Guide*, 5STARCooks, 2018.

Reference Book:

Erika Heidi, *An Introduction to Configuration Management*, Tutorials, 2019.

EFB

Full Stack

UCS542: UI & UX SPECIALIST

L	T	P	Cr
2	0	2	3.0

Course Objectives: The main objective of this course is to impart concepts related with web technology which are essential for the development of web applications. The key technology components include web markup languages, client and server side programming.

HTML& CSS: Introduction to HTML, Introduction, HTML Page Structure, Create HTML document, Understand the various elements available in HTML, HTML Use, Attributes in HTML, Need of Attributes, Common Attributes, HTML forms, Apply validations to the form elements, Creating web pages with HTML5, HTML5 introduced features, HTML5 form validate/no validate, HTML5 canvas, embedding audio, and video in a webpage, drag and drop, HTML5 Local Storage, HTML5 web workers and server sent events. What is CSS, how to insert CSS in HTML, How CSS adds value to HTML, Difference between Semantic and HTML mark- up, CSS3, CSS Selectors, Buttons, CSS float and clear, CSS align - horizontal and center, CSS Padding, CSS Links, CSS Lists, CSS Tables.

JavaScript: What is JavaScript, Importance of JavaScript, What can JavaScript Do?, JavaScript with HTML Attributes, JavaScript with CSS, Operators, JavaScript Syntax, JavaScript Data Types, JavaScript Functions, Setting up Environment, Variables, Control flow, if. Else, switch, loops, JavaScript HTML DOM Elements, JavaScript Syntax, Operators, Data Types, JavaScript String Methods, JavaScript Functions, Arrays, Sorting, Joins, Reduce map.

Frontend Architecture: Introduction to Frontend Development, History, MVC, MVP, MVVM& Web Apps, Resource Optimization, Development of AJAX, Introduction to DOM, Basic DOM Manipulation, Reactive Programming, Refreshing ES6 Specifications and Features ECMA Script, ES6 let and const, The arrow functions, New Literal Syntax, Classes, Inheritance using extends, Default Parameter Values, Spread Operator (...), Iterators and Generators, Features of React, Practical Application, Why need React, How React Works, Leveraging Virtual DOM, Setting up React.

REST API, JSX: Why JSX, Modular and Scalable Design, Embedding JavaScript Expression in JSX, JSX Attributes, JSX Comments, Styling and Representation as Object, The State of the Component, Changing the State, Props of Component, Using Props, Props Validation in React, Similarities Between State & Props RESTAPI : Intro to API, History of API Development, Development of AJAX, CRUD; GraphQL; HTTP ,HTTP 1.1,HTTP/2, Stream prioritization, Introduction to React Native, Setting up React Native, The Expo Client, Working up on the First Project, Style, Fexbox Layout.

Node.js: Introduction to Node.js, History, Why Node.js, Node.js Architecture, Features, Working of Node.js, Installation & Setting Up Node, setting up React, REPL Environment,

REPL Commands, Variable, Components of Node.js, Local Modules, Module Exports: Export Object, Export Class, Loading Module from Separate Folder, Operating System, File Systems.

Course Objectives (COs):

After the completion of the course the student will be able:

1. To acquire knowledge and skills for creation of full web framework considering both client and server side programming.
2. To develop a dynamic and interactive WebPages by the use of HTML, XHTML, CSS, Java script and DHTML.
3. To build web apps with fully functional frontend with database connectivity.
4. To undertake transfer of multimedia content over the Internet between the multi platforms web applications through communication protocols and the learner will also be able to build web API.

Text Books:

1. M. Srinivasan, Web Technology: Theory and Practice, Pearson (2012).
2. A.Kumar, Web Technology: Theory and Practice, CRC Press (2018).

Reference Books:

1. Web programming with HTML, XHTML and CSS, 2e, Jon Duckett, Wiley India.
2. Web technologies: A Computer Science Perspective, Jeffrey C. Jackson ,1st ed., Pearson (2007).

UCS677: DATA ENGINEERING

L	T	P	Cr
2	0	2	3.0

Course Objectives: Basic concepts of database, Mongo DB, SQL, and Java script.

Getting started with MongoDB: No SQL Databases, Features of MongoDB, Installation overview, Documents, Collections, Databases, What isthe NoSQL approach? Why Use the NoSQL Approach, Benefits of No SQL, Types of Databases, Key-Value Stores, Wide-column Stores/ Columnar Databases, Document/Document-store/Document-oriented Databases, Graph- based Databases, Starting and stopping MongoDB

Javascript in MongoDB: Javascript in MongoDB, Execution of a JavaScript file in MongoDB, Making the output of find readable in shell, Complementary Terms, Installation, Basic commands on mongo shell, HelloWorld, Create, Update Delete, Read, Update of embedded documents, more update operators, Updating multiple documents.

Collections: List all collections in the database, List all databases, Find(), FindOne( ), limit, skip, sort and count the results of the find() method, Query Document – Using AND, OR and IN Conditions, find() method with Projection, Find() method with Projection, $set operator to update specified field(s) in document(s), Insert a document, Create a Collection, Drop Collection, Aggregation

Indexes: Indexes, Index Creation Basics, Dropping/Deleting an Index, Sparse indexes and Partial indexes, Get Indices of a Collection, Compound, Unique Index, Single field, Delete, List, Mongoas Shards

Sharding Environment Setup: Managing Database for Availability and Performance, Database Scaling, Database Distribution Models, Database Replication, Types of Database Replication, Master-Slave Replication, Peer-to-Peer Replication, Advantages and Disadvantages of Peer-to- Peer Replication, Introduction to Sharding, Why Sharding, The Lookup Strategy, Basic configuration with three nodes, Mongo as a Replica Set, Mongoose.

Course Objectives (COs):

After the completion of the course the student will be able to:

1. Learn NoSQL, Mongo DB, and Javascript.
2. Apply basic concepts of indexing and collections using MongoDB.
3. Formulate the basic usage of database such as graph-based database, creating and dropping of collections, andindexing and solving real-world problems.
4. Able to handle and update multiple documents using Mongo DB.
5. Able to manage database for availability and performance, and scaling of database.

Text Books:

1. Andreas Kretz, The Data Engineering Cook Book, 6th ed. (2019)

2. Alex Petrov, Database Internals: A deep dive into how distributed data systems work, O‘REILLY Publication (2021).

Reference Books:

1. S. Bradshaw, Eoin Brazil, and Kristina Chodorow, MongoDB: The definite guide: Powerful and Scalable Data Storage, O‘REILLY Publication (2021).
2. Dan Sullivan, NoSQL for Mere Mortals, O‘REILLY Publication (2021).

UCS662: TEST AUTOMATION

L	T	P	Cr
2	0	2	3.0

Course Objectives: The course provides understanding of software testing and how to use various tools (like Selenium and TestNG etc.) used for automation of software testing.

Introduction to Software Testing: Seven principles of Software Testing, SDLC vs STLC, Testing Life Cycle, Usability Testing, why do we need Usability Testing, how to do Usability testing, Advantages & Disadvantages, Functional Testing, End to End Testing, Methods, Advantages & Disadvantages, Compatibility Testing, Types GUI testing, Techniques API testing, Advantages

Test Automation: Selenium: Selenium components, Selenium Architecture, TestNG: Installing TestNG in Eclipse, TestN Gannotations – Understanding usage, setting priority of execution for test cases, Hard Assertion, Soft Assertion, TestNG Reports, ANT- Downloading & Configuring, XSLT report generation using TestNG and Ant.

Introduction to Selenium 3.x: Describe Selenium 3.x advantages and implementation, Define drivers for Firefox, IE, chrome, IPhone, Android etc. Analyze first Selenium Code, differentiate between Close and Quit, Describe Firepath and firebug Add-ons installation in Mozilla, inspect elements in Mozilla, Chrome and IE, Identifying Web Elements using id, name, class, Generate own CSS Selectors. Differentiate between performance of CSS Selectors as compared to Xpaths, define class attribute, Handle Dynamic objects/ids on the page, Analyze whether object is present on page or not

Manual Testing: Manual Testing, Manual Testing – How to Approach? Manual Testing – Myth and fallacy, Defect Life Cycle, Qualities of a good Manual Tester, Manual Testing Vs Automation Testing, Types, System Testing, Acceptance Testing, Unit Testing, Techniques, Integration Testing, Smoke- Sanity Testing

Introduction to Test Design and Sustainable Software Testing Practices: Test Scenario, Test Case Design, Test Basis Traceability Matrix, Energy-efficient coding and testing strategies, Integrating sustainable practices in STLC, Balancing scalability and energy consumption during automated testing, Case studies of sustainable software projects.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Understand the concepts related to software testing and test automation.
2. Take into account the different considerations when planning automated tests vs. manual tests
3. Architect the test project and fit it to the architecture of the tested application
4. Design and implement highly reliable automated tests
5. Understand how different types of automated tests will fit into your testing strategy, including unit testing, load and performance testing, visual testing, and more

Text Books:

1. Axelrod Arnon, Complete Guide to Test Automation: Techniques, Practices, and

Patterns for Building and Maintaining Effective Software Projects, A press (2018).
2. Gundecha U. and Cocchiaro C., Learn Selenium: Build data-driven test frameworks for mobile and web applications with Selenium Web Driver 3, Packt (2019).

Reference Books:

1. Diego Molina, Selenium Fundamentals, Packt (2018).
2. Aditya P. Mathur, Foundations of Software Testing, Pearson Education(2008).

UCS745: CLOUD & DEVOPS

L	T	P	Cr
2	0	2	3.0

Course Objectives: The objective of the course is to teach techniques to automate the process of integration and deployment software product.

Introduction to DevOps: Definition of DevOps, Challenges of traditional IT systems &processes, History and emergence of DevOps, DevOps definition and principles governing DevOps, DevOps and Agile, The need for buildinga business use case for DevOps, Purpose of DevOps, Application Deployment, Automated Application Deployment, Application Release Automation (ARA), Components of Application Release Automation (ARA), Continuous Integration, Best Practices of CI, Benefits of CI, Continuous Delivery

Typical Toolkit for DevOps: DevOps, An Overview, Achieving DevOps, Continuous Practices, Continuous Integration (CI), How does CI Work? Continuous Integration Practices, Benefits of Continuous Integration A Quick Recap of Continuous Delivery, Continuous Delivery Process, Benefits of Continuous Delivery, Continuous Deployment

Source Code Management: History of Version Control Systems (VCS), Basic operations in a VCS, Examples of version control systems, Subversion (SVN), Features and Limitations, Mercurial, Git, Overview, History - Linux and Git by Linus Torvalds, Advantages of Git, Explain how local version control works, Centralized Version Control Systems (CVCS), Distributed Version Control Systems (DVCS), advantages of DVCS, Private Workspace, Easier merging, Easy to scale horizontally, List the disadvantages of DVCS, Explain how CVCS and DVCS compare with each other, Describe the working of the multiple repositories model Unit IV Application Containerization Understanding Containers: Transporting Goods Analogy, Problems in Shipping Industry before Containers, Shipping Industry Challenges, Container: Virtualization Introduction, Hypervisor, Scope of Virtualizations, Containers vs Virtual Machines, Understanding Containers, Containerizations Platform, Runtime and Images, Container Platform, Container Runtime, The Chroot System, FreeBSD Jails, LinuX Containers (LXC), Docker

Introduction to Containerization: Docker architecture, Docker Daemon (Container Platform), Docker Rest API, CLIDifferent environments: (Dev, QA and Prod), Overcoming issues with different environments, Development Environment Docker Swarm and Kubernetes, Architecture, AWS (ECS, EKS), AWS Elastic Container Services Architecture, Azure Kubernetes Services, Openshift, Kubernetes on cloud, Monitoring of container

Green Cloud Computing : Concept of Green Cloud, Virtualization in Green Cloud, Strategies for Sustainability in Green Cloud .

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Understand the benefits of DevOps over other software development processes.
2. Understand the phases of software development-delivery pipeline and automation benefits.
3. Identify and apply continuous integration and deployment prerequisites, process and benefits.
4. Understand and implement the continuous delivery engineering practices and release process.
5. Identify & use the test-driven deployment and various tools/frameworks used for continuous integration and delivery in DevOps.
6. Demonstrate the different DevOps Tools like Git, Docker, and Kubernetes etc.

Text Books:

1. Arundel, J., & Domingus, J., Cloud Native DevOps with Kubernetes: building, deploying, and scaling modern applications in the cloud. O'Reilly Media, (2019).
2. Kim, G., Humble, J., Debois, P., & Willis, J., The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT revolution Press 1st ed(2016).
3. Bass, L., Weber, I., & Zhu, L., DevOps: A software architect's perspective. Addison-Wesley Professional (2015).

Reference Books:

1. Fox, A., Patterson, D. and Joseph, S., Engineering Software as a Service: An Agile Approach Using Cloud Computing, 1st Edition (2013).
2. Rossel, S., Continuous Integration, Delivery, and Deployment: Reliable and Faster Software Releases with Automating Builds, Tests, and Deployment. Packt Publishing, 1st ed (2017).

3. EFB Conversational AI

UCS551: Conversational AI: Accelerated Data Science

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course will provide students with fundamental and advanced methods of data science: from Data Collection to Analytics and Machine Learning with RAPIDS on Text and Graphical problems.

