Narula Institute of Technology

R25
CURRICULUM
FOR B.TECH
UNDER
AUTONOMY
(NEP-2020
IMPLEMENTED)

COMPUTER SCIENCE AND TECHNOLOGY (DATA SCIENCE)
(CSE-DS)

1st Year 1st Semester									
Sl.No.	Broad Category	Category	Course Code	Course Title	Hours per week				**Credits**
					L	T	P	Total	
A. THEORY									
1	ENGG	Major	CS101	Introduction to Programming and Problem Solving	3	0	0	3	3
2	SCI	Multidisciplinary	PH101	Engineering Physics	3	0	0	3	3
3	SCI	Multidisciplinary	M101	Engineering Mathematics I	3	0	0	3	3
4	HUM	Value added course	HU101	Environmental Science	2	0	0	2	2
5	HUM	Value added course	HU102	Indian Knowledge System	1	0	0	1	1
B. PRACTICAL									
1	ENGG	Major	CS191	Introduction to Programming and Problem-Solving Lab	0	0	3	3	1.5
2	SCI	Multidisciplinary	PH191	Engineering Physics Lab	0	0	3	3	1.5
3	ENGG	Skill Enhancement Course	ME191	Engineering Graphics & Computer Aided Design Lab	0	0	3	3	1.5
4	HUM	Ability Enhancement Course	HU191	Communication & Presentation Skill	0	0	3	3	1.5
C. MANDATORY ACTIVITIES / COURSES									
1	Mandatory	MC181	Induction Program	0	0	0	0	0	0

	Course								
Total of Theory, Practical								**24**	**18**

HUM: Humanities; ENGG: Engineering; SCI: Science; PRJ: Project

Course Name: Introduction to Programming and Problem-Solving
Course Code: CS101
Contact (Periods/Week): 3 periods
Total Contact Hours:36
Credits:3

Prerequisite: Basic programming overview of class 12.

Course Objective: The objectives of this course are to enable the students to:

- Describe the architecture, memory systems, and evolution of computers.
- Convert between number systems and analyze binary arithmetic including IEEE 754 representation.
- Construct algorithms and flowcharts for basic computational problems.
- Implement control structures, arrays, pointers, and functions in C programs.
- Demonstrate structured data types and file I/O using the C programming language.

Course Outcome(s): After the completion of the course students will be able to

CO1	Describe the architecture, memory hierarchy, and generations of computers, and classify hardware and software components, demonstrating a foundation of engineering knowledge required for understanding computing systems.
CO2	Convert values between number systems and analyze signed and IEEE754 floating-point representations, applying core concepts of mathematics and engineering fundamentals to

	solve complex engineering problems.
CO3	Construct flowcharts and algorithms for problem solving and develop modular programs in C using appropriate control logic, reflecting skills in design and development of solutions and modern tool usage.
CO4	Implement programs in C using control structures, arrays, pointers, and storage classes, and differentiate between memory management techniques, showcasing proficiency in problem analysis and engineering practice.
CO5	Demonstrate structured data types, file handling, and system-level I/O operations, and evaluate their effectiveness in ensuring data persistence and interfacing with hardware, promoting effective engineering tool usage and lifelong learning.

CO-PO-PSO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	2	2						2			
CO2	2	2	3	3	3									
CO3	2	3	2	2	2									
CO4	3	2	2	3	3									
CO5	2	2	2	1	1						2			

Course Content

Module1: Basics of Computing & Number Representation (7L)

- History and generations of computers.
- Classification: Digital, Analog, Hybrid, Micro, Mini, Mainframe.
- Computer architecture: Input/Output units, Memory (Primary &Secondary), CPU.
- Number systems: Binary, Octal, Decimal, Hexadecimal.

- Conversions among number systems.
- Signed number representations:1’s, 2’scomplement.
- Floating point representation: IEEE754 single& double precision.
- ASCII codes.
- Overview of compiler, interpreter, assembler.

Module2: Problem Solving & Introduction to C Programming (7L)

- Algorithm, flowchart, and pseudocode.
- Procedural vs Structured programming.
- C basics: keywords, identifiers, variable naming(Hungarian Notation).
- Datatypes, constants, declaration, storage size, endianness.
- Operators: Arithmetic, Logical, Relational, Bitwise, Conditional.
- Operator precedence and type conversions.
- Input/Output: scanf(), printf().

Module3: Control Structures & Program Design(7L)

- Control structures: if, if-else, switch, nested conditions.
- Loops: while, for, do-while, break, continue.
- Goto and labels (with discussion on structured vs unstructured programming).
- Functions: declaration, definition, prototypes.
- Parameter passing, return types, recursion.
- Storage classes: auto, static, extern, register.
- Preprocessor directives and macros.

Module4: Arrays, Pointers and Strings(8L)

- Arrays:1D&2D, array to function passing.
- Pointers: basics, pointer arithmetic, pointer to arrays.
- Strings: character arrays, string library functions, array of strings.
- Dynamic memory allocation: malloc(), calloc (), realloc (), free().

Module5: Structured Data Types, File Handling & System Interface(7L)

- Structures: definition, initialization, array of structures, pointers to structures.
- Unions and enum, typedef, bitfields.
- File I/O in C: fopen(), fclose(), fprintf(), fscanf(), fgetc(), fputc().
- Command line arguments.

Text Books:

1. **"Schaum's Outline of Programming with C"**, Byron S. Gottfried, McGraw-Hill Education,1st Edition (1996).
2. **"LetUsC"**, Yashavant Kanetkar, BPB Publications, 17th Edition.
   "Computer Fundamentals", P.K.Sinha and Priti Sinha, BPB Publications, 6th Edition

Reference Books:

1. **"The C Programming Language"**, Brian W. Kernighan and Dennis M. Ritchiem, Prentice Hall, 2nd Edition.
2. **"Fundamentals of Computers"**,V. Rajaraman and Neeharika Adabala, PHI Learning, 6th Edition.
3. **"Computer Organization and Architecture: Designing for Performance"**, William Stallings, Pearson Education, 10th Edition.
4. **"Mastering C"**, K.R. Venugopal and S.R.Prasad,Tata McGraw-Hill Education, 2nd Edition.
5. **"Programming in ANSI C"**, E. Balagurusamy, McGraw Hill Education 8th Edition.

Course Name: Engineering Physics
Course Code: PH101
Contact (Periods/Week): 3 periods

Total Contact Hours:36

Credits:3

Prerequisite: Knowledge of Physics up to 12*th* standard.

Course Objective: The objective of this course is to enable the students to:

- Provide foundational understanding of core physical principles such as optics, quantum mechanics, solid-state physics, and statistical mechanics relevant to engineering disciplines.
- Develop the ability to apply theoretical knowledge of physical sciences in interpreting engineering phenomena and solving problems using scientific reasoning and quantitative analysis.
- Expose students to the working principles of modern devices and

technologies like lasers, fiber optics, semiconductors, and nanomaterials used in engineering and industrial applications.

- Encourage scientific curiosity and innovation by connecting physical theories with practical tools and techniques in emerging fields like nanotechnology and quantum systems.
- Understand the role of physics in interdisciplinary domains for the advancement of science, technology, and sustainable development through real-life engineering contexts.

Course Outcome(s): After the completion of the course students will be able to

CO1	Explain the principles of lasers, fibre optics, and holography and apply them in modern optical and communication systems.
CO2	Identify different crystal structures and compute structural parameters such as Miller indices and packing factors; distinguish between metals, semiconductors, and insulators using band theory.
CO3	Utilize the principles of quantum theory, wave-particle duality, and Schr̈odinger equation—to interpret fundamental quantum phenomena.
CO4	Illustrate the basic concepts of statistical mechanics and examine their implications on microscopic particle behaviour.
CO5	Describe the properties of nanomaterials and display/storage devices and analyze their applications in modern technology.

CO-PO-PSO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3								2		2			
CO2	3	3							2		2			

CO3	3	3							2		2			
CO4	3	3							2		2			
CO5	3	3							2		2			

Course Content

Module 1: Modern Optics (11L)

Laser:(6L) Concepts of various emission and absorption processes, Einstein A and B coefficients and equations, working principle of laser, metastable state, population inversion, condition necessary for active laser action, optical resonator, illustrations of Ruby laser, He-Ne laser, Semiconductor laser, applications of laser, related numerical problems.

Fibre Optics:(3L) Principle and propagation of light in optical fibers (Step index, Graded index, single and multiple modes) - Numerical aperture and Acceptance angle, Basic concept of losses in optical fiber, related numerical problems.

Holography:(2L) Theory of holography (qualitative analysis), viewing of holography, applications

Module 2: Solid State Physics(5L)

Crystal Structure:(3L) Structure of solids, amorphous and crystalline solids (definition and ex- amples), lattice, basis, unitcell, Fundamental types of lattices–Bravais lattice, simple cubic, fc cand bcc lattices, Miller indices and miller planes, co-ordination number and atomic packing factor, Bragg's equation, applications, numerical problems.

Semiconductor:(2L) Physics of semi conductors, electrons and holes, metal, insulator and semiconductor, intrinsic and extrinsic semiconductor, p-n junction.

Module 3: Quantum and Statistical Mechanics (14L)

Quantum Theory:5L Inadequacy of classical physics-concept of quantization of energy, particle concept of electromagnetic wave (example: Black body radiation, Photoelectric and Compton Effect:no derivation required), wave particle duality; phase velocity and group velocity; de Broglie hypothesis; Davisson and Germer experiment, related numerical problems.

Quantum Mechanics 1:4L Concept of wave function, physical

significance of wave function, prob- ability interpretation; normalization of wave functions-Qualitative discussion; uncertainty principle, relevant numerical problems, Introduction of Schrödinger wave equation (only statement).

Statistical Mechanics:5L Concept of energy levels and energy states, phase space, microstates, macro states and thermodynamic probability, MB, BE, FD, statistics (Qualitative discussions)-physical significance, conception of bosons, fermions, classical limits of quantum statistics, Fermi distribution at zero & non-zero temperature, Concept of Fermi level-Qualitative discussion.

Module 4: Physics of Nanomaterials: (4L)

Reduction of dimensionality, properties of nanomaterials, Quantum wells (two dimensional), Quantum wires (one dimensional), Quantum dots (zero dimensional); Quantum size effect and Quantum confinement. Car- bon allotropes. Application of nanomaterials (CNT, graphene, electronic, environment, medical)

Module 5: Storage and display devices (2L)

Different storage and display devices-Magnetic storage materials, Operation and application of CRT, CRO, LED and OLED.

Text Books:

1. **"Concepts of Modern Engineering Physics"**, A.S. Vasudeva, S.Chand Publishers
2. **"Engineering Physics"**, Rakesh Dogra.
3. **"Introduction to Nanoscience and Nanotechnology, An Indian Adaptation"**, Charles P. Poole, Jr., Frank J. Owens.
4. **"Quantum Mechanics"**, S.N. Ghosal.
5. **"Nanotechnology"**, K.K. Chattopadhyay.

Reference Books:

1. **"Optics"**, Ajay Ghatak (TMH).
2. **"Solid state Physics"**, S.O.Pillai.
3. **"Quantum mechanics"**, A.K. Ghatak and SLokenathan.
4. **"Fundamental of Statistical Mechanics"**, B.B. Laud.
5. **"Perspective & Concept of Modern Physics"**, Arthur Beiser.

Course Name: Engineering Mathematics-I
Course Code: M101
Contact (Periods/Week): 3 periods

Total Contact Hours:36

Credits:3

Prerequisite: The students to whom this course will be offered must have the understanding of (10+2) standard algebraic operations, coordinate geometry, and elementary calculus concepts including limits, continuity, differentiation, and integration.

Course Objective: The objective of this course is to enable the students to:

- Develop a strong foundation in both fundamental and advanced concepts of linear algebra and calculus essential for engineering applications.
- Build competency in applying integration techniques in multiple dimensions, including line, surface, and volume integrals, to solve problems relevant to engineering and applied sciences.
- Gain proficiency in analyzing multivariable functions using differentiation techniques such as partial and total derivatives, Jacobians, and methods for finding extrema.

Course Outcome(s): After the completion of the course students will be able to

CO1	Apply linear algebra methods to perform matrix operations, classify matrix structures, solve systems of linear equations, and compute eigenvalues and eigenvectors in engineering contexts.
CO2	Apply differential and integral calculus to evaluate and approximate the behavior of single-variable and multivariable real-valued functions relevant to engineering scenarios.
CO3	Analyze the properties of eigenvalues and eigenvectors to

	assess matrix diagonalizability and interpret linear transformations using the Cayley-Hamilton theorem in engineering systems.
CO4	Analyze single-variable and multivariable real-valued functions using differential and integral calculus to model and interpret complex behavior in engineering applications.

Course Content

Module1: Linear Algebra(11L)

Echelon form and normal (canonical) form of a matrix; Inverse and rank of a matrix; Consistency and inconsistency of system of linear equations, Solution of system of linear equations; Eigenvalues and eigenvectors; Diagonalization of matrix, Cayley-Hamilton theorem.

Module2: Single Variable Calculus(5L)

Rolle's Theorem, Meanvalue theorems, Taylor's and Maclaurin theorems with remainders; Taylor's series.

Module3: Multivariable Calculus(Differentiation)(13L)

Function of several variables; Concept of limit, continuity and differentiability; Partial derivatives, Total derivative and its application; chain rules, Derivatives of implicit functions Euler's theorem on homogeneous function; Jacobian; Maxima and minima of functions of two variables.

Module4: Multivariable Calculus(Integration)(7L)

Double Integral, Triple Integral; Change of order in multiple integrals; Line Integral, Surface Integral, Volume Integral. Change of variables in multiple integrals.

Textbooks:

1. **"Higher Engineering Mathematics"**,Grewal,B.S.,Khanna Publishers,36thEdition,2010.
2. **"Advanced Engineering Mathematics"**, Kreyszig,E., JohnWiley&Sons.

Reference Books:

1. **"A Textbook of Engineering Mathematics-**

I",Guruprasad,S., New Age International Publishers.
2. **"Higher Engineering Mathematics"**, Ramana, B.V., TataMcGrawHill, NewDelhi,11*th*Reprint, 2010.
3. **"Engineering Mathematics for First Year"**,Veerarajan,T.,TataMcGraw-Hill,NewDelhi,2008.
4. **"A text book of Engineering Mathematics"**, Bali, N.P.and Goyal, M., Laxmi Publications, Reprint, 2008.
5. **"Calculus and Analytic geometry"**, 9*th*Edition, Thomas, G.B. and Finney, R.L., Pearson, Reprint, 2002.
6. **"Calculus,Volumes1and2"**,(2ndEdition),Apostol,M.,WileyEastern,1980.
7. **"Linear Algebra-A Geometric approach"**,Kumaresan,S., Prentice Hall of India,2000.
8. **"LinearAlgebra: A Modern Introduction"**, 2ndEdition, Poole,D.,Brooks/Cole,2005.
9. **"Schaum's Outline of Matrix Operations"**,Bronson,R.,1988.
10. **"Differential and Integral Calculus, Vol.I & Vol.II,"**, Piskunov,N.,MirPublishers,1969.

Course Name: Environmental Science
Course Code: HU101
Contact (Periods/Week):2 periods

Total Contact Hours:24

Credits:2
Prerequisite: Knowledge of 10+2 standard.

Course Objective: The objective of this course is to enable the students to:

- Realize the importance of environment and its resources.
- Apply the fundamental knowledge of science and engineering to assess environmental and health risk.
- Know about environmental laws and regulations to develop guidelines and procedures for health and safety issues.
- Solves scientific problem-solving related to air, water, land, and noise pollution.

Course Outcome(s): After the completion of the course students will be able to

CO1	Able to understand the natural environment and its relationships with human activities.
CO2	The ability to apply the fundamental knowledge of science and engineering to assess environmental and health risk.
CO3	Ability to understand environmental laws and regulations to develop guidelines and procedures for health and safety issues.
CO4	Acquire skills for scientific problem-solving related to air, water, noise & land pollution.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1	2	2	3			2	3	1			1			
CO2	3	3	3	1	1	2	3	1			1			
CO3	3	3	3	2	1	2	3	1			1			
CO4	1	3	3			2	1	1			1			

Course Content

Module1: Resources and Ecosystem(6L)

1. **Resources(4L)**
   Types of resources, Human resource, Population Growth models: Exponential Growth, Logistic growth curve with explanation. Maximum Sustainable Yield [Derivation] Alternative sources of Energy [Solar energy, tidal energy, geothermal energy, biomass energy]
2. Ecosystem(2L)
   Components of ecosystem, types of ecosystems, Forest ecosystem, Grassland ecosystem, Desert ecosystem, Pond eco system, Food chain, Food web.

Module2: Environmental Degradation(10L)

1. Air Pollution and its impact on Environment(3L)
   Air Pollutants, primary & secondary pollutants, Criteria pollutants, Smog, Photochemical smog and London smog, Greenhouse effect, Global Warming, Acid rain, Ozone Layer Depletion.
2. Water Pollution and its impact on Environment(4L)
   Water Pollutants, Oxygen demanding wastes, heavy metals, BOD [Rate equation], COD, Eutrophication, Hardness, Alkalinity, TDS and Chloride, Heavy metal (As, Hg, Pb) poisoning and toxicity. Numerical on BOD, Hardness.
3. Land Pollution and its impact on Environment(1L)
   Solid wastes, types of Solid Waste, Municipal Solid Wastes, hazardous wastes, bio-medical wastes, E-wastes,
4. Noise Pollution and its impact on Environment(2L)
   Types of noise, Noise frequency, Noise pressure, Measurement of noise level and decibel (dB) Noise intensity, Noise threshold limit, Effect of noise pollution on human health. Numerical on Measurement of noise level and decibel(dB)and Noise Threshold limit.

Module3: Environmental Management(6L)

1. Environmental Impact Assessment(1L)
   Environmental Auditing, Environmental laws and Protection Acts of India, carbon footprint, green building practices.(GRIHA norms).
2. Pollution Control and Treatment (2L)
   Air Pollution controlling devices, Catalytic Converter, Electrostatic Precipitator. Wastewater Treatment (Surface water treatment &Activated sludge process), Removal of hardness of water (Temporary &Permanent -Permutate process).
3. Waste Management(3L)
   Solid waste management, Open dumping, Landfilling, incineration, composting & Vermicomposting, E-waste management, and Biomedical Waste management.

Module4: Disaster Management(2L)

1. **Study of some important disasters(1L)**

Natural and Man-made disasters, earthquakes, floods drought, landslide, cyclones, volcanic eruptions, tsunami, oil spills, forest fires.

2. Disaster Management Techniques(1L)

Basic principles of disaster management, Disaster Management Cycle, Disaster Management Policy, Awareness generation program.

Text Books:

1. **" Basic Environmental Engineering and Elementary Biology"**, Gourkrishna Dasmohapatra, Vikas Publishing.
2. **"Basic Environmental Engineering and Elementary Biology"** ,Dr.Monindra Nath Patra & Rahul Kumar Singha, Aryan Publishing House.
3. **Textbook of Environmental Studies for Undergraduate"**, ErachBarucha for UGC, Universities Press .

Reference Books:

1. **"A Text Book of Environmental Studies"**, Dr. D.K. Asthana& Dr.Meera Asthana, S. Chand Publications.

2. **"Environmental Science(AsperNEP2020)"**,Subrata Roy, Khanna Publisher.

Course Name: Indian Knowledge System

Course Code: HU102

Contact (Periods/Week): 1 period

Total Contact Hours:12

Credits:1

Prerequisite: A basic knowledge (10+2level) of Indian history, civilization and culture.

Course Objective: The objective of this course is to enable the students to:

- Understand the extent and aspects of ancient Indian cultural, philosophical, and scientific heritage.
- Explore the philosophical roots of Indian knowledge, the scientific temper, and quest for advanced understanding of the universe and deeper knowledge of the self.
- Identify and describe the Indian scientific and technological tools, techniques and discoveries and assess their significance and continuing relevance.
- Develop liberality and open-mindedness of outlook to foster lifelong learning.
- Acquire the skills to apply traditional knowledge in their everyday lives.

Course Outcome(s): After the completion of the course students will be able to:

CO1:	Define, identify, describe and classify the philosophical, literary and socio-religious heritage of ancient India and the core concepts of the Vedic corpus and way of life.
CO2:	Discover, enumerate, compare, contrast and categorize the importance of pioneering developments in science and mathematics and evaluate their continuing relevance.
CO3:	Analyze, appraise, correlate and describe the ancient Indian heritage in science and technology and examine technological correlations with present-day technological applications.
CO4:	Discover, assess and describe traditional knowledge in healthcare, architecture, agriculture and other sectors, and to explore the history of traditional Indian art forms.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1						1	2		3		3			
CO2	2		1			2			3		3			
CO3	3	1	1	1	2	3			3		3			
CO4	2		1		2	3			3		3			

Course Content

Module 1: An Overview of Indian Knowledge System(3L)

- Importance of Ancient Knowledge
- Definition of IKS
- Classification framework of IKS
- Unique aspects of IKS
- The Vedic corpus: Vedas and Vedangas
- Distinctive features of Vedic life
- Indian philosophical systems: Different schools of philosophy(Orthodox and Unorthodox)

Module2: Salient Features of the Indian Numerical System(3L)

- Developments in Indian Mathematics in ancient India
- Importance of decimal representation
- The discovery of zero and its importance
- Unique approaches to represent numbers
- Contribution of ancient Indian mathematicians
- Highlights of Indian Astronomy: Historical Development and Key Contributions

Module3: Indian Science and Technology Heritage(3L)

- Metals and metalworking
- Mining and ore extraction
- Structural engineering and architecture in ancient India: planning, materials, construction, and approaches
- Dyes and painting
- Shipbuilding

Module4: Traditional Knowledge in Different Sectors (3L)

- Traditional knowledge and engineering
- Traditional agricultural practices: resources, methods, technical aids
- Traditional medicine and surgery
- History of traditional art forms and culture

Text Books :

1. **"Traditional Knowledge System in India"**, A.L.Basham, New Delhi: Picador,2019
2. **"Aspects of Science and Technology in Ancient India."**, Arun Kumar Jha and Seema Sahay, Oxford and New Delhi:TaylorandFrancis,2023.
3. **"Indian Knowledge Systems. Vols 1and2."**, Kapil Kapoor and Awadhesh Kumar Singh, New Delhi :D. K. Print world, 2005.
4. **"History of Astronomy in India"**, S.N.Sen and K.S.Shukla, NewDelhi:Indian National Science Academy, 2nd edition, 2000.
5. **"Indian Knowledge System."**, Arpit Srivastava, AKSUniversity,2024.

Course Name: Introduction to Programming and Problem-Solving Lab

Course Code: CS191

Contact (Periods/Week): :3 periods

Total Contact Hours:36

Credits:1.5

Prerequisite: Knowledge of Physics up to 12^{th} standard.

Course Objective: By the end of this course, students will be able to:

- Understand the fundamentals of programming logic through algorithmic thinking.
- Implement and debug C programs using various control structures.
- Apply memory management concepts using pointers and arrays.
- Develop structured programs involving functions and recursion.
- Demonstrate file operations and manipulate data using structures and pointers.

Course Outcome(s): After successful completion of the course, students will be able to:

CO1	Identify fundamental programming constructs such as datatypes, operators, control structures, and apply them to solve basic computational problems.
CO2	Design modular programs using functions, arrays, and structures, and develop reusable solutions to solve real-world problems.
CO3	Demonstrate the use of pointers and dynamic memory management to analyze memory-efficient solutions for complex problems.
CO4	Construct file-based applications that enable persistent data storage and illustrate communication of results through formatted outputs.

CO5	Integrate multiple programming concepts to create a functional mini-project, demonstrating teamwork, project management skills, and adaptability to emerging challenges.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1	3	3				2						2		
CO2	2	2	3		3							2		
CO3	2	3			3							2		
CO4	2	2			3				2			2		
CO5	2	3	2	2	3			3	2	2	3			

Course Content

LabNo.	Title/Experiment	Learning Focus
1	Introduction to C, Basic Input/Output, Data Types, and Operators	I/O operations, operator precedence, expressions
2	Problems on Conditionals: if,if-else, nested if, switch-case	Decision-making constructs
3	Looping Constructs: for, while, do-while	Iterative problem solving
4	Nested Loops: Pattern Printing, Series Problems	Logical structuring using loops
5	Functions: call by value, return types, recursion	Modular programming and recursion
6	Arrays:1D and 2D array manipulation, search/sort problems	Data storage and iteration
7	Strings: string manipulation functions, array of strings	Character arrays and string operations
8	Pointers: pointer arithmetic, pointers with arrays, and functions	Memory-level data access
9	Dynamic Memory Allocation using malloc(),	Run-time memory management

	calloc(), free()	
10	Structures and Unions: defining, accessing, an array of structures, a pointer to a structure	Composite data types and access
11	File/O:fopen(),fprintf(),fscanf(), fgetc(), fputc()	Persistent data storage
12	Mini Project: Combining structures, functions, and file I/O for a real-world scenario	Integration and application of concepts

Text Books:

1. Practical Physics by Chatterjee &Rakshit(Book &Allied Publisher).
2. Practical Physics by K.G. Mazumder(New Central Publishing).
3. Practical Physics by R.K. Kar(Book& Allied Publisher).

Course Name: Engineering Physics Lab
Course Code: PH191
Contact (Periods/Week): :3 periods

Total Contact Hours:36

Credits:1.5

Prerequisite: Knowledge of Physics upto12thstandard.