Introduction: Fundamentals of Data Science, GPU Acceleration, RAPIDS Framework

Data Collection: Collecting Data, Scraping Data, Popular Scraping libraries, Data Annotation

Data Pre-processing (ETL): Introduction to Data-preprocessing, Data Cleaning & Statistical Preprocessing, Data Cleaners: OpenRefine and Wrangler, Feature Selection: Introduction to Filter Methods, Introduction to Model- based methods, Feature Reduction: PCA.

Introduction to Machine Learning - Supervised: Introduction to Supervised Learning, Linear Model, RAPIDS acceleration: Linear Regression, Overfitting and Cross Validation, Decision Tree, Visualizing Classification: {ROC, AUC, Confusion Matrix}, Bagging, Random Forests, RAPIDS Acceleration: Random Forest, Boosting, RAPIDS acceleration: K-NN, XGBoost.

Introduction to Machine Learning - Unsupervised: Introduction to Unsupervised Learning, Kmeans & Hierarchical Clustering, RAPIDS acceleration: K-Means, DBSCAN, PCA, t-SNE, UMAP, Visualizing Clusters, RAPIDS acceleration: PCA [t-SNE], UMAP, DBSCAN.

Graph Analytics: How to Represent & Store Graphs, Graph Power Laws, Centralities: Degree, Betweenness, Clustering Coefficient, PageRank & Personalized PageRank, Interactive Graph Exploration, RAPIDS Acceleration: Graphistry & cuXFilter.

Introduction to Deep Learning: Introduction to Artificial Neural Networks (ANNs), Artificial Neurons, Layers, Perceptron, Multilayer Perceptron, Advanced Deep Neural Networks (DNNs), Batch Normalization, Hyperparameter tuning, Activation Functions, Metrics, Optimization, Regularization.

Laboratory Work

1. Introduction to Dockers & Containers, Introduction to NVIDIA GPU Cloud (NGC).
2. Practical on Traditional Data Science packages (Numpy, Pandas, Scipy, Scikit-Learn).
3. Accelerated Data Science framework RAPIDS: Introduction to RAPIDS and cuDF.
4. Data Collection via API/Web Scraping.
5. Decision Tree Classification Clustering in RAPIDS.
6. Random Forest Classification in RAPIDS.
7. KMeans Clustering Implementation in RAPIDS.
8. Dimensionality Reduction and Visualization in RAPIDS.

9. Graph Analytics with cuGraph.
10. Latent semantic indexing for text via singular value decomposition(cuML).
11. Accelerating Workloads using RAPIDS
12. Introduction to DL Frameworks: PyTorch, and Tensorflow (Keras)

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. In depth understanding of Data Analytics, Pre-processing and Visualization Toolkits.
2. Comprehend and apply different classification and clustering techniques.
3. Understand GPU computing for building advanced data science applications.
4. Understand the concept of Neural Networks and its implementation using deep learning frameworks.

Text Books:

1. Mitchell M., T., Machine Learning, McGraw Hill (1997) 1st Edition.
2. Alpaydin E., Introduction to Machine Learning, MIT Press (2014) 3rd Edition.
3. Vijayvargia Abhishek, Machine Learning with Python, BPB Publication (2018).

Reference Books:

1. Bishop M., C., Pattern Recognition and Machine Learning, Springer-Verlag (2011) 2nd Edition.
2. Michie D., Spiegelhalter J. D., Taylor C. C., Campbell, J., Machine Learning, Neural and Statistical Classification. Overseas Press (1994).

UCS664: Conversational AI: Natural Language Processing

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course provides a broad introduction to deep learning and natural language processing. It offers some of the most cost-effective approaches to automated knowledge acquisition in the emerging field of natural language understanding using deep learning and GPU Computing.

Introduction: Natural Language Processing and its Applications, Introduction to Deep Learning, NVIDIA Toolkits, SDKs and platforms for Deployment.

Introduction to Natural Language Processing: What is NLP, Principles and Traditional Methods, Linguistics, Why Machine Learning and Deep Learning.

Introduction to NLP using Deep Learning: Embeddings: Feedforward NN, Word2Vec, GloVE, Contextualization (ELMo etc.), Deep Recurrent Models: RNNs, GRUs, LSTMs.

Advanced NLP using Deep Learning: Introduction to NeMo, Self-Attention, Transformer Networks: BERT and its Variants, Megatron etc, Working with open source datasets: GLUE Benchmarks.

Applications of NLP: Exploring NLP Problem Statements- Information Retrieval, Intent Slot Filling, Machine Translation, Punctuation & Capitalization, Question and Answering Machine Machine, Relation Extraction, Sentiment Analysis, Token Classification in NeMo.

Introduction to NVIDIA Toolkits and SDKs: Transfer Learning Toolkit, TensorRT Optimization, Triton Inference Server for Inferencing and Deployments, Various Visualization Tools, Kubernetes Deployment.

Laboratory Work:

- Introduction to DL Frameworks: TLT, PyTorch, and Tensorflow (Keras).
- Binary Classification with Perceptron and Logistic Regression.
- Neural Modules (NeMo) for Training Conv AI Models, Exploring NeMo Fundamentals, Exploring NeMo Model Construction, Nemo Swap App Demo.
- Sentiment Analysis & Text Classification with NeMo.
- Intent Slot Filling for ChatBot using Joint Bert Model with NeMo.
- Machine Translation with NeMo.
- Question & Answering Machine with NeMo.
- Information Retrieval, Punctuation & Capitalization, Relation Extraction, Sentiment Analysis, Token Classification with NeMo.
- Hands-on practical on TensorRT Optimization, Triton Inference Server.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Understand the concept of Neural Networks and its implementation in the context of Machine Learning.

2. Comprehend and apply recurrent neural networks on various NLP applications.
3. Understand the concept of basic transformer networks and its variants.
4. Apply transformer-based networks and its variants for NLP applications like text classification, question-answering and machine translation systems.

Text Books:

1. Schmidhuber, J. (2015). ―Deep Learning in Neural Networks: An Overview". Neural Networks 61: 85-117.
2. Bengio, Y., LeCun, Y., and Hinton, G. (2015). ―Deep Learning". Nature 521: 436-44.
3. Allen, James, Natural Language Understanding, *Second Edition, Benjamin/Cumming, 1995.*
4. Bengio, Y., Courville, A., and Vincent, P. (2013). ―Representation learning: A review and new perspectives", IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (8): 1798-1828.
5. Deep Natural Language Processing course offered at the University of Oxford: https://github.com/oxford-cs-deepnlp-2017/lectures
6. "The Unreasonable Effectiveness of Recurrent Neural Networks" by Andrej Karpathy: https://karpathy.github.io/2015/05/21/rnn-effectiveness/ Manning, Christopher and Heinrich, Schutze, Foundations of Statistical Natural Language Processing, *MIT Press, 1999.*

UCS749: Conversational AI: Speech Processing & Synthesis

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course will provide students with the overall structure of the Conversational AI pipeline including Speech Processing, Recognition, and Synthesis and building end to end workflows using NeMo and Jarvis SDK.

Introduction: Fundamentals of Speech Processing, Applications of Speech Processing and Deploying NLP, ASR and TTS modules in Jarvis.

Fundamentals of Speech Processing: Introduction to Statistical Speech Processing, HMMs for Acoustic Modeling, WFTS for Automatic Speech Recognition (ASR), Basics of Speech Production, Tied State HMMs, Introduction to NNs in Acoustic Modeling (Hybrid/TDNN/Tandem).
{Papers}

Automatics Speech Recognition (ASR): ASR - DNN models (Jasper, QuartzNet, Citrinet, Conformer-CTC), Open-source Datasets, Language Modelling: N-Gram, Neural Rescoring.
{Survey , Jasper, QuartzNet, CitriNet , Nemo}

Applications of Speech Processing: Speech Commands: Speech Commands Recognition using MatchboxNet. Overview of Noise Augmentation, Voice Activity Recognition and Speaker Recognition.
{Survey, Nemo}

Speech Synthesis: Text Normalization: Preparing Dataset and Text Normalization for input to Speech Synthesis model. Introduction to Text-to-Speech (TTS) Models:- Mel Spectrogram Generator: - Tacotron-2, Glow-TTS, Audio Generators:- WaveGlow, SqueezeWave.
{Papers, Nemo}

Jarvis Deployment: Introduction to Jarvis, Overview of Jarvis ASR, NLU and TTS APIs, Introduction to Jarvis Dialog Manager. Jarvis Deployment:- Nemo model deployment for ASR, NLP and TTS.

Laboratory Work:

- Practical Exercise on Statistical Speech Processing. {Traditional Signal Processing}
- Automatic Speech recognition with NeMo on English Dataset.
- Automatic Speech recognition with NeMo on Indic Language(Hindi) Dataset.
- NeMo Speech Commands Recognition using MatchboxNet, Noise Augmentation, and Speaker Recognition.
- Text to Speech using Tacotron-2 and WaveGlow with NeMo on English Dataset.
- Text to Speech using Tacotron-2 and WaveGlow with NeMo on Indic Language (Hindi) Dataset.
- End-to-End Conversational AI Model (Any Language): ASR/NLP/TTS with NeMo and Jarvis.

Course Learning Outcomes (CLOs) / Course Objectives (COs): After the completion of the course the student will be able to:

1. Understand Speech Processing pipeline for various applications with accelerated

computing. ASR, Speaker Recognition etc.

2. Exploring Speech Synthesis on various data sets.
3. Deep practical hands-on experience from training to deployment of these applications using NVIDIA GPUs and Toolkits:- NeMo, and Jarvis.

Text Books:

1. Jurafsky, Dan and Martin, James, Speech and Language Processing, *Second Edition, Prentice Hall, 2008.*
2. Daniel Jurafsky and James H. Martin, "Speech and Language Processing", 3rd edition draft, 2019 [JM-2019].
3. Mark Gales and Steve Young, The application of hidden Markov models in speech recognition, Foundations and Trends in Signal Processing, 1(3):195-304, 2008.

Reference Books:

1. Daniel Jurafsky and James H. Martin. 2009. Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 2nd edition. Prentice-Hall.
2. Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury, Deep Neural Networks for Acoustic Modeling in Speech Recognition, IEEE Signal Processing Magazine, 29(6):82-97, 2012.

UCS748: GENERATIVE AI

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course introduces students to the field of generative artificial intelligence with a focus on Large Language Models (LLMs). Students will learn the theoretical foundations behind LLMs and gain hands-on experience in training and fine-tuning these models for various generative tasks such as text generation, image generation, and more.

Introduction to Generative AI: Generative AI: Meaning, Capabilities and Potential, Applications of Generative AI, Tools for text, images, videos, audios and code generation.

Generative AI Models: Introduction to Large Language Models (LLMs), History and evolution of LLMs, Transformer architecture: Attention mechanism, Pre-training and fine-tuning of LLMs, Language modeling objectives (e.g., Masked Language Modeling, Next Sentence Prediction), Data preprocessing for LLMs, Training strategies and best practices, Fine-tuning on custom datasets, Handling domain-specific data and tasks.

Text Generation with LLMs: Text generation techniques**,** Conditional text generation**,** Sampling strategies**,** Evaluation metrics for text generation

Image Generation with LLMs: Overview of image generation tasks**,** Generative Adversarial Networks (GANs) vs. LLMs for image generation**,** Fine-tuning LLMs for image generation**,** Evaluation metrics for image generation

Beyond Text and Images: Multi-Modal Generation: Introduction to multi-modal generation**,** Combining LLMs with other generative models**,** Applications of multi-modal generation.

Prompt Engineering: Meaning of Prompts and Prompt Engineering, Text-to-text prompt techniques: Interview Pattern Approach, Chain-of-Thought Approach, Tree-of-Thought Approach.

Laboratory Work:

1. Pre-train a small language model on a text corpus and fine-tune it on a specific task or dataset

2. Fine-tune pre-trained language models on custom datasets for specific tasks like sentiment analysis or text classification.

3. Implement autoregressive decoding to generate text using a pre-trained language model and explore its limitations.

4. Build a conditional text generation model that takes input prompts or contexts to generate relevant responses.

5. Build a basic GAN architecture using a deep learning framework like TensorFlow or PyTorch.

6. Explore techniques to fine-tune pre-trained language models like GPT for image generation tasks using frameworks like CLIP.

7. Experiment with combining LLMs like GPT with other generative models like GANs or Variational Autoencoders (VAEs) for multi-modal generation tasks.

8. Implement a multi-modal generation model that generates coherent captions for given images or generates images from textual descriptions.

9. Explore real-world applications of multi-modal generation such as image captioning, visual question answering (VQA), and generating visual explanations from textual input.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
After the completion of the course the student will be able to:

1. Understand the theoretical foundations of Large Language Models (LLMs).
2. Apply LLMs for various generative tasks including text generation, image generation, and more.
3. Train and fine-tune LLMs on custom datasets.
4. Explore advanced topics and applications of LLMs in research and industry.

Text Books:

1. "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
2. "Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play" by David Foster.
3. "Dive into Deep Learning" by Aston Zhang.

Reference Books:

1. "Attention is All You Need" by Ashish Vaswani et al.
2. "Natural Language Processing with PyTorch" by Delip Rao and Brian McMahan
3. "GPT-3: Language Models are Few-Shot Learners" by Brown et al.

EFB

Robotics and Edge AI

UCS547: Edge AI and Robotics: Accelerated Data Science

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course will provide students with fundamental knowledge of GPU Computing and Machine Learning and Deep Learning Primer.

Introduction: GPU Computing, Parallel Programming, Machine Learning with RAPIDS, Model Compression

Image Processing and Parallel Programming: GPU Programming, CUDA C/C++/Python, Accelerated Image Processing, nvJPEG, Numba.

Introduction to Machine Learning - Supervised: Introduction to Supervised Learning, Linear Model, RAPIDS acceleration: Linear Regression, Overfitting and Cross Validation, Decision Tree, Visualizing Classification: {ROC, AUC, Confusion Matrix}, Bagging, Random Forests, RAPIDS Acceleration: Random Forest, Boosting, RAPIDS acceleration: K-NN, XGBoost.

Deep Learning Model Compression: Introduction to model pruning, quantization and distillation

Optimization Framework: Using TensorRT optimization, Deploying model on Triton Inference server

Laboratory Work:

- Practical on Traditional Data Science packages (Numpy, Pandas, Scipy, Scikit-Learn).
- CUDA C/C++ for Accelerated Computing.
  {DLI Online Course Section: Fundamentals of Accelerated Computing with CUDA C/C++}
- Numba to compile CUDA kernels for Numpy Acceleration in Python.
  {DLI Online Course Section: Fundamentals of Accelerated Computing with CUDA Python}
- Getting started with Accelerated Data Science with RAPIDS AI (cuPy, cuDF, cuSignal, cuML).
- Decision Tree Classification Clustering in RAPIDS.
- Random Forest Classification in RAPIDS.
  {DLI Online Course Section: Fundamentals of Accelerated Data Science with RAPIDS, Section 2: GPU-accelerated Machine Learning}
  Model Pruning, Post Training Quantization, Quantization Aware Training, Distillation

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Demonstrate ability to deploy GPU accelerated code for image processing applications .
2. Apply model compression techniques while deploying deep learning architectures. .
3. Implement Rapids AI framework for different machine learning tasks. .

4. Analyze and evaluate performance of deep learning based inference models using TensorRT optimization and Triton Server .

Text Books:

1. Mitchell M., T., Machine Learning, McGraw Hill (1997) 1st Edition.
2. Alpaydin E., Introduction to Machine Learning, MIT Press (2014) 3rd Edition.
3. Vijayvargia Abhishek, Machine Learning with Python, BPB Publication (2018).

Reference Books:

1. Bishop M., C., Pattern Recognition and Machine Learning, Springer-Verlag (2011) 2nd Edition.
2. Michie D., Spiegelhalter J. D., Taylor C. C., Campbell, J., Machine Learning, Neural and Statistical Classification. Overseas Press (1994).

UCS668: Edge AI and Robotics: Data Centre Vision

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course will provide students with basic fundamental understanding and practical hands-on training of computer vision and deep learning models on data centre GPU servers.

Introduction: Introduction to Deep Learning, Formulating Computer Vision Problem Statements, Image Classification using CNN Architectures like VGG, Inception, ResNet(18/34/50/152). Working towards building Object detection and Segmentation pipelines

Introduction to Deep Learning: Introduction Advanced Deep Neural Networks (DNNs), Batch Normalization, Hyperparameter tuning, Activation Functions, Metrics, Optimization, Regularization.

Applications of Computer Vision (Image Classification): Introduction to NVIDIA Frameworks: {Transfer Learning using Transfer Learning Toolkit (TLT), Mixed Precision, DALI}, Image Classification using Deep CNN Architecture like VGG, ResNet18/34/50, re-training on custom dataset.

Applications of Computer Vision (Object Detection & Segmentation): Introduction to Object Detection, Data Preprocessing, CNN Architecture like {SSD, YOLOv3,EfficientDet, Spinenet}, Metrics, Loss Functions, re-training on custom dataset, Segmentation: FCN-ResNet, Unet, MaskRCNN, Metrics and Loss functions.

Graph Neural Network and Synthetic Data Generation: **:** Introduction to Graph Neural Networks, Omniverse Replicator based synthetic data generation using 3D assets.

.

Laboratory Work:

- Introduction to DL Frameworks: TLT, PyTorch, and Tensorflow (Keras).
  {DLI Online Course: Getting Started with Deep Learning}
  {DLI Online Course: Deep Learning at Scale with Horovod}
- Training Classification Models with and without Mixed Precision and Multi-GPU on Open & Custom Datasets.
- Training Detection Models with and without Mixed Precision and Multi-GPU on Open & Custom Datasets.
- Training Segmentation Models with and without Mixed Precision and Multi-GPU on Open & Custom Datasets.
  {DLI Online Course: Getting Started with Image Segmentation}
  {DLI Online Course: Synthetic Data Generation for Training Computer Vision Models}
  {DLI Online Course: Introduction to Graph Neural Networks}

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Apply computer vision based techniques for problem solving in various domains .
2. Demonstrate deployment of deep learning models, trained with mixed precision. .
3. Analyze and evaluate performance of models for classification, detection and segmentation tasks.
4. Implement synthetic image generation using generative models

Text Books:

1. Mitchell M., T., Machine Learning, McGraw Hill (1997) 1st Edition.
2. Alpaydin E., Introduction to Machine Learning, MIT Press (2014) 3rd Edition.
3. Vijayvargia Abhishek, Machine Learning with Python, BPB Publication (2018).
4. Computer Vision: Algorithms and Applications, R. Szeliski, Springer, 2011.
5. Computer Vision: A Modern Approach, D. Forsyth and J. Ponce, Prentice Hall, 2nd ed., 2011.

Reference Books:

1. Bishop M. C., Pattern Recognition and Machine Learning, Springer-Verlag (2011) 2nd Edition.
2. Introductory techniques for 3D computer vision, E. Trucco and A. Verri, Prentice Hall, 1998.
3. "Visualizing and Understanding Convolutional Networks" by Matthew D. Zeiler and Rob Fergus (2014)
4. "Convolutional Neural Networks for Visual Recognition" (Stanford course given by Fei-Fei Li, Andrej Karpathy, and Justin Johnson, 2016): http://cs231n.github.io/

UCS671: Edge AI and Robotics: Embedded Vision

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course will provide students with advanced conceptual knowledge and practicals on various computer vision and deep learning applications and provide the overall environment for end-to-end pipeline development from data collection to deployment.

Introduction: Utilizing Jetpack SDK and other NVIDIA Toolkits to deploy CNN models on Jetson, Creating Jetbot kits and deploying various applications, Working with NVIDIA Robotics toolkit: Isaac SIM SDK and Gazebo for collision avoidance, path following.

Introduction to Edge AI: AI at the Edge & IoT, Jetson Architecture, Getting Started with Jetpack, NGC Containers in Jetson, Getting started with NGC & Containers on Jetson.

Introduction to NVIDIA Toolkits and SDKs: Transfer Learning Toolkit, Kubernetes Deployment, Deepstream SDK, Deploying Classification, Detection and Segmentation CNN Models on Jetson Devices.

Perception & Autonomous Navigation: Building JetBot Kits, Introduction to basic motion on JetBot, Collision Avoidance: Stop/Go classifier (JetBot), freespace detection, Path Following: Recording user input/video + DNN training (DriveNet), Simulation: Gazebo & Isaac SIM.

Advanced Vision & SLAM: Pose Recognition (Deploying Human pose model), Depth Estimation: Mono/Stereo depth and point extraction, Visual Odometry: Camera pose estimation from DNNs, SLAM on JetBot.

Laboratory Work:

- Setting up the Jetson Project kit.
  {DLI Online Course: Getting Started with AI on Jetson Nano.}
- Deployment of Various Classification, Object Detection and Segmentation models in Jetson Nano.
- Getting started building various Jetbot Kits.
- Basic Motion with Jetbot
- Collision Avoidance with Jetbot kit
- Object following and Road following (DriveNet) with Jetbot.
- Teleoperation with Jetbot.
- Human Pose Estimation in Jetson Nano/JetBot.
- Implementing SLAM on Jetbot.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Demonstrate deployment of realtime vision based applications on edge devices
2. Implement kubernetes framework using Jetson Nano Devices. .
3. Apply vision based pose estimation techniques.

4. Implement SLAM using Jetbot kits and vision sensors.

Text Books:

1. Computer Vision: Algorithms and Applications, R. Szeliski, Springer, 2011.
2. Computer Vision: A Modern Approach, D. Forsyth and J. Ponce, Prentice Hall, 2nd ed., 2011.
3. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.

Reference Books:

1. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004.
2. K. Fukunaga; Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990.
3. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison- Wesley, 1992.
4. Christopher M. Bishop; Pattern Recognition and Machine Learning, Springer, 2006.

UCS760: Edge AI and Robotics: Reinforcement Learning & Conversational AI

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course will provide students with introduction to the basic mathematical foundations of Reinforcement Learning for building real world computer vision applications, and Conversational AI for developing Chatbots.

Introduction: GPU Computing, Implementing Behaviours of Robots such as Manipulation, and Task Learning, Fundamentals of Reinforcement Learning for Vision and Deploying Conversational AI pipelines in Jetson.

Manipulation: Overview of Manipulation in Robotics, Inverse Kinematics and Control, Gripping & Task Learning.

Reinforcement Learning: Introduction to RL: RL agents, Dynamic Programming, Monte Carlo's and Temporal-Difference Methods, OpenAI Gym, RL in Continuous Spaces.
{Added Lectures, Summaries}

Conversational AI (NLP): Natural Language Processing: Introduction to NLP, BERT, Megatron, Applications of NLP: Information Retrieval, Intent Slot Filling, Machine Translation, Punctuation & Capitalization, Question and Answering Machine Machine, Relation Extraction, Sentiment Analysis, Token Classification in NeMo.

Conversational AI (Speech Processing): Automated Speech Recognition: Introduction to ASR, Architectures: Jasper/QuartzNet/CitriNet, Text to Speech: TTS-Tacotron2/WaveGlow and Jarvis Deployment.

Laboratory Work:

- Manipulation Lab: Building Pick-n-place.
- Manipulation Lab: Object Assembly.
- Game Agent: Open AI Gym (Jetbot in simulation).
- Conversational AI VoiceBot: Verbal JetBot commands/feedback, ect (optional mic/speaker).

Course Learning Outcomes (CLOs) / Course Objectives (COs):

After the completion of the course the student will be able to:

1. Apply reinforcement learning for robot navigation.
2. Deploy computer vision models in various simulation environments
3. Implement applications with conversational AI
4. Utilize programming and AI training & deployment tools for relevant model building in both edge hardware devices and simulation environments.

Text Books:

1. Wiering, Marco, and Martijn Van Otterlo. "Reinforcement learning." Adaptation, learning, and optimization 12 (2012): 3.
2. Russell, Stuart J., and Peter Norvig. "Artificial intelligence: a modern approach."Pearson Education Limited, 2016.
3. Jurafsky, Dan and Martin, James, Speech and Language Processing, Second Edition, Prentice Hall, 2008.
4. Daniel Jurafsky and James H. Martin, "Speech and Language Processing", 3rd edition draft, 2019 [JM-2019].

Reference Books:

1. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. "Deep learning." MIT press, 2016.
2. Mark Gales and Steve Young, The application of hidden Markov models in speech recognition, Foundations and Trends in Signal Processing, 1(3):195-304, 2008.
3. Daniel Jurafsky and James H. Martin. 2009. Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 2nd edition. Prentice-Hall.
4. "Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto: https://webdocs.cs.ualberta.ca/~sutton/book/the-book-2nd.html
5. David Silver's course: http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
6. "Deep Reinforcement Learning: Pong from Pixels" by Andrej Karpathy: https://karpathy.github.io/2016/05/31/rl/
7. Talks on Deep Reinforcement Learning by John Schulman: https://www.youtube.com/watch?v=aUrX-rP_ss4 , and his Deep Reinforcement Learning course http://rll.berkeley.edu/deeprlcourse/

EFB

Cyber Forensics and Ethical Hacking

UCS550: NETWORK DEFENCE

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course is designed to provide the fundamental skills needed to analyse the internal and external security threats against a network, and to implement security mechanisms to protect an organization's information. The course helps to evaluate network and Internet security issues and provides security solutions such as designing a security policy, troubleshooting networks, and digital signatures.

Network Attacks and Defence Strategies: Essential terminologies related to network security attacks ▪ Various examples of network-level attack techniques ▪ Various examples of application-level attack techniques ▪ Various examples of social engineering attack techniques ▪ Various examples of email attack techniques ▪ Study attack vectors targeting IoT devices in smart cities and their impact on urban sustainability

Administrative Network Security: Obtain Regulatory Frameworks Compliance ▪ Discuss Various Regulatory Frameworks, Laws, and Acts ▪ Learn to Design and Develop Security Policies ▪ Conduct Security Awareness Training ▪ Explore regulatory frameworks for securing renewable energy systems and smart grids.