Course Objective:

- Become familiar with scientific instruments and measurement techniques used to determine various physical parameters of materials and systems.
- Reinforce theoretical concepts learned in classroom physics by performing related practical experiments and observing real-time outcomes.
- Develop a systematic and analytical approach to collecting, organizing, and interpreting experimental data for error analysis and validation of physical laws.
- Engage in the experimental validation of physical laws through laboratory activities involving classical mechanics, optics,

electronics, and quantum phenomena.

- Encourage innovation and problem-solving abilities through hands-on investigation of advanced and application-oriented physics experiments, including specially designed extension activities.

Course Outcome(s): After the completion of the course, students will be able to

CO1	Determine mechanical properties such as Young's modulus and rigidity modulus through hands-on experiments and analyze material behavior under applied forces
CO2	Perform optical experiments including Newton's Rings, laser diffraction, and optical fiber characterization, and interpret the results based on wave optics principles.
CO3	Investigate quantum effects such as the photoelectric effect and atomic transitions, and relate experimental outcomes to basic quantum principles.
CO4	Study the performance of semiconductor and electronic devices like solar cells, LEDs, and LCR circuits, and investigate their operational characteristics.
CO5	Conduct experiments, e/m determination, prism dispersion, or optical rotation to demonstrate the application of advanced physical principles in practical scenarios.

CO-PO-PS Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1	3	3						3	2		2			
CO2	3	3			3			3	2		2			
CO3	3				3			3	2		2			
CO4	3	3			3			3	2		2			
CO5	3	3			3			3	2		2			

Course Content

Module 1: General idea about Measurement sand Errors(One Mandatory)

Error estimation using Slide calipers/Screw-gauge/and traveling microscope for one experiment.

Module 2: Experiments on Classical Physics(Any 4 to be performed from the following experiments)

1 Study of Torsional oscillation of Torsional pendulum & determination of time using various load of the oscillator.
2 Determination of Young's moduli of different materials.
3 Determination of Rigidity moduli of different materials.
4 Determination of wavelength of light by Newton's ring method.
5 Determination of wavelength of light by Laser diffraction method.
6 Optical Fiber-numerical aperture, power loss.

Module 3: Experiments on Quantum Physics(Any 2 to be performed from the following experiments)

7 Determination of Planck s constant using photoelectric cell.
8 Verification of Bohr's atomic orbital theory through Frank-Hertz experiment.
9 Determination of Stefan's Constant.
Study of characteristic dorsosacral(illumination, areal, spectral).
Study of characteristics of solar cell (I-V characteristics, Power-load characteristics, Power-wavelength characteristics)

Module 4: Perform at least one of the following experiments

11 Determination of Q factor using L C R Circuit.
12 Study of I-V characteristics of a LED/LDR.
13 Determination of band gap of semiconductor.
**In addition, it is recommended that each student
Should carry out at least one experiment beyond the syllabus/one experiment as Innovative experiment.

Module 5: Probable experiments beyond the syllabus

1 Determination of the specific charge of the electron(e/m) from the path of an electron beam by Thomson method.
2 Determination of Hall co-efficient of a semiconductor and measurement of magnetoresistance of a given semiconductor, Study of dispersive power of material of prism.
3 Determination of thermal conductivity of a bad/good conductor using Lees-Charlton / Searle apparatus.
4 Determination of the angle of optical rotation of a polar solution using polarimeter.
5 Any other experiment related to the theory.

Text Books:

1. Practical Physics by Chatterjee &Rakshit (Book&AlliedPublisher).
2. Practical Physics by K.G.Mazumder (New Central Publishing).
3. Practical Physics by R.K.Kar (Book&AlliedPublisher).

Course Name: Engineering Graphics & Computer Aided Design Lab
Course Code: ME191
Contact (Periods/Week): 3 periods

Total Contact Hours:36

Credits:1.5

Prerequisite: Basic knowledge of geometry.

Course Objective:

- To learn all the skills associated with the tools and inventory associated with the IDEA Lab.
- Learn useful mechanic and electronic fabrication processes.

- Learn necessary skills to build a useful and standalone system/project with enclosures.
- Learn necessary skills to create print and electronic

documentation for the system/project.

Course Outcome(s): After the completion of the course students will be able to

CO1	Use common drafting tools with the knowledge of drafting standards.
CO2	Understand the concepts of engineering scales, projections, sections.
CO3	Apply computer aided drafting techniques to representline, surface or solid models in different Engineering viewpoints.
CO4	Produce part models; carry out assembly operation and represent design project work.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1			3		2	1		2		1				
CO2			3		2	1		2		1				
CO3			3		3	1		2		2				
CO4			3		3	1		2		2				

Course Content

Basic Engineering Graphics(3P)

- Principles of Engineering Graphics
- Ortho graphic Projection
- Descriptive Geometry

- Drawing Principles
- Isometric Projection
- Surface Development
- Perspective
- Reading Drawing
- Sectional Views
- Dimensioning Tolerances
- True Length, Angle, Intersection, Shortest Distance

Module 1: Introduction to Engineering Drawing(6P)

- Principles of Engineering Graphics and their significance
- Usage of Drawing Instruments, Lettering
- Conic Sections including Rectangular Hyperbola(General method only)
- Cycloid, Epicycloid and Involute
- Scales–Plain, Diagonal and Vernier Scales

Module 2: Orthographic & Isometric Projections(6P)

- Principles of Orthographic Projections–Conventions
- Projections of Points and Lines inclined to both Planes
- Projections of Plane son Inclined Planes–Auxiliary Planes
- Projection of Solids inclined to both the Planes–Auxiliary Views
- Isometric Scale and Views of Lines, Planes, Simple and Compound Solids
- Conversion of Isometric Views to Orthographic Views and Vice-versa

Module 3: Sections and Sectional Views of Right Angular Solids (6P)

- Drawing Sectional Views of Solids–Prism, Cylinder, Pyramid, Cone
- Project the True Shape of the Sectioned Surface
- Auxiliary Views
- Development of Surfaces of Right Regular Solids–Prism, Pyramid, Cylinder, Cone
- Sectional Orthographic Views of Objects from Industry and Dwellings (Foundation to Slabonly)

Computer Graphics(3P)

- Engineering Graphics Software
- Spatial Transformations
- Orthographic Projections
- Model Viewing
- Coordinate Systems

- Multi-view Projection
- Exploded Assembly
- Animation, Surface & Solid Modeling
- Spatial Manipulation

Module 4: Overview of Computer Graphics(3P)

- Demonstration of CAD Software:
  - Menu System, Toolbars(Standard, Properties, Draw, Modify, Dimension)
  - Drawing Area(Background, Crosshairs, Coordinate System)
  - Dialog Boxes and Windows
  - Shortcut Menus(Button Bars),Zooming Methods
  - Select and Erase Objects

Module 5:CAD Drawing, Customization, Annotations, Layering (6P)

- Drawing Page Setup including Scale Settings, ISO and ANSI Standards
- Drawing Methods – Straight Lines, Circles
- Applying Dimensions and Annotations
- Layers–Creation and Management
- Line Editing(Extend/Lengthen)
- Drawing Sectional Views of Solids
- Annotations, CAD Modeling of Parts and Assemblies with Animation
- Parametric and Non-parametric Solid, Surface and Wireframe Modeling
- Part Editing and Printing Documents

Module 6:Demonstration of a Simple Team Design Project(3P)

- Illustrating Geometry and Topology of Engineered Components
- Creation of Engineering Models
- Presentation in Standard 2D Blueprint Form and 3D Wireframe/Shaded Solids
- Use of Solid-Modeling Software for Component and Assembly-level Models

Textbooks

1. Bhatt N.D., PanchalV.M. & IngleP.R., (2014), *Engineering Drawing*, Charotar Publishing House.

2. K.Venugopal, *Engineering Drawing+AutoCAD*, NewAge International Publishers.

Reference Books

1. Pradeep Jain, Ankita Maheswari, A.P. Gautam, *Engineering Graphics& Design*, Khanna Publishing House.
2. Agrawal B.&Agrawal C.M.(2012),*Engineering Graphics*, TMH Publication.
3. Shah,M.B.&RanaB.C.(2008),*Engineering Drawing and Computer Graphics*, Pearson Education.
4. Narayana,K.L.&P.Kannaiah(2008),*Textbook on Engineering Drawing*, Scitech Publishers.

Course Name: Communication and Presentation Skill
Course Code: HU191
Contact (Periods/Week): 3 periods

Total Contact Hours:36

Credits:1.5

Prerequisite: Basic knowledge of LSRW skills

Course Objective: The objectives of the course are to make the students able to-

- Acquire interpersonal communication skills of listening comprehension and speaking in academic and professional situations.
- Understand English pronunciation basics and remedy errors.
- Operate with ease in reading and writing interface in global professional contexts.
- Deliver professional presentations before a global audience.
- Develop confidence as a competent communicator.

Course Outcome(s): After the completion of the course students will be able to

CO1	Recognize, identify and express advanced skills of Technical Communication in English and Soft Skills through Language Laboratory.
CO2	Understand, categorize, differentiate and inner listening, speaking, reading and writing skills in societal and professional life.
CO3	Analyze, compare and adapt the skills necessary to be a competent interpersonal communicator in academic and global business environments.
CO4	Deconstruct, appraise and critique professional writing documents, models and templates.
CO5	Adapt, negotiate, facilitate and collaborate with communicative competence in presentations and work-specific conclaves and interactions in the professional context.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1								1	2	3				
CO2						2				3				
CO3						2			3	3				
CO4								3		3				
CO5						3			3	3				

Course Content

Module 1: Introduction–Theories of Communication and Soft Skills

- Communication and the Cyclic Process of Communication (Theory, benefits and application)
- Introduction to Workplace Communication(Principles and Practice)
- Non-Verbal Communication and its Application

- Soft Skills Introduction:
  - What is Soft Skills? Significance of Soft Skills
  - Soft Skills vs .Hard Skills
  - Components of Soft Skills
  - Identifying and Exhibiting Soft Skills(Through classroom activity)

Module 2: Active Listening

- What is Active Listening?
- Listening Sub-skills–Predicting,Clarifying, Inferencing, Evaluating, Note-taking
- Differences between Listening and Hearing, Critical Listening, Barriers to Active Listening,Improving Listening
- Listening in Business Telephony and Practice
- **Practical**: Roleplays, Case studies

Module 3:Speaking Skills

- Effective Public Speaking:
  - Selecting the topic, understanding the audience
  - Organizing the main ideas, language and style, delivering the speech
  - Voice clarity
  - **Practical:** Extempore
  - **Self-learning Topics:** Preparation, Attire, Posture, and Delivery techniques
- Pronunciation Guide–Basics of Sound Scripting, Stress and Intonation
- Fluency-focused Activities–JAM, Conversational Role Plays, Speaking using Picture/Audio-Visual Inputs
- Group Discussion: Principles, Do's and Don'ts, and Practice

Module 4:Writing and Reading Comprehension

- Reading and Writing a Book Review(classroom activity)
- Writing a Film Review after watching a short film(classroom activity)
- Reading Strategies: Active Reading, Note-taking, Summarizing, Using Visual Aids like Diagrams and Graphs
- Solving Company-Specific Verbal Aptitude Papers(Synonyms, Antonyms, Error Correction, RC Passages)

Module 5:Presentation Skills

- Kinds of Presentation, Presentation Techniques
- Planning and Structuring the Presentation: Preparation, Research, Evidence
- Delivering the Presentation, Handling Questions, Time Management, Visual Aids
- Self-Introduction, Creation of Video Resume
- Need for Expertise in Oral Presentation
- **Assignment:** Oral Presentation
- Rules of Making a Micro Presentation(PowerPoint)
- **Assignment:** Micro Presentation

Textbooks

1. PushpLata and Sanjay Kumar. *A Handbook of Group Discussions and Job Interviews*. New Delhi: PHI, 2009.
2. Jo Billingham. *Giving Presentations*. NewDelhi: Oxford University Press,2003.
3. B. JeanNaterop and RodRevell. *Telephoning in English.* 3rded.Cambridge: Cambridge University Press, 2004.
4. Jeyaraj John Sekar. *English Pronunciation Skills: Theory and Practice*. NewDelhi: Authorspress,2025.
5. Career Launcher. *IELTS Reading : A Step-by-Step Guide*. G.K.Publications,2028.

Reference Books

1. Ann Baker. *Shipor Sheep: An Intermediate Pronunciation Course*. Cambridge University Press, 2006.
2. Barry Cusack and SamMcCarter. *Improve Your IELTS: Listening and Speaking Skills*. Macmillan, 2007.
3. EricH.Glendinning and BeverlyHolmstr̈om. *Study Reading*. Cambridge UniversityPress,2004.
4. Malcolm Goodale. *Professional Presentations*. Cambridge University Press,2005.
5. MarkHancock. *English Pronunciation in Use*. Cambridge University Press,2003.
6. Tony Lynch. *Study Listening*. Cambridge University Press,2004.
7. J.D.O'Connor. *Better English Pronunciation*. Cambridge University Press,2005.
8. Peter Roach. *English Phonetics and Phonology: A Practical Course*. Cambridge University Press, 2000.

1st Year 2nd Semester

Sl. No.	Broad Category	Category	Course Code	Course Title	Hours per week				Credits
					L	T	P	Total	
A.THEORY									
1	ENGG	Major	CS201	Data Structure & Algorithms	3	0	0	3	3
2	ENGG	Major	CS203	Digital Electronics and Computer Organization	3	0	0	3	3
3	SCI	Multidisciplinary	CH201	Engineering Chemistry	2	0	0	2	2
4	SCI	Multidisciplinary	M201	Engineering Mathematics II	3	0	0	3	3
5	ENGG	Minor	CS202	Introduction to Artificial Intelligence	2	0	0	2	2
6	HUM	Value added course	HU205	Constitution of India & Professional Ethics	1	0	0	1	1
7	HUM	Ability Enhancement Course	HU203	Design Thinking& Innovation	1	0	0	1	1
B. PRACTICAL									
1	ENGG	Major	CS291	Data Structure & Algorithm Lab	0	0	3	3	**1.5**
2	ENGG	Major	CS293	Digital Electronics and Computer Organization	0	0	3	3	**1.5**
3	SCI	Multidisciplinary	CH291	Engineering Chemistry Lab	0	0	2	2	**1**

4	SCI	Skill Enhancement Course	ME293	IDEA Lab Workshop	0	0	3	3	**1.5**
5	ENGG	Minor	CS292	Introduction to Artificial Intelligence Lab	0	0	3	3	**1.5**
C.MANDATORY ACTIVITIES / COURSES									
1	Mandatory Course	MC281	NSS/ Physical Activities / Meditation & Yoga / Photography/ Nature Club		0	0	0	0	**0**
Total of Theory, Practical								29	**22**
TOTAL FIRST YEAR CREDIT									**40**

Course Name: Data Structure and Algorithms

Course Code: CS201

Contact (Periods/Week): 3 periods

Total Contact Hours:36

Credits:3

Prerequisite:

1. Familiarity with the fundamentals of other programming languages.
2. A solid background in mathematics, including probability, set theory.

Course Objective: By the end of this course, students will be able to:

- Gain a strong foundation in data abstraction, datatypes, and data structures, and the importance of structured data organization in solving engineering problems.
- Formulate and analyze algorithms, perform asymptotic analysis using the notation BigO,θ(Theta), and Ω (Omega), and understand the trade-offs between the complexities of time and space.
- Design and implement linear and non-linear data structures such as arrays, linked lists, stacks, queues, trees, heaps, and graphs, and apply them effectively in computational problem-solving.
- Evaluate and compare various searching, sorting, and hashing algorithms based on their performance, and choose appropriate methods for optimized data handling.
- Appreciate the role of data structures in real-world applications, foster a mindset of lifelong learning, and develop the adaptability to utilize modern programming tools and emerging technologies.

Course Outcome(s): After the completion of the course, students will be able to

CO1: Apply fundamental knowledge of datatypes, abstract datatypes, and data structures to analyze real-world computational problems and their memory/time constraints.

CO2: Design and implement linear data structures(arrays, linked lists, stacks, queues) using appropriate programming constructs to solve well-defined problems efficiently.

CO3: Develop recursive algorithms and simulate stack-based computations such as expression conversion and evaluation using appropriate engineering tools.

CO4: Construct and evaluate non-linear data structures (Binary Tree, BST, AVL Tree, heaps, graphs) and associated operations (search, insertion, deletion, traversal) to address complex engineering problems.

CO5: Compare and optimize sorting, searching, and hashing algorithms based on performance analysis and recognize their suitability in dynamic problem contexts to support lifelong learning.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1	3	3												
CO2	3	2	3		3									
CO3	2	2	3		3									
CO4	3	3	2	3	3									
CO5	3	3			2						3			

Course Content

Module1: Introduction(4L)

- Concepts of Data and Information
- Abstract Data Type(ADT),Data Structure and Data Type
- Classification of Data Structures:
  - Primitive and Non-Primitive Data Structures
  - Linear and Non-Linear Data Structures
- Need for Data Structures
- Concept of Algorithms and Programs
- Methods of Representing Algorithms
- Algorithm Analysis: Time and Space Complexity
- Asymptotic Notations:
  - BigOhO
  - SmallOho
  - BigOmegaΩ
  - SmallOmegaω
  - ThetaΘ

  (Definition and Significance)

Module 2: Non-Restricted Linear Data Structure(9L)

- List or Linear List:
  - Definition, Example, ListasADT
  - Sequential and Linked Representation
- Array:
  - Sequential Representation, Linearization of Multidimensional Array
  - Applications: Polynomial Representation, Sparse Matrix Representation

- LinkedList:
  - Introduction, Implementation of:
  - ∗Singly LinkedList
  ∗Doubly LinkedList
    ∗Circular LinkedList
    ∗Circular Doubly LinkedList
  - Application: Polynomial Representation

Module 3: Restricted Linear Data Structure(6L)

- Stack:
  - Definition and Implementation using Array and Linked List
  - Applications: Infix to Postfix Conversion, Postfix Evaluation
- Recursion:
  - Principles, Use of Stack, Tail Recursion
  - Tower of Hanoi Problem
- Queue:
  - Definition, Array Implementation(Physical, Linear, Circular Models)
  - Linked List Implementation
  - Dequeue–Definition and Types

Module 4: Non-linear Data Structures(9L)

- Trees and Binary Trees:
  - Basic Terminologies, Differences between Tree and Binary Tree
  - Representations(Array and LinkedList)
  - Traversals: Pre-,In-, Post-order
  - Threaded Binary Tree: Definition, Insertion, Deletion
  - Binary Search Tree (BST): Definition, Insertion, Deletion, Searching
  - AVLTree(HeightBalancedTree):Definition,Insertion,Deletion(ExamplesOnly)
- m-Way Search Tree:
  - B Tree: Definition, Insertion, Deletion(Examples Only)
  - B+ Tree: Definition, Insertion, Deletion(Examples Only)
- Heap:
  - Definition(Min and Max Heap), Creation, Insertion, Deletion
  - Application: Priority Queue, Sorting

- Graphs:
  - Definition,Representations(AdjacencyMatrix,IncidenceMatrix,AdjacencyList)
  - Traversals: Depth-FirstSearch(DFS),Breadth-FirstSearch(BFS)
  - Edge Types in DFS/BFS: Tree, Back, Cross, Forward Edge

Module 5: Sorting and Searching(8L)

- Sorting:
  - Definition, Need for Sorting
  - Types: Internal, External, Stable, In-place, Comparison-based
  - Factors Affecting Sorting
  - Algorithms(with Time Complexity):
    - Bubble Sort
    - Insertion Sort
    - Selection Sort
    - Quick Sort
    - Merge Sort
    - Radix Sort
- Searching:
  - Factors Affecting Searching
  - Sequential Search with and without Sentinel(Time Analysis)
  - Binary Search and Interpolation Search (Time Analysis)
- Hashing:
  - Introduction, Purpose of Hashing
  - Hash Functions: Division, Folding, Mid-Square
  - Collision Resolution Techniques

Textbooks

1. Samiran Chattopadhyay, Debabrata Ghosh Dastidar, Matangini Chattopadhyay, *Data Structures Through'C Language*, BPB Publications, Edition:2001.
2. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, *Fundamentals of Data Structures in C*, 2nd Edition, Universities Press.

Reference Books

1. ThomasA.Standish, *Data Structures, Algorithms, and Software*

Principles in C, 1st Edition, Pearson.
2. SeymourLipschutz, *Data Structures*, Special Indian Edition, Tata McGraw-Hill Education.
3. RobertL.Kruse,BruceP.Leung, *Data Structures and Program Design in C*, 2nd Edition, Pearson.
4. AaronM.Tenenbaum, *Data Structures in C*, 1stEdition, Pearson.

Course: Name: Introduction to Artificial Intelligence

Course Code: CS202

Contact (Periods/Week):2 periods

Total Contact Hours:30

Credits:2

Prerequisite: Basic Computer Knowledge.

Course Objective: The objective of the course is to make the students able to–

- Comprehend the fundamental concepts of Knowledge Representation and Inferencing in Artificial Intelligence and its utilitarian importance in current technological context.
- Formulate a problem as a State-Space Exploration Framework or an Inferencing Framework of Artificial Intelligence.
- Use the strategies of AI-Heuristics to find acceptable solutions, avoiding brute-force techniques.
- Evaluate and compare various searching, sorting, and hashing algorithms based on their performance, and choose appropriate methods for optimized data handling.
- Design AI-Frameworks for Inferencing based on knowledge base.
- Analyse the effectiveness of AI-Inferencing Model in offering solutions to the respective problem.

Course Outcome(s):

After the completion of the course student will be able to

CO1: Understand and explain the fundamental concepts of Knowledge Representation and Inferencing in Artificial Intelligence and its utilitarian importance in current technological context for further exploration leading towards lifelong learning.

CO2: Identify and formulate an engineering problem primarily to fit a State-Space Exploration Framework or an Inferencing Model/Agent Design Framework within the scope of the Artificial Intelligence paradigm.

CO3: Explore relevant literature and apply the concept of Heuristic Techniques of Artificial Intelligence to solve problems.

CO4: Develop Inferencing Models for proposing solutions to the problems of Artificial Intelligence.

CO5: Implement Inferencing Models of Artificial Intelligence through developing feasible algorithms and investigate their effectiveness by analysing their performances in solving the relevant problems.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1	3	2									3	3		
CO2	2	3											3	
CO3	2	2	3	2								2		2
CO4	2	2	2	3							2	2		2
CO5	2	2	3	3	2						2	2	2	3

Course Content

Module 1:IntroductiontoArtificialIntelligence(3Lectures)

- Why AI
- Definition of AI
- Goals of AI
- History and evolution of AI
- Types of AI: Narrow, General, Super
- Human vs Artificial Intelligence
- Applications of AI in various domains
- AI for social good

Module 2: Intelligent Agents and Logic-Based Thinking(8 Lectures)

- Intelligent systems
- Agents and environments
- Decision-making using rules and logic
- Symbolic AI concepts
- Propositional Logic: Knowledge Representation and Inference
- Predicate Logic: Knowledge Representation, Inference, and Answer Extraction

Module 3:Overview of AI Branches and Perception(8Lectures)

- Machine Learning
- Deep Learning
- Natural Language Processing
- Computer Vision
- Expert Systems
- Fuzzy Logic
- Evolutionary Algorithms
- Reinforcement Learning

- Planning and Scheduling
- Human-AI Collaboration

Module 4:Basics of Machine Learning(6Lectures)

- What is Machine Learning
- AI vs ML
- Types of learning: Supervised, Unsupervised
- Concept of dataset, features, and labels
- ML model and prediction flow
- Common ML applications
- Introduction to decision trees(concept only)
- ML pipeline overview

Module 5:Applications and Ethics of AI(5Lectures)

- AI in robotics and automation
- AI-enabled smart applications
- Industry 4.0 and intelligent systems
- AI in different sectors: Healthcare, Agriculture, Transport, Education, etc.
- Human-AI teamwork
- Basics of AI Ethics: Bias, Fairness, Privacy

- Career opportunities and future scopes in AI

Text Books

1. Saptarsi Goswami, Amit Kumar Das, Amlan Chakrabarti, *AI for Everyone:ABeginner'sHandbook for Artificial Intelligence*, Pearson.
2. Rich,E.,Knight,K.,Shankar,B.,*ArtificialIntelligence*,TataMcGraw Hill,3rdEdition,2009.
3. Russell,S.,Norvig,P.,*ArtificialIntelligence-AModernApproach*,PrenticeHall,3rdEdition,2015.

Reference Books

1. ReemaThareja, *Artificial Intelligence: Beyond Classical AI*, Pearson.
2. Patterson,*IntroductiontoArtificialIntelligenceandExpertSystems*,Pearson.

Course Name: Digital Logic and Computer Organization

Course Code: CS203

Contact (Periods/Week): 3 periods

Total Contact Hours:36

Credits:3

Prerequisite: Basic knowledge of programming fundamentals.

Course Objective: By the end of this course, students will be able to:

- To introduce number systems, logic gates, and design of combinational and sequential circuits.
- To develop an understanding of data processing using micro-operations and instruction formats.
- To explain how CPU, memory, and I/O units are organized and interact during instruction execution.
- To describe arithmetical algorithms and control unit designs in processor architecture.
- To build a foundation for advanced topics like microprocessors, computer architecture, and embedded systems.