Technical Network Security: The principles of access control, the terminologies used, and the different models, The different aspects of IAM such as identity management, authentication, authorization, and accounting ▪ The various cryptographic security techniques ▪ The various cryptographic algorithms ▪ The security benefits of network segmentation techniques ▪ The various essential network security solutions

Network Perimeter Security: Understand firewall security concerns, capabilities, and limitations ▪ Understand different types of firewall technologies and their usage ▪ Understand firewall topologies and their usage ▪ Distinguish between hardware, software, host, network, internal, and external firewalls

Endpoint Security-Windows Systems: Windows OS and security concerns ▪ Windows security components ▪ Windows security features ▪ Windows security baseline configurations ▪ Windows user account and password management

Endpoint Security-Linux Systems: Understand Linux OS and security concerns ▪ Discuss Linux installation and patching ▪ Discuss Linux OS hardening techniques ▪ Discuss Linux user access and password management

Endpoint Security-Mobile Devices: Common mobile usage policies in enterprises ▪ Security risks and challenges associated with enterprise mobile usage policies ▪ Security guidelines to mitigate the risks associated with enterprise mobile usage policies

Endpoint Security IoT-Devices: Understand IoT devices, their need, and application areas ▪ Understand the IoT ecosystem and communication models ▪ Understand security challenges and risks associated with IoT-enabled environments ▪ Discuss security in IoT-enabled environments ▪ Implement energy-efficient communication protocols for IoT security.

Administrative Application Security: Implement application whitelisting and blacklisting ▪ Implement application sandboxing ▪ Implement application patch management

Data Security: Understand data security and its importance, Discuss the implementation of data access controls ▪ Discuss the implementation of ―data at rest‖ encryption ▪ Discuss the implementation of ―data in transit‖ encryption

Introduction to Ethical Hacking: Describe various hacking methodologies and frameworks ▪ Describe hacking concepts and hacker classes ▪ Explain ethical hacking concepts and scope

Footprinting and Reconnaissance: Describe footprinting concepts ▪ Perform footprinting through search engines and using advanced Google hacking techniques ▪ Perform footprinting through web services and social networking sites

Scanning Networks: Describe the network scanning concepts ▪ Use various scanning tools ▪ Perform host discovery to check for live systems ▪ Perform port and service discovery using various scanning techniques

Enumeration: Describe enumeration concepts ▪ Explain different techniques for NetBIOS enumeration ▪ Explain different techniques for SNMP enumeration

Vulnerability Analysis: Understand vulnerability, vulnerability research, vulnerability assessment, and vulnerability scoring systems ▪ Describe the vulnerability management life cycle (vulnerability assessment phases) ▪ Understand various types of vulnerabilities and vulnerability assessment techniques ▪ Analyze vulnerabilities in green technologies and recommend secure implementations.

Laboratory Work:

Learn the Workings of SQL Injection Attacks, Implement Password Policies Using Windows Group Policy, Just Enough Administration to Secure Privileged Access, External NetworkBased Firewall Functionality, Remote Patch Management using BatchPatch, Linux Security Auditing and System Hardening Using Lynis, Enterprise Mobile Security Using Miradore MDM Solution, Secure IoT Device Communication Using TLS/SSL, Application Whitelisting Using AppLocker, Encrypt Data at Rest Using VeraCrypt, Footprinting and Reconnaissance, Scanning Networks, Enumeration, Vulnerability Analysis.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

1. On completion of this course, the students will be able to:
2. 1. Implement Network security management, network security policies and procedures, data security techniques.
3. 2. Implement Optimum firewall, IDS/IPS configuration.
4. 3. Implement Windows, Linux and other endpoint security.
5. 4. Understand Mobile and IoT device security.
6. 5. Understand concepts of hacking
7. 6. Perform Scanning Networks

Text Books:

8. **1.** Certified Network Defender (CNDv2) Academia Series – EC- council
9. 2. Ethical Hacking and Countermeasures (CEHv12) Academia Series – EC-Council

UCS714: Agentic AI

L	T	P	Cr
2	0	2	3.0

Course Objective: The course aims to develop foundational understanding of Generative AI and Large Language Models, introduce the principles and architectures of Agentic AI systems, train students in prompt engineering for autonomous agents, enable the design and development of single and multi-agent systems, and equip learners with evaluation, optimization, and governance techniques for responsible deployment of agentic AI solutions.

Foundations of Generative and Agentic AI: Overview of Generative AI; fundamentals of Large Language Models and their capabilities and limitations; applications of Generative AI across domains; comparison of traditional AI and Agentic AI; components of intelligent agents including goals, memory, reasoning, and action; autonomy and feedback mechanisms in agent systems.

Prompt Engineering and Reasoning Strategies for Agents: Role prompting; zero-shot and few-shot prompting; Chain-of-Thought prompting; ReAct prompting; self-reflection prompting; system prompts versus user prompts in agent architectures; prompt structuring for multi-agent interactions; prompt injection risks and prompt hygiene practices.

Agent Architectures and Design Paradigms: Characteristics of intelligent agents including autonomy, reactivity, proactiveness, and social ability; reactive and deliberative agent models; reflection design pattern; tool-use pattern; planning agents and task decomposition; multi-agent collaboration architectures; architectural considerations for robust and scalable agentic systems.

Collaborative Multi-Agent Systems and Workflow Orchestration: Agent communication models including direct and mediated communication; structured and unstructured message passing; shared memory models such as blackboard systems; DAG-based workflow orchestration; recursive delegation and feedback-driven iteration; interoperability protocols including Model Context Protocol (MCP) and Agent-to-Agent (A2A) protocol; secure cross-agent communication mechanisms.

Deployment and Evaluation of Agentic AI: Workflow orchestration using low-code tools such as N8N; triggers, webhooks, and automation nodes; API integration, authentication, and secure token management; integration with databases and external systems; deployment of scalable multi-agent pipelines; error handling, fallback mechanisms, and retry strategies; performance metrics including task success rate, latency, reasoning depth, response coherence, and cost tracking; prompt optimization and versioning; debugging and monitoring of agent behavior.

Lab Experiments - Exploring LLM APIs and prompt testing, implementing goal-driven single agents, building reasoning-based agents, designing tool-using agents, implementing planning and reflection agents, multi-agent collaboration experiment, Building agent workflows with automation tools, API integration and deployment, Performance benchmarking and evaluation

Course Outcomes (COs)

At the end of the course, the student will be able to:

CO1: Explain the principles of Generative AI and Agentic AI systems.
CO2: Apply prompt engineering techniques for reasoning and autonomous behavior.
CO3: Design and implement agentic AI architectures using standard design patterns.
CO4: Develop end-to-end multi-agent pipelines integrating tools and APIs.
CO5: Evaluate agentic AI systems using performance metrics

Text Books

1. "Multi-Agent Systems: Algorithmic, Game-Theoretic, and Logical Foundations" – Yoav Shoham and Kevin Leyton-Brown.
2. Mastering Al Agents: From Design to Deployment of Autonomous Systems by Elbert Gale, ISBN-13: 979-8307403822
3. Building Al Agent: Learn Building AI Agents, A Practical Guide for Building Intelligent Agents by Natenapis Faraksa.

Reference Books

1. "Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back" – Pattie Maes.
2. Build Your Own Al Agent: A Hands-On Guide for Beginners by Sujai G, ISBN-13: 979-8306885971

UCS673: Ethical Hacking-I

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course is designed to provide the fundamental skills needed to analyse the internal and external security threats against a network, and to implement security mechanisms to protect an organization's information. The course helps to evaluate network and Internet security issues and provides security solutions such as designing a security policy, troubleshooting networks, and digital signatures. The course helps to understand and apply the basic hacking techniques.

Enterprise Virtual Network Security: Evolution of network and security management concepts in modern virtualized IT environments, Essential concepts in virtualization, Network virtualization (NV) security, Software-defined network (SDN) security, OS virtualization security, Explore energy-efficient virtual network configurations .

Enterprise Cloud Network Security: Cloud computing fundamentals, Cloud security, Evaluate the Cloud Service Providers (CSPs) for security before consuming a cloud service, Security in Amazon cloud (AWS)

Enterprise Wireless Network Security: Understand the fundamentals of wireless networks, Understand the encryption mechanisms used in wireless networks, Understand the authentication methods used in wireless networks, Study wireless network designs optimized for sustainability in urban areas.

Network Traffic Monitoring and Analysis: Understand the need for and advantages of network traffic monitoring, Setting up the environment for network monitoring, Determine baseline traffic signatures for normal and suspicious network traffic, Perform network monitoring and analysis for suspicious traffic using Wireshark, Study tools that optimize traffic flow to reduce energy consumption.

Network Logs Monitoring and Analysis: Logging concepts, Log monitoring and analysis on Windows systems, Log monitoring and analysis on Linux systems

Incident Response and Forensic Investigation: Understand the Concept of Incident Response, Understand the Role of the First Responder in Incident Response, Discuss Do's and Don'ts in First Response, Describe the Incident Handling and Response Process

Business Continuity and Disaster Recovery: Business Continuity (BC) and Disaster Recovery (DR), BC/DR activities, Business Continuity Plan (BCP) and Disaster Recovery Plan (DRP)

Risk Anticipation with Risk Management: Understand risk management concepts, Learn to manage risk through a risk management program, Learn different risk management frameworks (RMFs)

Threat Assessment with Attack Surface Analysis: Attack surface analysis, Attack surface, Identify Indicators of Exposures (IoEs), Evaluate attack surfaces in energy-efficient IT systems

Threat Prediction With Cyber Threat Intelligence: Understand the Role of Cyber Threat Intelligence (CTI) In Network Defense, Understand the Different Types of Threat Intelligence, Understand the Indicators of Threat Intelligence: Indicators of Compromise (IoCs) and Indicators of Attack (IoAs)

System Hacking: Explain the different techniques to gain access to a system, Apply privilege escalation techniques, Explain different techniques to gain and maintain remote access to a system, Describe different types of rootkits

Malware Threats: Describe the concepts of malware and malware propagation techniques, Explain Potentially unwanted applications (PUAs) and adware, Describe the concepts of advanced persistent threats (APTs) and their lifecycle

Sniffing: Describe sniffing concepts, Explain different MAC attacks, Explain different DHCP attacks, Describe ARP poisoning, Explain different spoofing attacks, Study sniffing attacks on energy-efficient wireless networks and IoT devices.

Social Engineering: Describe social engineering concepts, Perform social engineering using various techniques, Describe insider threats, Perform impersonation on social networking sites

Denial of Service: Describe DoS/DDoS concepts, Describe botnets, Understand various DoS/DDoS attack techniques, Explain different DoS/DDoS attack tools, Study DoS/DDoS attacks on smart city systems and energy-efficient infrastructures.

Laboratory Work:
Learn how to Audit Docker Host Security Using Docker-Bench-Security Tool, Learn how to Implement Amazon Web Services Identity and Access Management, Learn how to Configure Security on a Wireless Router, Learn how to Monitor and Detect Network Reconnaissance, Access and Denial-of-Service/Distributed Denial-of-Service Attempts, Learn how to Identify Suspicious Activities Using Log Monitoring and Analysis, Learn how to Work with Incident Tickets in OSSIM, Learn how to Perform Vulnerability Management using OSSIM, Learn how to Perform Vulnerability Analysis Using the Nessus, Learn how to identify an Attack Surface in Windows using the Microsoft Attack Surface Analyzer, System Hacking, Gain Access to the target system using Trojans, Perform Active Sniffing, Perform Social Engineering using various techniques.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
On completion of this course, the students will be able to:

1. Examine the evolution of network security management in virtualized IT and cloud environments.
2. Understand wireless networking concepts, logging concepts and advantages of network traffic monitoring.
3. Examine various network security incidents and develop policies, processes, and guidelines for incident handling, disaster recovery and business continuity.
4. Implement Network security management, network security policies and procedures, data security techniques.
5. Compare and contrast different hacking techniques and analyze the legal implications of hacking.
6. Examine different vulnerabilities, threats, and attacks to information systems and recommend the countermeasures.

Text Books:

1. Certified Network Defender (CNDv2) Academia Series – EC-Council
2. Ethical Hacking and Countermeasures (CEHv12) Academia Series – EC-Council

UCS674: Ethical Hacking-II

L	T	P	Cr
2	0	2	3.0

Course Objective: This undergraduate-level course provides students with an in-depth understanding of the principles, techniques, and tools used in ethical hacking. Ethical hacking, also known as penetration testing or white-hat hacking, involves the authorized and legal exploration of computer systems, networks, and applications to identify vulnerabilities and weaknesses before malicious hackers can exploit them. This course will provide the knowledge and skills necessary to conduct ethical hacking assessments, identify security vulnerabilities, and recommend appropriate remediation measures to enhance the security posture of organizations and protect against cyber threats. Further, this course will help to understand the ethical and legal considerations involved in ethical hacking activities, and will help students to develop careers in cybersecurity and information technology.

Ethical Hacking: Ethical hacking, Attack Vectors, Cyberspace and Criminal Behaviour, Clarification of Terms, Traditional Problems associated with Computer Crimes, Realms of Cyber world, brief history of the internet, contaminants and destruction of data, unauthorized access, Explore attack vectors targeting green IT infrastructures and smart city networks.