Course Outcome(s): After the completion of the course, students will be able to

CO1: Explain various number systems and coding schemes, and apply Boolean algebra laws and Karnaugh maps to simplify logical expressions.

CO2: Design and construct combinational and sequential logic circuits, including adders, multiplexers, flip-flops, and counters for implementing digital functions.

CO3:Develop and analyze data path units such as ALU, control units, and register organizations to support instruction execution in CPU architectures.

CO4: Demonstrate arithmetic operations like Booth's multiplication and division, and illustrate various addressing modes and instruction formats used in CPUs.

CO5:Compare memory hierarchy systems and I/O techniques, and evaluate their role in enhancing overall processor performance.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1	3	3			2									
CO2	3	3	3		3									
CO3	3	2			3									
CO4	3	2			2									
CO5	3	2			2					2	3			

Course Content

Module 1: Number Systems, Boolean Algebra, and Logic Simplification (6 Lectures)

- Binary, BCD, ASCII, EBCDIC, Gray Code, and conversions [1L]
- Boolean Algebra–Laws, Theorems[1L]
- Boolean Functions, Minterm and Maxterm, SOP and POS Forms [2L]
- Karnaugh Map(upto4-variable),Algebraic Simplification[2L]

Module2 :Combinational Circuits(6Lectures)

- Half and Full Adder/Subtractor, Serial and Parallel Adders, CLAAdder[2L]
- Parity Generator, Encoder, Decoder, Multiplexer, Demultiplexer[2L]
- Comparator, Code Converters[2L]

Module3:Sequential Circuits and Registers(6Lectures)

- Flip-Flops: SR ,JK, Master-SlaveJK, D,T; Characteristic and Excitation Tables [2L]
- Counters: Synchronous/Asynchronous, Ring and Johnson, Mod-N Counters [2L]
- Registers:SISO,SIPO,PIPO,PISO[1L]
- Applications of Counters and Registers[1L]

Module 4:Data Representation and Arithmetic Operations (5 Lectures)

- Integer Arhmetic(Addition,Subtraction),Booth'sMultiplicationAlgorithm[2L]

- Restoring and Non-Restoring Division[1L]
- Instruction Formats and Addressing Modes[2L]

Module 5: CPU and Control Unit Organization(6Lectures)

- Register Transfer Language(RTL), Bus Architecture, Micro-operations[1L]
- ALU Design, Status Flags, General Register and Stack Organization [2L]
- Control Unit: Hardwired vs. Microprogrammed Control, Sequencing[2L]
- Basic Instruction Cycle and Execution Pipeline[1L]

Module 6: Memory and I/O Organization(7Lectures)

- RAM, ROM Types, Memory Hierarchy: Cache, Main, Secondary[1L]
- Cache Mapping: Direct, Associative, Set-Associative; Write Policies[3L]
- Virtual Memory: Paging, Segmentation, FIFO and LRU[1L]
- I/OTransferModes:ProgrammedI/O,Interrupt-DrivenI/O,DMA[1L]
- Interrupts: Maskable/Non-Maskable, Daisy Chaining; I/O Processor[1L]

Textbooks

1. *Digital Logic and Computer Design*, M.Morris Mano, Pearson Education,1st Edition.
2. *Computer Organization and Architecture: Designing for Performance*, William Stallings, Pearson Education, 10th Edition.

Reference Books

1. *Digital Design*, M.Morris Mano, MichaelD. Ciletti, Pearson Education, 5thEdition.
2. *Computer Organization and Embedded Systems*, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, McGraw- Hill Education, 6th Edition.
3. *Computer Organization and Design: The*

Hardware/Software Interface, David A. Patterson, John L. Hennessy, Morgan Kaufmann Publishers, RISC-V Edition.
4. *Fundamentals of Logic Design*, Charles H.RothJr., LarryL.Kinney, Cengage Learning,7thEdition.
5. *Digital Fundamentals*, ThomasL.Floyd, Pearson Education, 11th Edition.

Course Name: Engineering Chemistry

Course Code: CH201

Contact (Periods/Week): 2 periods

Total Contact Hours:24

Credits:2

Prerequisite: 10+2

Course Objective:

- Understand the basic principles of atomic structures and periodic properties of elements, different engineering materials, and advanced polymers.
- Apply the knowledge of free energy, energy storage devices, and semiconductors to design environment-friendly and sustainable devices.
- Apply the concept of corrosion and fuel to improve its efficacy and application for industrial purposes.
- Analyze the organic reaction with the structure of organic molecules by applying the knowledge of different spectroscopic techniques.
- Evaluate the electrical, optical, and structural properties of semiconductors to analyze their potential applications in modern electronic and energy devices.

Course Outcome(s): After the completion of the course, students will be able to

CO1: Understand the basic principles of atomic structures and periodic

properties of elements, different engineering materials, and advanced polymers.
CO2: Apply the knowledge of free energy, energy storage devices, and semiconductors to design environment-friendly and sustainable devices.
CO3: Utilize the concept of corrosion and fuel to improve its efficacy and application for industrial purposes.
CO4: Analyze the organic reaction with the structure of organic molecules by applying the knowledge of different spectroscopic techniques.
CO5: Evaluate the electrical, optical, and structural properties of semiconductors to analyze their potential applications in modern electronic and energy devices.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1	3													
CO2	3								2		2			
CO3	3								2		2			
CO4	3	3							2		2			
CO5	3	3	3						2		2			

Course Content

Module 1(6Lectures)

Quantum Properties of Atoms(4L)

- Schrodinger Wave Equation(time independent–basic principles only)
- de Broglie Equation
- Heisenberg Uncertainty Principle
- Quantum numbers
- Effective nuclear charge, Slater’s rule
- Penetration of orbitals, variation of orbital energies in the periodic table
- Atomic and ionic sizes, ionization energies, and electronegativity
- Oxidation properties

Chemistry of Materials(2L)

- Semiconductor-Based Memory Materials (Si &Ge): Introduction, Properties, and Roles
- Intensive and Extensive Semiconductor Properties

Module 2(7Lectures)

Chemical Thermodynamics(5L)

- First and Second Law of Thermodynamics
- Tendency for maximum randomness
- Carnot Heat Engine(Derivation)
- Entropy characteristics, mathematical explanation, and physical significance
- Entropy change of ideal gas for isothermal reversible process
- Gibbs free energy function, Standard free energy
- Criterion of spontaneity

Electricity Production through Chemical Reactions(2L)

- Electrochemical Cell, cell notation
- Free energy and EMF, Criterion of spontaneity
- Nernst Equation(only expression, no derivation)and applications
- EMF of a cell, single-electrode potential, calculation of Kc and Kc from ΔG^0
- Working principle and applications of Lithium-ion batteries

Module3(6Lectures)

Polymers for Engineering Applications(3L)

- Classification of polymers (based on origin, chemical structure, polymeric structure, tacticity, and molecular forces)
- Synthesis and applications of Bakelite, Nylon6,6,HDPE and LDPE
- Conducting polymers–types, examples, and applications
- Biodegradable polymers–definition, examples, and uses

Industrial Chemistry(3L)

- Types of corrosion, Electrochemical theory of corrosion, rusting of iron
- Comparison of chemical and electrochemical corrosion(mechanisms excluded)
- Factors affecting corrosion rate: nature of metal(physical state, purity, position in Galvanic series), environment

- Corrosion control methods: Cathodic protection, Anodic protection, Inorganic coatings
- Classification of fuels (LPG, CNG, BIOGAS), Definitions of calorific value, Octane number, Cetane number, HCV, LCV

Module4(5Lectures)

Organic Reactions and Drug Synthesis(3L)

- Comparison of acidity and basicity of acids, alcohols, and amines
- Nucleophilic substitution and electrophilic addition reactions
- Markovnikov's rule, peroxide effect
- Synthesis and uses of Paracetamol and Aspirin(Name reactions not included)

Spectroscopy(2L)

- Electromagnetic spectrum, Lambert-Beer Law
- Finding of λ max value & concentration of the unknown solution,
- Applications of UV-VIS spectroscopy, Chromophores and Auxochromes
- Applications of IR spectroscopy, Fingerprint region

Text Books

1. *Chemistry–I*, Gourkrishna Das Mohapatro
2. *A Textbook of Engineering Chemistry*, Dr.Rajshree Khare
3. *Engineering Chemistry*,U.N.Dhar
4. *Physical Chemistry*,P.C.Rakshit

Reference Books

1. *Engineering Chemistry*, Jain&Jain
2. *Engineering Chemistry(NPTELWeb-book)*, B.L.Tembe, Kamaluddin and M.S.Krishna
3. *Textbook of Engineering Chemistry*, JayaShreeAnireddy

Course Name: Engineering Mathematics

Course Code: M201

Contact (Periods/Week): 3 periods

Total Contact Hours:36

Credits:3

Prerequisite: The students to whom this course will be offered must have the understanding of (10+2) standard algebraic operations, coordinate geometry, and elementary calculus concepts including limits, continuity, differentiation, and integration.

Course Objective:

- Develop a thorough understanding of ordinary differential equations and their role in modeling real-world systems.
- Build competency in applying the Laplace transform as a tool for solving initial value problems and linear differential equations in engineering contexts.
- Gain proficiency in numerical techniques for solving mathematical problems where analytical methods are difficult or impossible.

Course Outcome(s): After the completion of the course, students will be able to

CO1: Apply analytical methods to solve ordinary differential equations in engineering contexts.
CO2: Apply the properties and inverse of Laplace Transforms to compute improper integrals and determine solutions of linear ordinary differential equations with constant coefficients in engineering scenarios.
CO3: Apply numerical methods to interpolate data, perform numerical integration, and solve ordinary differential equations in engineering applications.
CO4: Analyze the behavior of solutions using analytical and numerical approaches, including Laplace transforms, to assess stability, convergence, and accuracy in engineering contexts.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1	3	2									1			

CO2	3	2								1			
CO3	3	2								1			
CO4	3	3	1	1						2			

Course Content

Module I: First Order Ordinary Differential Equations(9L)

Solution of first order and first-degree ODE: Exact ODE, rules for finding integrating factors, linear ODE, Bernoulli's equation. Solution of first-order and higher-degree ODE: solvable for p, solvable for y and x, and Clairaut's equation.

Module II: Second Order Ordinary Differential Equations(8L)

Solution of second-order ODE with constant coefficients: complementary function (CF), particular integral (PI), method of variation of parameters, and Cauchy-Euler equations.

Module III: Laplace Transform(12L)

Concept of improper integrals; definition and existence of Laplace Transform (LT); LT of elementary functions. First and second shifting properties, change of scale property, LT of tf (*t*) and f(*t*)*t*, LTof derivatives and integrals of *f*(*t*). Evaluation of improper integrals using LT, LT of periodic and step functions. Inverse LT: definition and properties. Convolution theorem (statement only) and its application. Solution of linear ODE with constant coefficients (initial value problem) using LT.

Module IV: Numerical Methods(7L)

Error analysis, calculus of finite differences.
Interpolation: Newton forward and backward, Lagrange's interpolation.
Numerical integration: trapezoidal rule, Simpson's $\frac{1}{3}$rule.
Numerical solution of ODE: Euler method, fourth-order Runge-Kutta method.

Text Books

1. *Higher Engineering Mathematics*, B.S.Grewal, Khanna Publishers, 36th Edition,2010.
2. *Advanced Engineering Mathematics*, E.Kreyszig, 9th Edition, JohnWiley & Sons, 2006.

Reference Books

1. *A Textbook of Engineering Mathematics I*,S.Guruprasad, NewAge International Publishers.
2. *Higher Engineering Mathematics*, B.V.Ramana, TataMcGraw-Hill,11thReprint,2010.
3. *Engineering Mathematics for First Year*, T.Veerarajan, TataMcGraw-Hill,2008.
4. *A Textbook of Engineering Mathematics*, N.P.BaliandM. Goyal, Laxmi Publications, Reprint,2008.
5. *Calculus and Analytic Geometry*, G.B. Thomas and R.L.Finney, 9thEdition, Pearson,2002.
6. *Calculus(Vol.1and2)*, M.Apostol, 2nd Edition, Wiley Eastern,1980.
7. *Linear Algebra: A Geometric Approach*, S.Kumaresan, Prentice Hall of India,2000.
8. *Linear Algebra: A Modern Introduction*, D.Poole, 2nd Edition, Brooks/Cole,2005.
9. *Schaum's Outline of Matrix Operations*, R.Bronson,1988.
10. *Differential and Integral Calculus (Vol.I&II)*, N.Piskunov, MirPublishers,1969.

2nd Year 3rd Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credits**
A. THEORY					L	T	P	Total	
1	ENGG	Major	DS301	Operating Systems	3	0	0	3	**3**
2	ENGG	Minor	DS302	Design and Analysis of Algorithms	3	0	0	3	**3**
3	ENGG	Major	DS303	Data Handling and Visualization	3	0	0	3	**3**
4	SCI	Minor	M(DS)301	Algebra and Differential Equations	3	1	0	3	**3**
B. PRACTICAL									
1	ENGG	Major	DS391	Operating Systems Lab	0	0	3	3	**1.5**
2	ENGG	Minor	DS392	Design and Analysis of Algorithms Lab	0	0	3	3	**1.5**
3	ENGG	Major	DS393	Data Handling and Visualization Lab	0	0	3	3	**1.5**
5	ENGG	Skill Enhancement Course	DS394	Introduction to Python Programming Lab	0	1	3	4	**2.5**
		Total of Theory & Practical						**25**	**19**

Course Name: Operating System
Course Code: DS301
Contact (L: T: P): 3:0:0
Total Contact Hours: 36
Credit: 3

Prerequisites:
1. Computer organization
2. Computer Architecture
3. Data Structures
4. Algorithms & Programming Concept

Course Objectives:
The objective of the course is to make the students able to -
1. Students will learn how Operating System is Important for Computer System.
2. To make aware of different types of Operating System and their services.
3. To learn different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
4. To know virtual memory concepts.
5. To learn secondary memory management.

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

CO1	**Understand** the fundamental concepts of Operating System, Protection & Security and differentiate different types of Operating System.
CO2	**Understand** and implement process & thread; understand, apply, compare different process synchronization algorithm and inter process communication to solve engineering problems
CO3	**Explain/analyze** different synchronization techniques, critical section problems and deadlock and apply them to solve engineering problems.
CO4	**Explain** different memory management techniques including virtual memory management; also, able to apply, compare, and implement different page replacement algorithms to solve engineering problems.
CO5	**Explain** different I/O mechanisms, File structures and disk management techniques and solving engineering problems applying different disk scheduling algorithms.

Course Content:

Module – 1: [3L]

Functionalities of Operating System, Evolution of Operating System.
Types of Operating System: batch, multi-programmed, time-sharing, real-time, distributed, parallel, Structural overview, Protection & Security. [3L]

Module – 2: [11L]

Processes: Concept of processes, process states, PCB, process scheduling, co-operating processes,
independent process, suspended process, Interaction between processes and OS, Inter-process communication: Message passing. [3L],
Threads: overview, benefits of threads, user and kernel level threads, Thread models. [2L]
CPU scheduling: Scheduling criteria, preemptive & non-preemptive scheduling, scheduling algorithms
(FCFS, SJF, SRTF, RR, priority, multilevel queue, multilevel feedback queue scheduling). [6L]

Module 3: [11L]

Process Synchronization: background, critical section problem, synchronization hardware, classical problems of synchronization (producer-consumer, readers-writer, dining philosophers, etc), semaphores, monitors. [6L]
Deadlocks: deadlock characterization, methods for handling deadlocks, deadlock prevention, deadlock
avoidance, deadlock detection, recovery from deadlock. [5L]

Module 4: [6L]

Background, logical vs. physical address space, swapping, contiguous memory allocation, paging,
Segmentation, TLB. [3L], Virtual Memory: background, demand paging, page replacement algorithms (FCFS, LRU, Optimal), thrashing, Working set model. [3L]

Module 5: [5L]

Disk structure, disk scheduling (FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK etc), disk reliability, disk formatting, boot block, bad blocks. [2L]
File: File concept, access methods, directory structure, file system structure, UNIX file structure, allocation methods (contiguous, linked, indexed), free-space

management (bit vector). [2L]
I/O: I/O hardware, polling, interrupts, DMA, caching, buffering, blocking-non blocking I/O. [1L]

CO-PO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PSO 1	PSO 2	PSO 3
CO 1	3	3	3	3	3	1	1	1	3	2	2	3	3	3
CO 2	3	3	3	3	3	1	1	1	3	2	2	3	3	3
CO 3	3	3	3	3	3	1	1	1	3	2	2	3	3	3
CO 4	3	3	3	3	3	1	1	1	3	2	2	3	3	3
CO 5	3	3	3	3	3	1	1	1	3	2	2	3	3	3

1= LOW, 3= HIGH

Textbooks:

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts.
2. Operating Systems & Systems Programming by P Balakrishna Prasad

Reference Book:

1. Dietel H. N., ―An Introduction to Operating Systems‖, Addison Wesley.
2. Andrew Tanenbaum, Modern Operating Systems, Prentice Hall.
3. William Stallings, Operating Systems, Prentice Hall.

Course Name: Design and Analysis of Algorithms
Course Code: DS302
Contact (L: T: P): 3:0:0
Total Contact Hours: 36
Credit: 3

Prerequisites:

1. Data Structures
2. Discrete Mathematics

3. Programming Concepts (C/C++/Java)

Course Objectives:

The objective of the course is to make the students able to -

1. To introduce fundamental techniques and concepts in the design and analysis of algorithms.
2. To teach efficient problem-solving approaches using divide-and-conquer, greedy methods, dynamic programming, and backtracking.
3. To help students analyze the time and space complexity of algorithms.
4. To explore algorithmic strategies for graph-related problems.
5. To introduce concepts of NP-completeness and approximation algorithms.

Course Outcomes (COs):

After completion of the course students will be able to

CO1	**Understand** and analyze asymptotic notations, recurrence relations, and algorithm complexity.
CO2	**Apply** divide-and-conquer and greedy approaches for solving optimization problems.
CO3	**Implement** and **analyze** algorithms using dynamic programming and backtracking strategies.
CO4	**Apply** graph algorithms to solve real-world and computational problems.
CO5	**Understand** NP-completeness and use approximation methods for hard problems.

Course Content:

Module – 1: Introduction and Analysis [8L]

Introduction to Algorithms: Definition, characteristics of an algorithm, importance of algorithm analysis. [2L], Asymptotic Notations: Big-O, Omega, Theta notations, little-o, little-omega. [3L], Recurrence Relations: Solving recurrence relations using the substitution method, recursion tree method, and Master theorem. [3L]

Module – 2: Divide and Conquer & Greedy Algorithms [9L]

Divide and Conquer: General method, binary search, merge sort, quick sort,

Strassen's matrix multiplication. [5L], Greedy Algorithms: General method, activity selection problem, Huffman coding, fractional knapsack problem, Dijkstra's algorithm. [4L]

Module – 3: Dynamic Programming [8L]

General method, optimal substructure, overlapping subproblems. [2L], Applications: Matrix chain multiplication, longest common subsequence, 0/1 knapsack problem, Floyd-Warshall algorithm, Bellman-Ford algorithm. [6L]

Module 4: Graph Algorithms [7L]

Graph Traversals: Breadth-First Search (BFS), Depth-First Search (DFS). [2L], Minimum Spanning Trees: Prim's algorithm, Kruskal's algorithm. [3L], Single-Source Shortest Paths: Bellman-Ford, Dijkstra's algorithm. [2L]

Module 5: Advanced Topics [4L]

Backtracking: N-Queens problem, Sum of subsets. [1L], Branch and Bound: Traveling Salesperson Problem, Assignment problem. [1L], NP-Completeness: P, NP, NP-hard, NP-complete classes, Cook's theorem (brief introduction). [1L], Approximation Algorithms: Introduction and examples. [1L]

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	2	2	1	1	1	2	1	2	3	3	2
CO2	3	3	3	3	3	1	1	1	2	2	2	3	3	2
CO3	3	3	3	3	3	1	1	1	2	2	2	3	3	2
CO4	3	3	3	3	3	1	1	1	2	2	2	3	3	2
CO5	3	3	2	2	2	1	1	1	2	1	2	3	3	2

1= LOW, 3= HIGH

Text Book:

1. "Introduction to Algorithms" by Cormen, Leiserson, Rivest, Stein.
2. "The Design and Analysis of Computer Algorithms" by Aho, Hopcroft, Ullman.
3. "Algorithm Design" by Kleinberg and Tardos.

4. Design & Analysis of Algorithms, Gajendra Sharma, Khanna Publishing House, New Delhi

Reference Book:

1. Design Analysis and Algorithms‖ by Hari Mohan Pandey.

Course Name: Data Handling and Visualization
Course Code: DS303
Contact (L: T: P): 3:0:0
Total Contact Hours: 36
Credit: 3

Prerequisites:

1. Basics knowledge of Mathematics.
2. Basic programming knowledge.

Course Objectives: The objective of the course is to make the students able to –

1. Master fundamental data analysis techniques
2. Develop effective data visualization skills
3. Apply data analysis to real-world problems
4. Understand data integrity and ethical considerations
5. Utilize industry-standard tools and software

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

CO1	**Understand** the use data analysis tools in the pandas library.
CO2	**Apply** Load, clean, transform, merge, and reshape data.
CO3	**Create** informative visualization and summarize data sets.
CO4	**Analyse** and manipulate time series data.
CO5	**Solve** real-world data analysis problems.

Course Content:

Module-1

Introduction: Introduction to Data Science, Exploratory Data Analysis, and Data Science Process. Motivation for using Python for Data Analysis, Introduction of Python shell iPython, and Jupyter Notebook. Essential Python Libraries: NumPy, pandas, matplotlib, SciPy, scikit-learn, stats models

Module-2

Getting Started with Pandas: Arrays and vectorized computation, Introduction to pandas Data Structures, Essential Functionality, Summarizing, and Computing Descriptive Statistics. Data Loading, Storage, and File Formats. Reading and Writing Data in Text Format, Web Scraping, Binary Data Formats, interacting with Web APIs, Interacting with Databases Data Cleaning and Preparation. Handling Missing Data, Data Transformation, and String Manipulation.

Module-3

Data Wrangling: Hierarchical Indexing, Combining and Merging Data Sets Reshaping and Pivoting.

Data Visualization matplotlib: Basics of matplotlib, plotting with pandas and seaborn, and other Python visualization tools

Module-4

Data Aggregation and Group operations: Group by Mechanics, Data aggregation, General split- apply-combine, Pivot tables, and cross tabulation.

Time Series Data Analysis: Date and Time Data Types and Tools, Time series Basics, date Ranges, Frequencies and Shifting, Time Zone Handling, Periods and Periods Arithmetic, Resampling and Frequency Conversion, and Moving Window Functions.

Module-5

Advanced Pandas: Categorical Data, Advanced Group By Use, Techniques for Method Chaining

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	1	1	1	3	2	2	3	3	3
CO2	3	3	3	3	3	1	1	1	3	2	2	3	3	3
CO3	3	3	3	3	3	1	1	1	3	2	2	3	3	3
CO4	3	3	3	3	3	1	1	1	3	2	2	3	3	3

CO 5	3	3	3	3	3	1	1	1	3	2	2	3	3	3

1= LOW, 3= HIGH

Text Books:

1. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and I Python. 2nd edition, McKinney, W. O'Reilly Media.
2. Doing Data Science: Straight Talk from the Frontline, O'Neil, C., & Schutt, R., O'Reilly Media.

Reference Books:

1. R cookbook, Teetor, P., O'Reilly. ISBN 9780596809157.
2. R graphics cookbook. Chang, W., O'Reilly. ISBN 9781449316952.
3. Discovering Statistics Using R., Andy Field, Jeremy Miles and Zoe Field., SAGE Publications Ltd. ISBN-13: 978-1446200469.
4. An Introduction to Statistical Learning with Applications in R, Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, Springer.

Web Reference:

1. https://www.openintro.org/stat/

Course Name: Algebra and Differential Equations
Paper Code: M(DS)301
Contact (L: T: P): 3: 1: 0
Total Contact Hours: 48
Credit: 4

Prerequisites: The students to whom this course will be offered must have the concept of (10+2) standard matrix algebra, and calculus.

Course Objectives:

1. To develop a solid understanding of matrix algebra including operations, determinants, rank, and their role in solving systems of linear equations.
2. To introduce vector spaces and linear transformations, enabling students to understand concepts like basis, dimension, and eigenvalues/eigenvectors.
3. To familiarize students with first and higher-order differential equations, their classifications, and analytical methods of solving them.
4. To enable students to apply differential equations in modeling physical systems, such as population growth, mechanical vibrations, and electrical

circuits.

5. To strengthen problem-solving skills through practical examples, integrating algebraic and differential techniques in engineering and scientific contexts.

Course Outcomes (COs): On successful completion of the learning sessions of the course, the learner will be able to:

CO1	Recall the properties related to Linear algebra and Differential and Integral Calculus.
CO2	Determine the solutions to the problems related to Linear Algebra and Differential Equations.
CO3	Apply the appropriate mathematical tools of Linear Algebra and Differential Equations for the solutions of real-life problems.
CO4	Analyze different engineering problems with the help of Linear Algebra and Differential Equations.
CO5	Understand the strengthen problem-solving skills through practical examples, integrating algebraic and differential techniques in engineering and scientific contexts.