Intrusion in cyber world: computer intrusions, white-collar crimes, viruses and malicious code, virus attacks, pornography, software piracy, mail bombs, exploitation, stalking and obscenity in internet, Cyber psychology, Social Engineering.

Laws related to cybercrime: Basics of Law and Technology, Introduction to Indian Laws, Scope and Jurisprudence, Digital Signatures, possible crime scenarios, law coverage, data interchange, mobile communication development, smart card and expert systems, Study international regulations concerning cybercrimes in energy and sustainability domains.

Digital Forensics: Introduction to Digital forensics, Forensic software and handling, forensic hardware and handling, analysis and advanced tools, forensic technology and practices, fingerprint recognition, Audio-video evidence collection, Preservation and Forensic Analysis. Definition and types of cybercrimes, electronic evidence and handling, electronic media, collection, searching and storage of electronic media, introduction to internet crimes, hacking and cracking, credit card and ATM frauds, web technology, cryptography, emerging digital crimes and modules, Investigate digital forensic techniques for renewable energy systems and eco-friendly technologies.

Computer Forensics in Today's World: Computer Forensics in Today's World, Fundamentals of Computer Forensics, Cybercrimes and their Investigation Procedures, Digital Evidence, Forensic Readiness, Incident Response, and the Role of SOC (Security Operations Centre) in Computer Forensics, Identify the Roles and Responsibilities of a Forensic Investigator, Challenges Faced in Investigating Cybercrimes, Understanding Legal Compliance in Computer Forensics

Computer Forensics Investigation Process: Forensic Investigation Process and its Importance, Pre-investigation Phase, Understanding First Response, Understanding the Post- Investigation Phase

Hard Disks and File Systems: Different Types of Disk Drives and their Characteristics, Logical Structure of a Disk, Booting Process of Windows, Linux and Mac Operating Systems, Various File Systems of Windows, Linux and Mac Operating Systems, Examine File System Using Autopsy and The Sleuth Kit Tools, Understanding Storage Systems, Examine energy-efficient storage technologies and their role in sustainability.

Forensic Tools and their applications: Forensic Tools and Processing of Electronic Evidence, Introduction to Forensic Tools, Usage of Slack space, tools for Disk Imaging, Data Recovery, Vulnerability Assessment Tools, Encase and FTK tools, Anti Forensics and probable counters, retrieving information, process of computer forensics and digital investigations, processing of digital evidence, digital images, damaged SIM and data recovery, multimedia evidence, retrieving deleted data: desktops, laptops and mobiles, retrieving data from slack space, renamed file, compressed files, Explore advanced techniques for recovering data from energy-efficient and eco-friendly devices.

Laboratory Work:
The course will incorporate hands-on lab exercises, practical demonstrations, and real-world scenarios to reinforce theoretical concepts and develop practical skills. Students will also engage in ethical hacking projects, allowing them to apply their knowledge in simulated environments and gain valuable experience in conducting ethical hacking assessments.

Course Learning Outcomes (CLOs) / Course Objectives (COs):
On completion of this course, the students will be able to:
1. Understanding the basics of Ethical Hacking and its role in industry, society and information system.
2. Describe various types of securities and vulnerabilities.
3. Understanding of cyber forensic that can be useful in the process of extracting and analysis of digital evidences.
4. Identify, Interpret and Evaluate Laws, Government Regulations and International Legal Systems Pertinent to cyber forensic.
5. Demonstration of cyber forensic tools and their uses in preventing various types of system attacks.

Text Books:
1. Simpson T. M., Backman K., Corley J., Hands-On Ethical Hacking and Network Defense, Delmar Cengage Learning (2011) 2nd edition.
2. Fadia A. and Zacharia M., Network intrusion alert: an ethical hacking guide to intrusion detection, Boston, MA: Thomas Course Technology 3rd edition (2008

UCS750: Computer Hacking and Forensic Investigation

L	T	P	Cr
2	0	2	3.0

Course Objectives: This course focuses on ethical hacking and security practices that cover the latest security threats, advanced attack techniques, and real-time demonstrations of hacking methods, tools, and protective measures. The students will examine digital evidence from computers, networks, and mobile devices. The course helps to realize the importance of integrating forensic practices into different operations to investigate attacks and system anomalies.

Hacking Wireless & Mobile Networks: Understanding Wireless Hacking Methodology, Wireless Hacking Tools, Understanding mobile network Threats and Attacks, Overview of Mobile Device Management (MDM), Mobile Security Guidelines and Tools, Explore vulnerabilities in mobile networks used for managing sustainable city infrastructures.

IoT Hacking: Overview of IoT Concepts, Understanding IoT Attacks, Understanding IoT Hacking Methodology, IoT Hacking Tools, IoT Countermeasures, Analyze IoT security challenges in renewable energy systems and smart grids.

Forensics: Understand Volatile and Non-volatile Data in Linux, Analyze Filesystem Images Using the Sleuth Kit, Demonstrate Memory Forensics Using Volatility & PhotoRec, Network Forensic Readiness, Perform Incident Detection and Examination with SIEM Tools, Monitor and Detect Wireless Network Attacks

Investigating Web Attacks: Understand Web Application Forensics, Understand Web Server Logs, Understand the Functionality of Intrusion Detection System (IDS) and Web Application Firewall (WAF), Investigate Web Attacks on Windows-based Servers, Detect and Investigate Various Attacks on Web Applications, Study intrusion prevention for eco-friendly e-commerce platforms and web services.

Dark Web Forensics: Understand the Dark Web, Determine How to Identify the Traces of Tor Browser during Investigation, Perform Tor Browser Forensics

Database Forensics: Understand & Perform MSSQL Forensics, Understand Internal Architecture of MySQL and Structure of Data Directory, Understand Information Schema and List MySQL Utilities for Performing Forensic Analysis, Perform MySQL Forensics on WordPress Web Application Database Directory, Explore forensic techniques for smart city database anomalies.

Investigating Email Crimes: Understand Email Basics, Understand Email Crime Investigation and its Steps, U.S. Laws Against Email Crime, Study spam filters optimized for energy-efficient cloud systems.

Malware Forensics: Define Malware and Identify the Common Techniques Attackers Use to Spread Malware, Understand Malware Forensics Fundamentals and Recognize Types of Malware Analysis, Understand and Perform Static Analysis of Malware, Analyze Suspicious Word and PDF Documents, Understand Dynamic Malware Analysis Fundamentals and Approaches, Analyze Malware Behavior on System Properties in Real-time

Laboratory Work: Learn about SQL Injection, Hacking Wireless Networks, Hacking Mobile Platforms, IoT and OT Hacking, Cloud Computing, Cryptography, Linux and Mac Forensics, Network Forensics, Investigating Web Attacks, Dark Web Forensics, Database forensics, Cloud Forensics, Investigating Email Crimes, Malware Forensics, Mobile Forensics, IoT Forensics

Course Learning Outcomes (CLOs) / Course Objectives (COs):

On completion of this course, the students will be able to:

1. Understand hacking of wireless networks and mobile platforms.
2. Implement various IoT hacking mechanisms, cloud computing hacking, and cryptography techniques.
3. Analyze forensics for operating systems (Linux and Mac).
4. Perform forensics on network, web and databases.
5. Investigate Email crimes, malware forensics, mobile forensics, and IoT forensics.

Text Books:

1. Certified Ethical Hacker (CEHv12), 12th Edition, EC-Council.
2. Computer Hacking Forensic Investigator (CHFIv11), 11th Edition, EC-Council.

UEC646: Network and Communication for Connected Vehicles

L	T	P	Cr
3	0	0	3.0

Course Objectives: To attain comprehensive overview of connected vehicles. To understand the intra and inter vehicle communication. Acquiring knowledge on Vanet. Able to tackle current problems in connected vehicle technology

Introduction to Connected Vehicles: Basics of cyber physical system- Architecture of connected cars. Vehicle sensor with on-board platform.

Intra and Inter vehicle network: Topology of LIN- CAN- Flex ray- WIFI and GPS- Automotive Ethernet- GPRS and 5G application for automotive systems communications- SOME-IP and DO-IP

Network architectures: Overview of the Architecture- channel models- properties of vehicle to vehicle and vehicle to infrastructure communication- performance of 802.11 in V2X.

VANET: Vehicular AD hoc network operations- single-hop broadcasting and multi-hop broadcasting- mobile IP solution in VANET.

V2V application: Emergency vehicle approach, Blind spot lane change warning, Rear end collision warning, Emergency brake light and Intersection movement assist.

V2I application: Traffic information system- Curve speed warning system- work zone warning system- weather impact warning system and Platoon detection.

Cyber security Risk management and Standards: Secure Vehicular Communication Architecture- Secure and Privacy-enhancing Vehicular Communication- Data Trustworthiness- Future challenges. SECOC for CAN protocol- MACSEC and TLS for Ethernet.

Contemporary Issues: Multi-Radio interfaces – Software Re-configurability for hardware update.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

On completion of this course, the students will be able to:

1. Understand the fundaments of connected vehicles.
2. Comprehend on intra and inter vehicle network.
3. Network architecture for connected vehicles.
4. Understand the VANET networking.
5. Application of V2V and V2I.
6. Security and privacy of vehicle network.

Text Books:

1. Christoph Sommer, Falko Dressler, Vehicular Networking, Cambridge University Press, 2015.
2. Markus Meuck, Ingolf Karls, Networking Vehicles to Everything, DE Gruyter press, 2018
3. Dominique Paret, Hassina Rebaine "Autonomous and Connected Vehicles Network Architectures from Legacy Networks to Automotive Ethernet" 2022 Wiley

Reference books:

1. Peter Han Joo Chong "Vehicular Networks Applications, Performance Analysis and Challenges", 2019, Nova Science Publishers.
2. Guojun Wang, Md Tabrez Nafis, Muhammad Arif, Mazin Abed Mohammed "Vehicular Ad Hoc Networks Futuristic Technologies for Interactive Modelling, Dimensioning, and Optimization" 2022, CRC Press
3. Jiajia Liu, Abderrahim Benslimane ·"Intelligent and Connected Vehicle Security", 2022 River Publishers.

UCSXXX: Cybersecurity for Mobility Systems

L	T	P	Cr
3	0	0	3.0

Course Objectives: To understand the basics of cyber security in automotive systems. Acquiring knowledge on securities in vehicle communication systems. To understand the Autosar embedded security concepts. To understand the trends and standards for cyber security

Introduction to Automotive cyber security: Network and Security Concepts- Symmetric and Asymmetric Encryption –Domain Name System (DNS)- firewall- attack value chain- holistic cybersecurity solution, Introduction to features of AIS Standards (189, 190), ISO 27001, 21434 -SAE J2534.

Hacking and automotive attack surface: Hacking- Antiforensics - Tunneling Techniques - Fraud Techniques Threat Infrastructure-Automotive attack surface- Intrusion detection and preventions- Phishing – Password cracking – Keyloggers and Spywares – Industrial CSMS
(cyber security management system), DoS and DDoS attacks, – SQL Injection.

Security and privacy: Cryptography-confidentiality-data integrity- authentication, Cryptography primitives, Elliptic Curve cryptography – Key distribution and Key exchange protocols- Autonomous vehicle cybersecurity, connected vehicle security, Privacy of digital personal data protection act, Europe (EU) GDPR act characteristic features and its importance, Comparison of India and EU act.

Cyber security for In-vehicle communication: In-vehicle networks, threat analysis and vulnerabilities, security mechanisms and architectures - In-Vehicle cybersecurity issues, - cybersecurity protection layers- cybersecurity for ECU. AIS Standard 190 regarding communication aspects, Over the Air (OTA) updates and its features.

Cyber security for Intra-vehicle communication: V2X communication -VANET Technology-Homomorphic Encryption in VANET- Security and privacy in V2X communication- IEEE 1609 & Wireless Access in Vehicular Environments (WAVE)- Attacking Wireless Systems- Potential attacks and Remote attacking in V2X.

AUTOSAR embedded security: Introduction to Autosar architecture - Autosar COM Stack - Autosar Diagnostic Stack, - Autosar Crypto Stack - Layered Automotive Security Approach- Automotive Security Controls- Gateway ECU- Automotive Network Topology- Diagnostics CAN Security.

Cyber security Risk management and Standards: Applied standards and cyber risk management, ISO standards towards risk - Seven Principles of Network Security Analysis Strategy, Network Traffic Monitoring and Analysis standards - cyber security of SAE level 2, 3, and 4 autonomous driving systems-Cyber-attacks in future autonomous vehicle.

Contemporary Issues: Working with OWASP - Honeypots, password guessing and cracking.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

On completion of this course, the students will be able to:

1. Understand potential cyber security for automotive systems.
2. Understand the hacking surface of vehicle
3. Identify the security for intelligent vehicle
4. Need of cyber security for IN and Intra vehicle communication
5. Applying embedded security for vehicle
6. Automotive standards for securities.

Text Books:

1. Shiho Kim , Rakesh Shrestha, "Automotive Cyber Security Introduction, Challenges, and Standardization", 2020, 1,Springer Singapore.
2. Dietmar P.F. Möller, Roland E. Haas, "Guide to Automotive Connectivity and Cybersecurity", 2019, Springer Cham.
3. Ahmad Mk Nasser, "Automotive Cybersecurity Engineering Handbook", 2023, 1, Pack. Publishing limited,

Reference books:

1. Craig Gibbs "Automotive Cybersecurity:Issues and Vulnerabilities" 2016 Nova Science Publishers.
2. Gloria D'Anna "Cybersecurity for Commercial Vehicles" 2018 SAE International.
3. Jiajia Liu, Abderrahim Benslimane ·"Intelligent and Connected Vehicle Security", 2022 River Publishers.