Course Content:

MODULE-1: System of Linear Equations [5L]
Gaussian elimination and Gauss Jordan methods, Elementary matrices, System of linear equations, LU factorizations.

MODULE-2: Vector Spaces [8L]
The Euclidean space and vector space, subspace, linear combination, span linearly dependent- independent, bases, dimensions, finite dimensional vector space, Replacement, Deletion and Extension of Bases.

MODULE-3: Linear Transformations and applications [12L]
Linear transformations, Basic properties, Row and column spaces, Rank and nullity, Bases for subspace, invertible linear transformation, matrices of linear transformations, vector space of linear transformations, change of bases, similarity.

MODULE-4: Inner Product Spaces [6L]
Dot products and inner products, the lengths and angles of vectors, matrix representations of inner products, Gram-Schmidt orthogonalization.

MODULE 5: First Order Ordinary Differential Equations (ODE) (9L)
Solution of first order and first-degree ODE: Exact ODE, Rules for finding Integrating factors, Linear ODE, Bernoulli's equation. Solution of first order and higher degree ODE: solvable for p, solvable for x, and solvable for y, and Clairaut's equation.

MODULE 6: Second Order Ordinary Differential Equations (ODE) (8L)
Solution of second-order ODE with constant coefficients: C.F. & P.I., D-operator, Method of variation of parameters, Cauchy-Euler equations. Simultaneous Differential Equation.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	-	-	-	-	-	-	-	1	3	1	-
CO2	3	3	2	1	1	-	-	-	-	-	2	3	2	1
CO3	2	3	3	2	1	1	1	-	-	1	2	2	3	2
CO4	3	2	2	1	2	-	-	-	-	-	1	3	1	
CO5	2	2	2	1	1	-	-	-	-	-	1	2	2	1

1= LOW, 3= HIGH

Text Books:
1. Grewal, B.S., Higher Engineering Mathematics, Khanna Publishers, 36th Edition, 2010.
2. Kreyszig, E., Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
3. Seymour Lipchutz, Marc Lipson, Schaum's Outline of Linear Algebra, 3rd Edition, 2017.

Reference Books:
1. Guruprasad, S. A text book of Engineering Mathematics-I, New age International Publishers.
2. Ramana, B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.

3. Veerarajan, T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
4. Bali, N.P. and Goyal, M., A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2008.
5. Kumaresan, S., Linear Algebra - A Geometric approach, Prentice Hall of India, 2000.
6. Poole, D., Linear Algebra: A Modern Introduction, 2nd Edition, Brooks/Cole, 2005.
7. Bronson, R., Schaum's Outline of Matrix Operations. 1988.

Course Name: Operating Systems Lab
Course Code: DS391
Allotted Hours: 36L
Credit: 1.5

Prerequisites:
1. Computer organization
2. Computer Architecture
3. Data Structures
4. Algorithms & Programming Concepts

Course Outcomes (COs):
After attending the course, students should be able to

CO1	Analyze and simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority.
CO2	Understand the concepts of deadlock in operating systems.
CO3	Implement them in a Multiprogramming system.
CO4	Create a process creation and implement inter-process communication
CO5	Analyze the performance of the various page replacement schemes

Course Contents:

1. Essential Linux Commands[9P]: Commands for files and directories cd, cp, mv, rm, mkdir, more, less, creating and viewing files, using cat, file comparisons, View files, kill, ps, who, sleep, grep, fgrep, find, sort, cal, banner, touch, file related commands – ws, sat, cut, grep etc. Mathematical commands –expr, factor, units, Pipes (use functions pipe, popen, pclose), named Pipes (FIFOs, accessing FIFO)

2. Shell Programming [6P]: Creating a script, making a script executable,

shell syntax (variables, conditions, control structures, functions, and commands).

3. Process [3P]: Starting a new process, replacing a process image, or duplicating a process image.

4. Semaphore [3P]: Programming with semaphores (use functions segment, semop, semaphore_p

5. POSIX Threads[6P]: Programming with pthread functions (viz. pthread_create, pthread_join, pthread_exit, pthread_attr_init, pthread_cancel).

6. Shared Memory [9P]: Create the shared memory, Attach the shared memory segment to the address space of the calling process, read information from the standard input and write to the shared memory, Read the content of the shared memory and write on to the standard output, Delete the shared memory.

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	-	-	-	3	3	-	-	-	-
CO2	3	3	3	3	3	-	-	-	3	3	-	-	-	-
CO3	3	3	3	3	3	-	-	-	3	3	-	-	-	-
CO4	3	3	3	3	3	-	-	-	3	3	-	-	-	-
CO5	3	3	3	3	3	-	-	-	3	3	-	-	-	-

Text Books:

1. Yash Avant P. Kanetkar, UNIX Shell Programming, 1st edition, BPB Publications
2. Beej's Guide to Unix IPC
3.W. Richard Stevens, UNIX Network Programming, 2nd edition, Prentice Hall

Course Name: Design and Analysis of Algorithms Lab
Course Code: DS392
Allotted Hours: 36L
Credit: 1.5

Prerequisites:
1. Computer organization
2. Computer Architecture
3. Data Structures
4. Algorithms & Programming Concepts

Course Outcomes (COs):
After attending the course, students should be able to

CO1	Analyze and simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority.
CO2	Understand the concepts of deadlock in operating systems.
CO3	Implement them in a Multiprogramming system.
CO4	Create a process creation and implement inter-process communication
CO5	Analyze the performance of the various page replacement schemes

Course Contents:

1. Divide and Conquer: Implement Binary Search using Divide and Conquer approach Implement Merge Sort using Divide and Conquer approach
2. Divide and Conquer: Implement Quick Sort using Divide and Conquer approach Find Maximum and Minimum element from an array of integer using Divide and Conquer approach
3. Dynamic Programming: Find the minimum number of scalar multiplications needed for chain of matrix
4. Dynamic Programming: Implement all pair of shortest paths for a graph (Floyed- Warshall Algorithm) Implement Traveling Salesman Problem
5. Dynamic Programming: Implement Single Source shortest Path for a graph (Dijkstra, Bellman Ford Algorithm)
6. Brunch and Bound: Implement 15 Puzzle Problems
7. Backtracking: Implement 8 Queen Problem
8. Backtracking (implement any one of the following problems): Graph Coloring Problem Hamiltonian Problem
9. Greedy method (implement any one of the following problems): Knapsack Problem Job sequencing with deadlines
10. Greedy method (implement any one of the following problems):

Minimum Cost Spanning Tree by Prim's Algorithm, Minimum Cost Spanning Tree by Kruskal's Algorithm

11. Graph Traversal Algorithm: Implement Breadth First Search (BFS) Implement Depth First Search (DFS)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	-	-	-	3	3	-	-	-	-
CO2	3	3	3	3	3	-	-	-	3	3	-	-	-	-
CO3	3	3	3	3	3	-	-	-	3	3	-	-	-	-
CO4	3	3	3	3	3	-	-	-	3	3	-	-	-	-
CO5	3	3	3	3	3	-	-	-	3	3	-	-	-	-

Text Books:

"Introduction to Algorithms" by Thomas H. Carmen, Charles E. Lieberson, Ronald L. Rivest, and Clifford Stein

1. "Algorithms Unlocked" by Thomas H. Carmen
2. "The Algorithm Design Manual" by Steven S. Skiena

Course Name: Data Handling and Visualization Lab
Course Code: DS393
Allotted Hours: 36L
Credit: 1.5

Prerequisites:

1. Basic knowledge of Mathematics.
2. Basic programming knowledge.

Course Objectives:

This course introduces students to data analysis and visualization in the field of exploratory data science using Python.

Course Outcomes (COs):

After completion of the course, students will be able to

CO1	Understand basic of Python Programming Language
CO2	Understand the use of Conditional statement and Loops
CO3	Learn functions in python and represent collection type data using List and Dictionary
CO4	Read and write data from & to files in Python
CO5	Understand NumPy array and numerical operations on Numpy array.

Course Contents:

1. Basics of Python: Python Installation, Python variables, data types, and Operator.
2. Loops: While and For loops, Python Syntax, Colon & Indentation, Conditional Statements: if, elif and else.
3. Functions: Defining Functions in python; passing arguments.
4. String: Python Programming to explore string functions
5. Lists: Python programs using Lists; understand the use of List methods, Slicing on List.
6. Sets: Working with Sets; Write programs to show different set operations.
7. Dictionary: Demonstrate the use of Dictionaries
8. File handling: Reading & Writing data from a file, Redirecting output streams to files.
9. Numpy: Numerical operations using Numpy array; slicing numpy array; stacking numpy arrays.
10. Write programs to show different numerical operations on numpy array.

CO-PO-PSO Mapping

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	2	-	-	2	-	-	-	-	-	-	1	3	3	3
CO2	2	3	-	2	-	-	-	-	-	-	-	3	3	3
CO3	2	3	3	2	-	-	-	-	-	-	1	3	3	3
CO4	2	3	2	2	-	-	-	-	-	-	-	3	3	3
CO5	3	3	3	3	-	-	-	-	-	-	2	3	3	3

Recommended Text Books

1. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. 2nd edition, McKinney, W. O'Reilly Media.
2. Doing Data Science: Straight Talk from the Frontline, O'Neil, C., & Schutt, R., O'Reilly Media.

Recommended Reference Books

1. R cookbook, Teetor, P., O'Reilly. ISBN 9780596809157.
2. R graphics cookbook. Chang, W., O'Reilly. ISBN 9781449316952.
3. Discovering Statistics Using R., Andy Field, Jeremy Miles and Zoe Field., SAGE Publications Ltd. ISBN-13: 978-1446200469.
4. An Introduction to Statistical Learning with Applications in R, Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, Springer.

Course Name: Introduction to Python Programming Lab
Course Code: DS394
Allotted Hours: 36L
Credit: 2.5

Prerequisites: Knowledge of Mathematics and basic concepts of Programming.

Course Objectives:

This course introduces students to data analysis and visualization in the field of exploratory data science using Python.

Course Outcomes (COs):

1. **Understand** fundamental programming concepts: Students will grasp core programming principles, including variables, data types, operators, and control flow, using Python.
2. **Develop** basic Python programming skills: Students will acquire the ability to write, debug, and execute simple Python scripts for various tasks.
3. **Utilize** Python for common data structures: Students will learn to

effectively work with built-in Python data structures like lists, tuples, dictionaries, and sets.

4. **Implement** functions and modules for code organization: Students will understand and apply the concepts of functions, modularity, and basic file I/O for creating reusable and organized code.
5. **Solve** introductory computational problems using Python: Students will be able to apply their Python knowledge to formulate and solve simple computational and logical problems.

Course Contents:

You can use the data set of your choice from Open Data Portal (https://data.gov.in/).

1. Practical based on numpy nd array
2. Practical based on Pandas Data Structures
3. Practical based on Data Loading, Storage and File Formats
4. Practical based on Interacting with Web APIs
5. Practical based on Data Cleaning and Preparation
6. Practical based on Data Wrangling
7. Practical based on Data Visualization using matplotlib

8. Practical based on Data Aggregation
9. Practical based on Time Series Data Analysis

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	2	3	2	2	-	-	-	-	-	-	-	-	-	-
CO2	2	3	2	2	-	-	-	-	-	-	-	-	-	-
CO3	2	3	3	2	-	-	-	-	-	-	-	-	-	-
CO4	2	3	3	2	-	-	-	-	-	-	-	-	-	-
CO5	2	3	3	2	-	-	-	-	-	-	-	-	-	-

Recommended Text Books

1. "Python Crash Course" by Eric Matthes
2. "Automate the Boring Stuff with Python" by Al Sweigart
3. "Learning Python" by Mark Lutz
4. "Python for Everybody: Exploring Data in Python 3" by Dr. Charles R. Severance (often referred to as "Dr. Chuck"

Recommended Reference Books

1. "Think Python: How to Think Like a Computer Scientist" by Allen B. Downey

2nd Year 4th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credits**
A.THEORY					L	T	P	Total	
1	ENGG	Major	DS401	Database Management Systems	3	0	0	3	**3**
2	ENGG	Major	DS402	Computer Networks	3	0	0	3	**3**
3	ENGG	Minor	IT(DS)403	Object Oriented Programming Language	3	0	0	3	**3**
4	SCI	Minor	M(DS)401	Probability and Statistics	3	1	0	4	**4**
5	HUM	Ability Enhancement Course	HU(DS)401	Principles of Management	2	0	0	2	**2**
B.PRACTICAL									
1	ENGG	Major	DS491	Database Management Systems Lab	0	0	3	3	**1.5**
2	ENGG	Major	DS492	Computer Networks Lab	0	0	3	3	**1.5**
3	ENGG	Minor	DS493	Object Oriented Programming Lab	0	0	3	3	**1.5**
4	SCI	Minor	M(DS)491	Numerical Methods Lab	0	0	3	3	**1.5**
5	HUM	Ability Enhancement Course	HU(DS)491	Soft Skill & Aptitude	2	0	0	2	**1**
		TOTAL CREDIT						29	**22**

***Note:** Course from Skill India to be done in consultation with the department. Certificate of completion is to be submitted to the department/COE Section by the end of the 4th semester as the mark of completion of MC481.