UCS679: Data Analytics in Automobile Engineering

L	T	P	Cr
3	0	0	3.0

Course Objectives: To enable students to understand the fundamentals of data analytics and various tools and software included in Automobile Engineering,To enable the students to understand the data hygiene including data privacy. Covers the data analytics pipeline, including the tools and techniques used to analyze data. To demonstrate how data analysis influence performance improvement in electric vehicles.

Introduction to Truck Components and Data Analytics: Overview of EV technologies, ICE (Hydrogen, diesel, alternate fuels), Fuel cell technologies – Understanding various components- Classification of Engines, Transmission types – Data Analysis fundamentals – common terms - Importance of Data Analytics in automobiles - Data Hygiene – cloud computing - Levels of Data Analytics - Data privacy – GDPR - Data ethics – Legal Frameworks and intellectual property - Ethical practices in handling and analyzing vehicle data - Storage and security

Mathematical Foundation for Data Analytics : Introduction to Statistics and Probability – Conditional Probability and Independence – Bayes' theorem – convergence and sampling - discrete and continuous random variables – key distributions – normal distributions and its spin-offs – covariance and correlation – central limit theorem – Probabilistic Modelling of Problems

Data Collection and Management in Automobiles: Types of Data Collected by Vehicles - Telematics/ Sensor data: GPS, speed, engine status, tire pressure, vehicle speed, torque, emissions, fuel economy, etc - Battery data: charge level, temperature, and health – Production data - Warranty & claims data – Big Data Handling - Introduction and basics of Cloud computing and related terms. Handling aggregated sensor data and high frequency time series data.

Understanding the basic Data Analytics Software's and Tools: Introduction to the software basis – version control (GIT) – software building testing – automation, Introduction of the basic data analysis tools used – Microsoft Excel – Python – Jupyter Notebook Apache Spark – Microsoft Power BI – SQL -Azure Databricks. How to choose a data analytics tool Basic idea about the installation and usage of the tools

Data Analytics Pipeline for Vehicle data Analytics: Cleaning and formatting vehicle data – Dealing with missing and inconsistent data – Feature Engineering – Creating new features to enhance model performance – Exploratory Data Analysis for Data - Visualizing and understanding patterns in data – Creating Dashboards – Design principles for effective dashboards - Extracting insights to inform analytics strategies – Good Practices - Build
an end-to-end scalable Data Analytics Pipeline

Predictive Modelling for Vehicle Performance: Overview of ML algorithms – Supervised and unsupervised learning - Classification vs. Prediction - Association Rules -Different Classification models, Clustering techniques, Time Series Models.

Choosing appropriate model for different type of use cases: models to predict battery range, fuel economy – Assessing and maintaining battery health, sensor levels through analytics - analyzing charging pattern, emission rates for optimization, Root cause Analysis for product issues.

Real Time Monitoring and Control: Importance of real-time monitoring for vehicle safety and efficiency - Technologies for implementing real-time analytics/alert systems in systems - Building adaptive systems using data feedback - Adjusting vehicle parameters based on real-time analytics - Data Quality Monitoring, Anomaly Detection - Examining successful examples of real-time analytics in the automobile industry

Contemporary Issues: Case studies of application of data analysis for use cases (Trucks) in automobiles.

Course Learning Outcomes (CLOs) / Course Objectives (COs):

On completion of this course, the students will be able to:

1. Participants will have acquired the skills and knowledge necessary to leverage data analytics for optimizing vehicle performance.
2. They will be equipped to analyze, interpret, and apply insights from vehicle data, contributing to the advancement of sustainable and efficient transportation.

Text Books:

1. Andrew Bruce, Peter Bruce, Peter Gedeck "Practical Statistics for Data Scientists"
2. Michael Berthold, David J. Hand, “Intelligent Data Analysis”, Springer, 2007
3. S. Angalaeswari, T.Deepa, L.Ashok Kumar , "Artificial Intelligence Applications in Battery Management Systems and Routing Problems in Electric Vehicles"

Reference books:

1. Anand Rajaraman and Jeffrey David Ullman, “Mining of Massive Datasets”, CUP,2012.
2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database System Concepts”, Sixth Edition, McGrawHill.
3. Jeff M. Phillips, "Mathematical Foundations for Data Analysis", First Edition, Springer Cham
4. T. Hastie, R. Tibshirani and J. Friedman, “Elements of Statistical Learning”, Springer, 2009.
5. Da Ruan, Guoquing Chen, Etienne E.Kerre, Geert Wets, “Intelligent Data Mining”, Springer, 2007.

UCS678: Intelligent Transportation Systems

L	T	P	Cr
3	0	0	3.0

Course Objectives: Understand ITS Fundamentals. Explore ITS Applications and Solutions. Analyze ITS Challenges and Future Trends. Understand the future technologies related to ITS

Basics of ITS: Introduction to Intelligent Transportation Systems (ITS) -Definition – Role and Responsibilities - Historical development and Evolution of ITS - Component and Technologies in ITS - ITS stakeholders and their roles and Key ITS terminologies.

Challenges and opportunities in ITS : Traffic congestion and environmental issues - Safety and security concerns - Economic and societal benefits of ITS - Emerging trends and future prospects - Case studies on successful ITS implementations

Systems engineering in ITS and ITS architecture: ITS Standards and Architecture - ITS Communication Systems - ITS System Integration - ITS and Security - ITS Policy Issues - ITS project development life cycle - Architectural frameworks (e.g. - C2C-CC - C-ITS) - Case studies on ITS architecture design.

ITS applications: Integration of ITS with autonomous vehicles - Autonomous vehicle technologies and levels of automation - ITS applications for autonomous vehicle navigation - Advanced Traveller Information System - Fleet Oriented ITS Services - Electronic Toll Collection - Automatic Vehicle Location (AVL) - Automatic Vehicle - Identification (AVI) - Case studies on autonomous vehicle-ITS integration.

Connected and autonomous vehicles (C&AV): Introduction to Cooperative Vehicle Communications - Introduction to DSRC - Introduction to Cellular based Vehicle to Everything (C-V2X) Communications - Communication Standards - Different types of C-V2X – V2V - V2I - V2N - V2P. General Architecture of connected V2X system as per cellular standards - Applications of C-V2X - Introduction and importance of QoS - Automotive surrounding sensing with GHz and THz signals. Introduction to Adaptive Traffic Control Systems and its technologies.

Supporting ITS Technologies: Sensor technologies (LiDAR - RADAR - cameras) - Communication protocols (DSRC - 5G - Wi-Fi) - Data analytics and machine learning in ITS - Integration of supporting technologies with ITS.

ITS standards and specifications: Overview of international ITS standards (e.g. - 3GPP - ETSI - C2C - IEEE - SAE - ISO) - C-V2X architectural frameworks with regards to 3GPP and ETSI - IT'S in regulatory frameworks.

Contemporary Issues: Expert Industry Lecture / Case Study

Course Learning Outcomes (CLOs) / Course Objectives (COs):

On completion of this course, the students will be able to:

1. Demonstrate a strong understanding of core ITS concepts and principles.
2. Apply ITS knowledge to address real-world transportation challenges effectively.
3. Stay informed about emerging trends and innovations in the field of Intelligent Transportation Systems.
4. Demonstrate a strong foundation in the wireless communication standards related to ITS technology.
5. Demonstrate a strong knowledge in the sensor and AI technologies supporting ITS and the application of autonomous vehicle.

Text Books:

1. Shanzhi Chen - Jinling Hu - Li Zhao - Rui Zhao - Jiayi Fang and Yan Shi and Hui Xu ”Cellular Vehicle to everything Communications (C-V2X)” - Springer Nature - Singapore - 2023.
2. Yu - Huafeng - Xin Li - Richard M. Murray - S. Ramesh - and Claire J. Tomlin - eds. Safe - autonomous and intelligent vehicles. Springer - 2018.
3. Dimitrakopoulos - George J. - Lorna Uden - and Iraklis Varlamis. The future of intelligent transport systems. Elsevier - 2020.

Reference books:

1. Garg - Sahil - Gagangeet Singh Aujla - Kuljeet Kaur - and Syed Hassan Ahmed Shah - eds. Intelligent cyber-physical systems for autonomous transportation. Springer - 2022.
2. Bazzan - Ana LC - and Franziska Klügl. Introduction to intelligent systems in traffic and transportation. Springer Nature - 2022.
3. Mouftah - Hussein T. - Melike Erol-Kantarci - and Sameh Sorour - eds. Connected and Autonomous Vehicles in Smart Cities. CRC Press - 2020.
4. Robert Gordon - “Intelligent Transportation Systems” - Springer International Publishing - 2016.

Generic Electives

UHU016: INTRODUCTORY COURSE IN FRENCH

L	T	P	Cr
2	0	0	2.0

Course Objectives:

The objectives of the course are to introduce to the students:

1. The basics of French language to the students. It assumes that the students have minimal or no prior knowledge of the language.
2. To help them acquire skills in writing and speaking in French, comprehending written and spoken French.
3. The students are trained in order to introduce themselves and others, to carry out short conversation, to ask for simple information, to understand and write short and simple messages, to interact in a basic way.
4. The main focus of the students will be on real life language use, integration of French and francophone culture, & basic phrases aimed at the satisfaction of needs of concrete type.
5. During class time the students are expected to engage in group & pair work.

Communicative skills: Greetings and Its Usage, Asking for and giving personal information, How to ask and answer questions, How to talk over the phone, Exchange simple information on preference, feelings etc. Invite, accept, or refuse invitation, Fix an appointment, Describe the weather, Ask for/give explanations, Describe a person, an object, an event, a place.

Grammar : Pronouns: Pronom sujets (Je/ Tu/Il/Elle/Nous/Vous/Ils/Elles), Nouns: Genders, Articles: Definite article and Indefinite articles, Verbs: Regular verbs (-er, -ir ending) Irregular verbs (-re ending), Auxiliary verbs (avoir, être, aller). Adjective: Description, Adjective possessive, Simple Negation, Tense: Present, Future, Questions, Singular & plural.

Vocabulary: Countries and Nationalities, Professions, Numbers (ordinal, cardinal), Colours, Food anddrinks, Days of the week, Months, Family, Places.

Phonetics: The course develops the ability, to pronounce words, say sentences, questions and give orders using the right accent and intonation. To express surprise, doubt, fear, and all positive or negative feelings using the right intonation. To distinguish voiced and unvoiced consonants. To distinguish between vowel sounds.

Course Learning Outcomes (CLOs):

Upon the completion of the course:

1. The students begin to communicate in simple everyday situations acquiring basic grammatical structure and vocabulary.
2. The course develops oral and reading comprehension skills as well as speaking and writing.
3. Students can demonstrate understanding of simple information in a variety of authentic materials such as posters, advertisement, signs etc.
4. Discuss different professions, courses and areas of specialisation.

5. Write simple messages, letters, composition and dialogues. Complete simple forms and documents.
6. Express feelings, preferences, wishes and opinions and display basic awareness of francophone studies.
7. Units on pronunciation and spelling expose students to the different sounds in the French language and how they are transcribed.

Recommended Books :

1. Alter ego-1 : Méthode de français by Annie Berthet, Catherine Hugot, Véronique M. Kizirion,Beatrix Sampsonis, Monique Waendendries, Editions Hachette français langue étrangère.
2. Connexions-1 : Méthode de français by Régine Mérieux, Yves Loiseau, Editions Didier
3. Version Originale-1: Méthode de français by Monique Denyer, Agustin Garmendia.
4. Marie-Laure Lions-Olivieri, Editions Maison des Langues, Paris 2009
5. Latitudes-1 : Méthode de français by Régine Mérieux, Yves Loiseau, Editions Didier
6. Campus-1 : Méthode de français by Jacky Girardet, Jacques Pécheur, Editions CLE International.
7. Echo-1 : Méthode de français by J. Girardet, J. Pécheur, Editions CLE International.

Evaluation Scheme:

Sr. No.	Evaluation Elements	Weightage (%)
1	MST	45
2	EST	55

UCS002: INTRODUCTION TO CYBER SECURITY

L	T	P	Cr
2	0	0	2.0

Course Objectives: In this course, the student will learn about the essential building blocks and basic concepts around cyber security such as Confidentiality, Integrity, Availability, Authentication, Authorization, Vulnerability, Threat and Risk and so on.

Introduction: Introduction to Computer Security, Threats, Harm, Vulnerabilities, Controls, Authentication, Access Control, and Cryptography, Authentication, Access Control, Cryptography

Programs and Programming: Unintentional (Non-malicious) Programming Oversights, Malicious Code—Malware, Countermeasures

Web Security: User Side, Browser Attacks, Web Attacks Targeting Users, Obtaining User or Website Data, Email Attacks

Operating Systems Security: Security in Operating Systems, Security in the Design of Operating Systems, Rootkit

Network Security: Network Concepts, Threats to Network Communications, Wireless Network Security, Denial of Service, Distributed Denial-of-Service Strategic Defenses: Security Countermeasures, Cryptography in Network Security, Firewalls, Intrusion Detection and Prevention Systems, Network Management

Cloud Computing and Security: Cloud Computing Concepts, Moving to the Cloud, Cloud Security Tools and Techniques, Cloud Identity Management, Securing IaaS

Privacy: Privacy Concepts, Privacy Principles and Policies, Authentication and Privacy, Data Mining, Privacy on the Web, Email Security, Privacy Impacts of Emerging Technologies, Where the Field Is Headed

Management and Incidents: Security Planning, Business Continuity Planning, Handling Incidents,Risk Analysis, Dealing with Disaster

Legal Issues and Ethics: Protecting Programs and Data, Information and the Law, Rights of Employees and Employers, Redress for Software Failures, Computer Crime, Ethical Issues in Computer Security, Incident Analysis with Ethics

Emerging Topics: The Internet of Things, Economics, Computerized Elections, Cyber Warfare.

Course Learning Outcomes (CLOs):

After completion of this course, the students will be able to:

1. Understand the broad set of technical, social & political aspects of Cyber Security and security

management methods to maintain security protection

2. Appreciate the vulnerabilities and threats posed by criminals, terrorist and nation states to national infrastructure
3. Understand the nature of secure software development and operating systems
4. Recognize the role security management plays in cyber security defense and legal and social issues at play in developing solutions.

Recommended Books:

1. Pfleeger, C.P., Security in Computing, Prentice Hall, 5th edition (2010)
2. Schneier, B., Applied Cryptography, Second Edition, John Wiley & Sons (1996)
3. Rhodes-Ousley, M., Information Security: The Complete Reference, Second Edition, InformationSecurity Management: Concepts and Practice. New York, McGraw-Hill, (2013).
4. Whitman, M.E. and Herbert J. M., Roadmap to Information Security for IT and Infosec Managers, Course Technology, Boston, MA (2011).

Evaluation Scheme:

Sr. No.	Evaluation Elements	Weightage (%)
1	MST	45
2	EST	55

UEN006: TECHNOLOGIES FOR SUSTAINABLE DEVELOPMENT

L	T	P	Cr
2	0	0	2.0

Course Objectives: To provide acquaintance with modern cleaner production processes and emerging energy technologies; and to facilitate understanding the need and application of green and renewable technologies for sustainable development of the Industry/society

Concepts of Sustainability and Industrial Processes: Industrialization and sustainable development; Cleaner production (CP) in achieving sustainability; Source reduction techniques - Raw material substitution; Process modification and equipment optimization; Product design or modification; Reuse and recycling strategies; Resources and by-product recovery from wastes; Treatment and disposal; CDM and Pollution prevention programs; Good housekeeping; CP audits,

Green Design: Green buildings - benefits and challenges; public policies and market-driven initiatives; Effective green specifications; Energy efficient design; Passive solar design; Green power; Green materials and Leadership in Energy and Environmental Design (LEED)

Renewable and Emerging Energy Technologies: Introduction to renewable energy technologies- Solar; wind; tidal; biomass; hydropower; geothermal energy technologies; Emerging concepts; Biomolecules and energy; Fuel cells; Fourth generation energy systems,

Course Learning Outcomes (CLOs):

Upon completion of the course, the students will be able to:

1. comprehend basic concepts in source reduction, waste treatment and management
2. Identify and plan cleaner production flow charts/processes for specific industrial sectors
3. examine and evaluate present and future advancements in emerging and renewable energy technologies

Recommended Books

1. Kirkwood, R,C, and Longley, A,J, (Eds,), Clean Technology and the Environment, Chapman & Hall,London (1995),
2. World Bank Group; Pollution Prevention and Abatement Handbook – Towards Cleaner Production, World Bank and UNEP; Washington DC (1998),
3. Modak, P,, Visvanathan, C, and Parasnis, M,, Cleaner Production Audit, Course Material on Cleaner Production and Waste Minimization; United Nations Industrial Development

Organization (UNIDP) (1995),
4. Rao, S, and Parulekar, B,B,, Energy Technology: Non-conventional; Renewable and Conventional; Khanna Pub,(2005) 3rd Ed,

Evaluation Scheme:

Sr. No.	Evaluation Elements	Weightage (%)
1	MST	45
2	EST	55

UHU018: INTRODUCTION TO CORPORATE FINANCE

L	T	P	Cr
2	0	0	2.0

Course Objective: This course aims to provide the students with the fundamental concepts, principles and approaches of corporate finance, enable the students to apply relevant principles and approaches in solving problems of corporate finance and help the students improve their overall capacities.

Introduction to corporate finance: Finance and corporate finance. Forms of business organizations, basic types of financial management decisions, the goal of financial management, the agency problem; the role of the financial manager; basic types of financial management decisions.

Financial statements analysis: Balance sheet, income statement, cash flow, fund flow financial statement analysis Computing and interpreting financial ratios; conducting trend analysis and Du Pont analysis.

The time value of money: Time value of money, future value and compounding, present value and discounting, uneven cash flow and annuity, discounted cash flow valuation.

Risk and return: Introduction to systematic and unsystematic risks, computation of risk and return, security market line, capital asset pricing model.

Long-term financial planning & Financial Decisions: Various sources of long term financing, the elements and role of financial planning, financial planning model, percentage of sales approach, external financing needed. Cost of capital, financial leverage, operating leverage. Capital structure, theories of capital structure net income, net operating income & M&M proposition I and II.

Short-term financial planning and management: Working capital, operating cycle, cash cycle, cash budget, short-term financial policy, cash management, inventory management, credit management.

Capital budgeting : Concepts and procedures of capital budgeting, investment criteria (net present value, payback, discounted payback, average accounting return, internal rate of return, profitability index), incremental cash flows, scenario analysis, sensitivity analysis, break-even analysis,

Dividend policy: Dividend, dividend policy, Various models of dividend policy (Residual approach, Walter model, Gordon Model, M&M, Determinants of dividend policy.

Security valuation: Bond features, bond valuation, bond yields, bond risks, stock features, common stock valuation, and dividend discount & dividend growth models. Common stock yields, preferred stock valuation.

Recommended Books:

1. Brealey, R. A., Myers. S.C., Allen, F., Principles of Corporate Finance (9th edition), The McGraw-Hill, London, (2006).
2. Ehrhardt, M.C., Brigham, E.F., Financial Management: Theory and Practice (10th edition) South Western-Cengage, New York (2011)
3. Van Horne, J.C., Wachowicz, J.M., Kuhlemeyer, G.A., 2005, Fundamentals of Financial Management, Pearson, Vancouver (2010)
4. Pandey, I. M., Financial management, Vikas Publishing House Pvt. Ltd., Noida (2011)
5. Elton, E.J. and Gruber, M.J., Modern Portfolio Theory and Investment Analysis, (7th Edition), John Wiley and Sons, New York (2007)

Evaluation Scheme:

Sr. No.	Evaluation Elements	Weightage (%)
1	MST	45
2	EST	55

UHU017: INTRODUCTION TO COGNITIVE SCIENCE

L T P Cr

2 0 0 2.0

Course Objectives: This course provides an introduction to the study of intelligence, mind and brain from an interdisciplinary perspective, It encompasses the contemporary views of how the mind works, the nature of reason, and how thought processes are reflected in the language we use, Central to the course is the modern computational theory of mind and it specifies the underlying mechanisms through which the brain processes language, thinks thoughts, and develops consciousness,

Overview of Cognitive Science: Newell's big question, Constituent disciplines, Interdisciplinary approach, Unity and diversity of cognitive science,

Philosophy: Philosophy of Mind, Cartesian dualism Nativism vs, empiricism, Mind-body problem, Functionalism, Turing Test, Modularity of mind, Consciousness, Phineas Gage, Physicalism.

Psychology: Behaviorism vs, cognitive psychology, The cognitive revolution in psychology, Hardware/software distinction , Perception and psychophysics, Visual cognition, Temporal dynamics of visual perception, Pattern recognition, David Marr's computational theory of vision, Learning and memory, Theories of learning, Multiple memory systems, Working Memory and Executive Control, Memory span, Dissociations of short- and long-term memory, Baddeley's working memory model.

Linguistics: Components of a grammar, Chomsky, Phrases and constituents, Productivity, Generative grammars, Compositional syntax, Productivity by recursion, Surface- and deep structures, Referential theory of meaning, Compositional semantics, Semantics, Language acquisition, Language and thought.

Neuroscience: Brain anatomy, Hierarchical functional organization, Decorticate animals, Neuroimaging, Neurophysiology, Neuron doctrine, Ion channels, Action potentials, Synaptic transmission, Synaptic plasticity, Biological basis of learning, Brain damage, Amnesia, Aphasia, Agnosia, Parallel Distributed Processing(PDP), Computational cognitive neuroscience, The appeal of the PDP approach, Biological Basis of Learning, Cajal's synaptic plasticity hypothesis, Long-term potentiation (LTP) and depotentiation (LTD), NMDA receptors and their role in LTP, Synaptic consolidation, Vertical integration, The Problem of representation, Shannon's information theory.

Artificial Intelligence: Turing machines, Physical symbol systems, Symbols and Search Connectionism, Machine Learning,, Weak versus strong AI, Subfields, applications, and recent trends in AI, Turing Test revisited, SHRDLU, Heuristic search, General Problem Solver (GPS), Means-ends analysis.

Cognitive architectures: Tripartite architecture, Integration, ACT-R Architecture Modularity.

Course Learning Outcomes (CLOs):

Upon completion of the course, the students will be able to:

1. identify cognitive science as an interdisciplinary paradigm of study of cross-cutting areas such as Philosophy, Psychology, Neuroscience, Linguistics, Anthropology, and Artificial Intelligence.
2. explain various processes of the mind such as memory and attention, as well as representational and modelling techniques that are used to build computational models of mental processes;
3. acquire basic knowledge of neural networks, linguistic formalism, computing theory, and the brain.
4. apply basic Artificial Intelligence techniques to solve simple problems.

Recommended Books

1. Bermúdez, J.L., Cognitive Science: An Introduction to the Science of the Mind (2nd Ed,),Cambridge, UK: Cambridge (2014).
2. Friedenberg ,J,D, and Silverman,G, Cognitive Science: An Introduction To The Study Of Mind,Sage Publications:, London (2014)
3. Thagard, P., Mind: An introduction to Cognitive Science, MIT Press, (2005)
4. Thagard, P., (1998) Mind Readings: Introductory Selections on Cognitive Science, MIT Press,Cambridge, Mass,

Evaluation Scheme:

Sr. No.	Evaluation Elements	Weightage (%)
1	MST	45
2	EST	55

UPH064: NANOSCIENCE AND NANOMATERIALS

L	T	P	Cr
2	0	0	2.0

Course Objective:

To introduce the basic concept of Nanoscience and advanced applications of nanotechnology,

Fundamental of Nanoscience: Features of Nanosystem, Free electron theory and its features, Idea of band structures, Density of states in bands, Variation of density of state and band gap with size of crystal,
Quantum Size Effect: Concepts of quantum effects, Schrodinger time independent and time dependent equation, Electron confinement in one-dimensional well and three-dimensional infinite square well, Idea of quantum well structure, Quantum dots and quantum wires,
Nano Materials: Classification of Nano Materials their properties, Basic concept relevant to application, Fullerenes, Nanotubes and nano-wires, Thin films chemical sensors, Gas sensors, Vapour sensors and Bio sensors,
Synthesis and processing: Sol-gel process, Cluster beam evaporation, Ion beam deposition, Chemical bath deposition with capping techniques and ball milling, Cluster assembly and mechanical attrition, Sputtering method, Thermal evaporation, Laser method,
Characterization: Determination of particle size, XRD technique, Photo luminescence, Electron microscopy, Raman spectroscopy, STEM, AFM,
Applications: Photonic crystals, Smart materials, Fuel and solar cells, Opto-electronic devices

Course outcomes:
Upon completion of the course, Students will be able to

1. discriminate between bulk and nano materials,
2. establish the size and shape dependence of Materials' properties,
3. correlate 'quantum confinement' and 'quantum size effect' with physical and chemical properties of nanomaterials,
4. uses top-down and bottom-up methods to synthesize nanoparticles and control their size and shape
5. characterize nanomaterials with various physico-chemical characterization tools and use them in development of modern technologies

Recommended Books:

1. Booker, R., Boysen, E., Nanotechnology, Wiley India Pvt, Ltd, (2008)
2. Rogers, B., Pennathur, S., Adams, J., Nanotechnology, CRS Press (2007)
3. Bandyopadhyay, A,K., Nano Materials, New Age Int,, (2007)
4. Niemeyer, C. N., and Mirkin, C, A., Nanobiotechnology: Concepts, Applications andPerspectives, Wiley VCH, Weinhein, Germany (2007)

Evaluation Scheme:

Sr. No.	Evaluation Elements	Weightage (%)
1	MST	45
2	EST	55

UMA069: GRAPH THEORY AND APPLICATIONS

L	T	P	Cr
2	0	0	2.0

Course Objective:

The objective of the course is to introduce students with the fundamental concepts in graph Theory, with a sense of some its modern applications. They will be able to use these methods in subsequent courses in the computer, electrical and other engineering,

Introduction: Graph, Finite and infinite graph, incidence and degree, Isolated vertex, Pendent vertex and null graph, Isomorphism, Sub graph, Walks, Paths and circuits, Euler circuit and path, Hamilton path and circuit, Euler formula, Homeomorphic graph, Bipartite graph, Edge connectivity, Computer representation of graph, Digraph.

Tree and Fundamental Circuits: Tree, Distance and center in a tree, Binary tree, Spanning tree, Finding all spanning tree of a graph, Minimum spanning tree.

Graph and Tree Algorithms: Shortest path algorithms, Shortest path between all pairs of vertices, Depth first search and breadth first of a graph, Huffman coding, Cuts set and cut vertices, Warshall's algorithm, topological sorting.

Planar and Dual Graph: Planner graph, Kuratowski's theorem, Representation of planar graph, five-color theorem, Geometric dual.

Coloring of Graphs: Chromatic number, Vertex coloring, Edge coloring, Chromatic partitioning, Chromatic polynomial, covering.

Application of Graphs and Trees: Konigsberg bridge problem, Utilities problem, Electrical network problem, Seating problem, Chinese postman problem, Shortest path problem, Job sequence problem, Travelling salesman problem, Ranking the participant in a tournament,Graph in switching and coding theory, Time table and exam scheduling, Applications of tree and graph in computer science.

Course Learning Outcomes (CLOs):

Evaluation Scheme:

Sr. No.	Evaluation Elements	Weightage (%)
1	MST	45
2	EST	55

UMA070: ADVANCED NUMERICAL METHODS

L	T	P	Cr
2	0	0	2.0

Course Objectives:
The main objective of this course is to motivate the students to understand and learn various advanced numerical techniques to solve mathematical problems governing various engineering and physical problems.

Non-Linear Equations: Methods for multiple roots, Muller‘s, Iteration and Newton-Raphson method for non-linear system of equations and Newton-Raphson method for complex roots.
Polynomial Equations: Descartes‘ rule of sign, Birge-vieta, Giraffe‘s methods.
System of Linear Equations: Cholesky and Partition methods, SOR method with optimal relaxation parameters.
Eigen-Values and Eigen-Vectors: Similarity transformations, Gerschgorin‘s bound(s) on eigenvalues, Given‘s and Rutishauser methods.
Interpolation and Approximation: Cubic and B – Spline and bivariate interpolation, Least squares approximations, Gram-Schmidt orthogonalisation process and approximation by orthogonal polynomial, Legendre and Chebyshev polynomials and approximation.
Differentiation and Integration: Differentiation and integration using cubic splines, Romberg integration and multiple integrals.
Ordinary differential Equations: Milne‘s, Adams-Moulton and Adam‘s Bashforth methods with their convergence and stability, Shooting and finite difference methods for second order boundary value problems.

Course Learning Outcomes:
Upon completion of this course, the students will be able to**:**
1) find multiple roots of equation and apply Newton -Raphson's method to obtain complex roots as well solution of system of non - linear equations.
2) learn how to obtain numerical solution of polynomial equations using Birge - Vitae and Giraffe's methods.
3) apply Cholesky, Partition and SOR methods to solve system of linear equations.
4) understand how to approximate the functions using Spline, B- Spline, least square .approximations
5) learn how to solve definite integrals by using cubic spline, Romberg and initial value problems and boundary value problems numerically.

Recommended Books

1) *Gerald, C.F. and Wheatley, P.O., Applied Numerical Analysis, Pearson Education (2008) 7th ed.*
2) *Gupta, S.R., Elements of Numerical Analysis, MacMillan India (2009).*

1) *Atkinson, K.E., An introduction to Numerical Analysis, John Wiley (2004) 2nd ed.*
2) *S.D. Conte, S.D. and Carl D. Boor, Elementary Numerical Analysis: An Algorithmic Approach, Tata McGraw Hill (2005).*
3) *Jain M. K., Iyengar. S.R.K. and Jain, R.K. Numerical Methods for Scientific and Engineering Computation, New Age International (2008) 5th ed.*

Evaluation Scheme:

Sr. No.	Evaluation Elements	Weightage (%)
1	MST	45
2	EST	55

UBT510: BIOLOGY FOR ENGINEERS

L	T	P	Cr
2	0	0	2.0

Course Objective: To learn about living world and basic functioning of biological systems. The course encompasses understanding of origin of life, its evolution and some of its central characteristics. It also aims to familiarize engineering students to some of the intricate biological phenomena and mechanisms.

Characteristics of life: Living versus non-living organisms, origin of life, theory of evolution, diversity of life, classification of life into animals, plants, fungi, protists, archea and bacteria. Phylogenetics and its relationship with evolution.
Introduction to biological systems: Cell as basic unit of life, cellular organelles and their functions, important biomacromolecules (carbohydrates, lipids, proteins and nucleic acids) and their properties.
Cell membrane: Membrane structure, selective permeability, transport across cell membrane, active and passive transport, membrane proteins, type of transport proteins, channels and pumps, examples of membrane transport in cell physiology.
Classical and molecular genetics: Heredity and laws of genetics, genetic material and genetic information, Structure and properties of DNA, central dogma, replication of genetic information, universal codon system, encoding of genetic information via transcription and translation.

Course Learning Outcomes (CLOs):
After completion of this course the students will be able to:

1. Explain the characteristic features of living-systems and differentiate them from non-living systems
2. Broaden the application of engineering knowledge of their branch by applying concepts of living systems.
3. Demonstrate familiarity with special properties of biological macromolecules
4. Upgrade their understanding about biological systems by drawing parallel with thermodynamics system and develop interface between an engineering specialization and living systems.
5. Design engineering products inspired by living creatures.
6. Plan application of computational tools in bioinformatics.

Recommended Books:

1. Nelson, D.L., Cox, M.M., Lehninger: Principles of Biochemistry, WH Freeman(2008) 5th ed.
2. Dhami, P.S., Srivastava, H.N. Chopra, G., A Textbook of Biology, Pradeep Publications (2008).
3. Das, H.K., Textbook of Biotechnology, John Wiley & Sons (2004) 3rd Edition.
4. Gardner, E.J., Simmons, M., Peter, S.D., Principles of Genetics, John Wiley & Sons

(2008)

5. Albert, B., Essential Cell Biology, Taylor & Francis, London (2009)**Evaluation Scheme:**

Sr. No.	Evaluation Elements	Weightage (%)
1	MST	45
2	EST	55

	UEC646: Network and Communication for Connected Vehicles	L	T	P	C
		3	0	0	3
Pre-requisite	**NIL**				

Course Objectives

1. To attain a comprehensive overview of connected vehicles.
2. To understand the intra- and inter-vehicle communication.
3. Acquiring knowledge on VANET.
4. Able to tackle current problems in connected vehicle technology

Course Outcomes

1. Understand the fundamentals of connected vehicles.
2. Comprehend network architectures, including intra-vehicle and inter-vehicle networks.
3. Familiarize with the VANET networking and V2X applications
4. Understand the Security and privacy-related issues of vehicular networks.

Module 1	**Introduction to Connected Vehicles**

Cyber Physical Systems in Transportation, Architecture and Components of Connected Cars, On-board Sensors and Platforms

Module 2	**Vehicle Communication Technologies**

Intra-Vehicle: LIN, CAN, FlexRay. **Inter-Vehicle:** Wi-Fi, GPS, GPRS, 5G. **Ethernet Standards:** Automotive Ethernet, SOME-IP, DO-IP

Module 3	**Network Architectures**

Overview of the Architecture- channel models- properties of V2V and V2I communication- performance of 802.11 in V2X.

Module 4	**VANET**

Vehicular AD hoc network operations- single-hop broadcasting and multi-hop broadcasting- mobile IP solution in VANET.

Module 5	**V2V Application**

Emergency vehicle approach, Blind spot lane change warning, Rear end collision warning, Emergency brake light, and Intersection movement assist.

Module 6	**V2I Application**

Traffic information system- Curve speed warning system- work zone warning system- weather impact warning system and Platoon detection.

Module 7	Cyber Security Risk Management and Standards
Secure Vehicular Communication Architecture- Secure and Privacy-enhancing Vehicular Communication- Data Trustworthiness- Future Challenges. SECOC for CAN protocol- MACSEC and TLS for Ethernet.	

Text Book(s)

1. Christoph Sommer, Falko Dressler, Vehicular Networking, Cambridge University Press, 2015.

2. Markus Meuck, Ingolf Karls, Networking Vehicles to Everything, DE Gruyter Press, 2018

3. Dominique Paret, Hassina Rebaine "Autonomous and Connected Vehicles Network Architectures from Legacy Networks to Automotive Ethernet" 2022 Wiley

Reference Books

1. Peter Han Joo Chong "Vehicular Networks Applications, Performance Analysis and Challenges", 2019, Nova Science Publishers.

2. Guojun Wang, Md Tabrez Nafis, Muhammad Arif, Mazin Abed Mohammed "Vehicular Ad Hoc Networks Futuristic Technologies for Interactive Modelling, Dimensioning, and Optimization" 2022, CRC Press

3. Jiajia Liu, Abderrahim Benslimane ·" Intelligent and Connected Vehicle Security", 2022 River Publishers.

Mode of Evaluation: CAT, written assignment, Quiz, FAT, Project, Seminar, group discussion, field work

Evaluation Scheme

MST-25

Sessional-30

EST-45

Recommended by Board of Studies DD-MM-YYYY

Approved by the Academic Council No. xx Date DD-MM-YYYY

UCS553 : ENTERPRISE WEB APPLICATION

L	T	P	Cr
3	0	2	4.0

Course Objectives: To equip undergraduate students with foundational and intermediate concepts of Java programming, object-oriented design, and hands-on coding. By the end of the course, students will be able to build a mini-project in Java that exhibits OOP concepts, exception handling, file operations, and collection usage.

Unit 1: Introduction to Java Programming: History and evolution of Java; Features of Java (platform independence, security, portability); JVM, JDK, JRE – architecture and workflow; Compilation & execution process; Structure of a Java program; Data types, variables, type casting; Operators and expressions; Input/output (Scanner, command-line arguments)

Unit 2: Control Statements & Arrays: Conditional statements (if, if–else, switch); Loops (for, while, do-while); Break, continue, labelled loops; Arrays (1D, 2D); Array operations and common algorithms; Intro to Strings and StringBuffer

Unit 3: Object-Oriented Programming in Java: Class and Object creation; Constructors & constructor overloading; Static members and static blocks; Encapsulation & data hiding; Inheritance (single, multilevel, hierarchical); Method overloading & overriding; Super and this keyword; Final, abstract classes, and interfaces; Polymorphism (compile-time & runtime)

Unit 4: Exception Handling & Packages: Types of errors: compile-time vs runtime; Exception hierarchy; try–catch–finally; throws & throw; Custom exceptions; Creating and using packages; Access modifiers (public, private, protected, default)

Unit 5: File Handling & Java Collections: File I/O using FileInputStream, FileOutputStream, FileReader, FileWriter; Serialization & deserialization; Collections framework overview; List, Set, Map interfaces; ArrayList, LinkedList, HashMap, TreeSet; Iterators & enhanced for-loop; Wrapper classes & autoboxing/unboxing

Unit 6: Multithreading & Basic GUI (Optional but Recommended): Thread lifecycle; Creating threads: Thread class & Runnable interface; Synchronization basics; Inter-thread communication basics; Introduction to Java GUI (AWT/Swing); Building simple forms: buttons, labels, text fields; Event handling model

Unit 7: Project Development Module: Requirement analysis & problem definition; Designing class diagrams (basic UML); Modular programming practices; Integrating multiple Java

concepts; Code testing, debugging & documentation; Final mini project demonstration

Course Learning Outcomes (CLOs) /Course Objectives (COs):

After completing this course, students will be able to:

1. Understand JVM architecture, data types, control constructs, and OOP concepts.
2. Implement classes, objects, inheritance, interfaces, and polymorphism.
3. Apply exception handling, file I/O, collections, and Java utilities.
4. Develop simple GUI/console-based applications.
5. Build and demonstrate a functional **mini-project** using Core Java.

Textbooks & References

- Herbert Schildt, *Java: The Complete Reference*, McGraw Hill
- E. Balaguruswamy, *Programming with Java*
- Kathy Sierra & Bert Bates, *Head First Java*

SUGGESTED MINI PROJECTS (Core Java Level)

Console-Based Projects

1. **Student Record Management System**
2. **Library Management System**
3. **Employee Payroll Management**
4. **Bank Account Simulation (ATM-like menu)**
5. **Car Rental Booking System**
6. **Course Registration System**
7. **Bill Management System (electricity/water)**
8. **Inventory Management System**

GUI-Based Projects (Swing/AWT)

1. **Simple Calculator**
2. **Notepad Text Editor**
3. **Personal Expense Tracker**
4. **Quiz Application**
5. **Tic-Tac-Toe Game**
6. **Basic Chat Simulation (offline, using threads)**
7. **Hotel Room Booking System**

Projects Involving File Handling

1. **Daily Logbook / Diary application**
2. **Password Manager (file-based encryption)**
3. **Data Serialization-Based Student Report System**