Course Name: Database Management Systems
Course Code: DS401
Contact (L: T: P): 3:0:0
Total Contact Hours: 36
Credit: 3
Prerequisites:

1. Logic of programming language
2. Basic concepts of data structure and algorithms

Course Objectives:

The objective of the course is to make the students able to -

1. To understand the different issues involved in the design and implementation of a database system.
2. To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
3. To understand and use data manipulation language to query, update, and manage a database
4. To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.
5. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.
6. To understand the different issues involved in the design and implementation of a database system.

Course Outcomes (COs):

After completion of the course, students will be able to

CO1	Understand and Describe the basic concepts and utility of Database management system, different data models of Database management system.
CO2	Design an Entity Relationship (E-R) Diagram and relational model for any kind of real-life application and able to Apply relational algebra operations, SQL, and Neo4j for solving
CO3	Analyze and Create the relational database for any real-life applications based on normalization.
CO4	Apply the query optimization techniques, different file organization techniques and Determine whether the transaction satisfies the ACID properties.
CO5	Implement and Organize the database of an organization as a team.
CO6	Understand the different issues involved in the design and

	implementation of a database system.

Course Content:

Module-1: Introduction [3L]

Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Three Schema architecture of DBMS.

Module-2: Entity-Relationship and Relational Database Model [11L]

Basic concepts, Design Issues, Mapping Constraints, Keys, Entity-Relationship Diagram, Weak Entity Sets, Extended E-R features, case study on E-R Model. Structure of relational Databases, Relational Algebra, Relational Calculus, Extended Relational Algebra Operations, Views, Modifications of the Database.

Module-3: SQL and Integrity Constraints [6L]

Concept of DDL, DML, DCL. Basic Structure, Set operations, Aggregate Functions, Null Values, Domain Constraints, Referential Integrity Constraints, assertions, views, Nested Subqueries, Database security application development using SQL, Stored procedures and triggers.

Module-4: Relational Database Design [8L]

Functional Dependency, Different anomalies in designing a Database., Normalization using functional dependencies, Decomposition, Boyce-Codd Normal Form, 3NF, Normalization using multi-valued dependencies, 4NF, 5NF, Case Study

Module-5: Internals of RDBMS [9L]

Physical data structures, Query optimization: join algorithm, statistics and cost bas optimization. Transaction processing, Concurrency control and Recovery Management: transaction model properties, state serializability, lock base protocols; two phase locking, Dead Lock handling.

Module-6 File Organization & Index Structures [6L]

File & Record Concept, placing file records on Disk, Fixed and Variable sized Records, Types of Single Level Index (primary, secondary, clustering), Multilevel Indexes

Textbook:

1. Henry F. Korth and Silberschatz Abraham, ―Database System Concepts‖, Mc.Graw Hill.
2. Elmasri Ramez and Novathe Shamkant, ―Fundamentals of Database Systems‖, Benjamin Cummings Publishing.Company.

Reference Book:

1. Fundamentals of Database Systems‖, Ramez Elmasri, Shamkant B.Navathe, Addison Wesley Publishing.
2. Ramakrishnan: Database Management System, Mc Graw Hill.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	1	2	1	1	1	1	1	2	2	2	1
CO2	3	3	2	2	3	1	1	1	1	1	2	3	2	2
CO3	3	3	2	3	3	1	1	1	1	1	2	3	3	2
CO4	3	3	3	3	3	1	1	1	1	1	2	3	3	2
CO5	3	3	3	3	3	2	2	2	2	2	3	3	3	3
CO6	3	3	2	1	2	1	1	1	1	1	2	2	2	1

1= LOW, 3= HIGH

Course Name: Computer Networks
Course Code: DS402
Contact: 3:0:0
Total Contact Hours: 36
Credits: 3

Prerequisite:

1. Familiarity and knowledge of Operating Systems and Computer Architecture.
2. Also require a little bit of programming language concepts like C, Java.

Course Objectives:

The objective of the course is to make the students able to –

Course Outcomes (COs):

After completion of the course, students will be able to

CO1	Understand the basics of computer networks and different architectures and topologies of computer networks, analyze the requirements for a given organizational structure, and select the most appropriate networking architecture and technologies.
CO2	Understand/analyze different protocols of the data link layer and apply them

	to solve engineering problems.
CO3	Understand/analyze different protocols of the Network and Transport Layer and apply them to solve engineering problems.
CO4	Understand and analyze different protocols at the session and application layers, and apply them to solve engineering problems.
CO5	Develop, analyze, and program. specify and design the Topological and routing strategies using socket
CO6	Understand Socket Programming.

Module 1: Introduction [6L]

Introduction (3L):

Introduction: Computer Network, data communication, topology, OSI & TCP/IP Reference Models, layers and characteristics, Wireless Network, comparison to wired and wireless network.

Physical Layer: [3L]

Overview of data (analog & digital), signal (analog & digital), transmission (analog & digital) & transmission media (guided & unguided); Circuit switching: time division & space division switch, TDM bus; Telephone Network;

Module 2: Data Link Layer [10L]

Framing, Error Control, Error Detection and Correction, Flow Control, Data Link Protocols, Simple Stop-and-Wait Protocol, ARQ mechanism, Sliding Window Protocols, One-Bit Sliding Window Protocol, Go-Back-N and Selective Repeat, HDLC, PPP Medium Access Control Sub-layer, The Channel Allocation. [5L]

Multiple Access Protocols: ALOHA, Carrier Sense Multiple Access Protocols, IEEE 802.x Ethernet, Switched Ethernet, Fast Ethernet, Gigabit Ethernet, 10 Gigabit Ethernet, Wireless LANs - IEEE 802.xx, Bluetooth, RFID, Bridges, Virtual LANs, Switching. [5L]

Module 3: Network Layer [10L]

IP Addressing, IPv4 and IPv6. Difference IPv4 and IPv6, Conversion of IPv4 and IPv6, Subnetting, Supernetting, Design Issues, Store-and-Forward Packet Switching, Virtual-Circuit and Datagram Networks, ARP, IP, ICMP, IPV6, BOOTP and DHCP–Delivery protocols, Other Protocols such as mobile IP in wireless networks. [5L]

Routing: Shortest Path Flooding, Distance Vector Routing, Link
State Routing, Hierarchical Routing, Broadcast Routing, Multicast
Routing, Anycast Routing: RIP, OSPF,BGP; Routing for Mobile Hosts. [5L]

Module 4: Transport Layer: [6L]
Process to Process delivery; UDP; TCP, SCTP, TCP

RENO, TCP/IP in Wireless environment, Congestion control in TCP: Congestion Control: Open Loop, Closed Loop choke packets; Quality of service: techniques to improve QoS: Leaky bucket algorithm, Token bucket algorithm. [5L]
Advanced topic such as Remote Procedure Call, Delay Tolerant Networks. [1L]

Module 5: Application Layer [3L]
Introduction to DNS, SMTP, SNMP, FTP, HTTP & WWW: Cryptography (Public, Private Key-based), Digital Signature, Firewalls

Module 6: Socket Programming [1L]
Introduction to Socket Programming, UDP socket, and TCP Socket

Textbooks:

1. B. A. Forouzan – ―Data Communications and Networking (3rd Ed.) ―– TMH
2. S. Tanenbaum – ―Computer Networks (4th Ed.) ‖ – Pearson Education/PHI
3. W. Stallings – ―Data and Computer Communications (5th Ed.)‖ – PHI/ Pearson Education
4. Zheng & Akhtar, Network for Computer Scientists & Engineers, OUP

Reference books:

1. Computer Networking -A top-down approach featuring the internet‖ – Pearson Education
2. Leon, Garica, Widjaja – Communication Networks‖ – TMH
3. Walrand – ―Communication Networks‖ – TMH.
4. Comer – ―Internetworking with TCP/IP, vol. 1, 2, 3(4th Ed) ‖ – Pearson Education/PHI

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	2	1	1	1	1	1	1	1	1	1	2	1	1
CO2	1	3	2	2	2	1	1	2	2	1	1	3	2	1
CO3	1	3	2	3	3	1	1	3	3	1	1	3	2	2
CO4	2	3	3	3	3	2	1	3	3	1	2	3	3	2
CO5	2	3	3	3	3	2	2	3	3	2	2	3	3	3
CO6	1	2	1	1	1	1	1	1	1	1	1	2	1	1

1= LOW, 3= HIGH

Course Name: Object Oriented Programming Language
Course Code: IT(DS)403
Contact: 3:0:0
Total Contact Hours: 36

Pre-requisite:
Partial Object-Oriented Programming using C++

Course Outcomes:
After attending the course, students should be able to

CO1	Design the process of interaction between Objects, classes & methods concerning Oriented Programming.
CO2	Acquire a basic knowledge of Object Orientation with different properties as well as different features of Java.
CO3	Analyze various activities of different string handling functions with various I/O
CO4	Discuss basic code reusability features with respect to inheritance, Package, and Interface.
CO5	Implement Exception handling, Multithreading, and Applet (Web program in java)

Course Content:
Module 1 [2L]
Introduction: Object Oriented Analysis (OOA) & Object Oriented Design (OOD) - Concepts of object-oriented programming language, Relationships among objects and classes-Generalization, Specialization, Aggregation, Association, Composition, links, Meta-class. [1L]; Object Oriented Programming concepts- Difference between Java and C++; Different features of Java [1L];
Module 2 [10L]
Java Basics: Basic concepts of java programming - Advantages of java, Byte-code & JVM, Data types, Different types of Variables.[1L] ;Java Operators & Control statements [1L]; Java loops. [1L]; Array [1L]; Creation of class, object, method. [1L]; Constructor- Definition, Usage of Constructor, Different types of Constructor [1L]; finalize method and garbage collection, Method & Constructor overloading. [1L]; this keyword, use of objects as parameter & methods returning objects [1L]; Call by value & call by reference. [1L]; Static variables & methods, Nested & inner classes [1L].
Module 3 [5L]
Basic String handling & I/O: Basic string handling concepts- Concept of mutable and immutable string, Methods of String class- charAt(), compareTo(), equals(),

equalsIgnoreCase(), indexOf(), length() , substring(). [1L]; toCharArray(), toLowerCase(), toString(), toUpperCase() , trim() , valueOf() methods, Methods of String buffer class- append(), capacity(), charAt(), delete(), deleteCharAt(). [1L];ensureCapacity(), getChars(), indexOf(), insert(), length(), setCharAt(), setLength(), substring(), toString(). [1L]; Command line arguments, basics of I/O operations – keyboard input using BufferedReader [1L] ; Scanner class in Java I/O operation [1L];

Module 4: [8L]

Inheritance and Java Packages: Inheritance - Definition, Advantages, Different types of inheritance, and their implementation. [1L], Super and final keywords, super () method. [1L]; Method overriding, Dynamic method dispatch. [1L]; Abstract classes & methods. [1L]; Interface - Definition, Use of Interface. [1L]; Multiple inheritance by using Interface. [1L]; Java Packages -Definition, Creation of packages. [1L]; Java Access Modifiers - public, private, default, and protected, Importing packages, member access for packages. [1L]

Module 5: [11L]

Exception handling - Basics, different types of exception classes. Difference between Checked & Unchecked Exceptions. [1L]; Try & catch related case studies. [1L]; Throw, throws & finally. [1L]; Creation of user-defined exception. [1L]; Multithreading - Basics, main thread [1L]; Thread life cycle.[1L]; Creation of multiple threads-yield(), suspend(), sleep(n), resume(), wait(), notify(), join(), isAlive().[1L]; Thread priorities, thread synchronization.[1L]; Interthread communication, deadlocks for threads[1L]; Applet Programming - Basics, applet life cycle, difference between application & applet programming[1L]; Parameter passing in applets. [1L]

Textbooks:

1. Herbert Schildt – "Java: The Complete Reference " – 9th Ed. – TMH.
2. E. Balagurusamy – " Programming With Java: A Primer " – 3rd Ed. – TMH.

Reference Books:

1. R.K Das – " Core Java for Beginners " – VIKAS PUBLISHING.
2. Rambaugh, James Michael, Blaha – " Object Oriented Modelling and Design " – Prentice Hall, India.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	1	1	1	2	1	1	2	1	1	2	1	2
CO2	3	2	2	2	2	2	2	1	2	2	1	3	2	2
CO3	3	2	2	3	2	2	1	2	1	2	2	3	2	2
CO4	2	3	3	3	3	3	2	2	2	2	2	3	3	2
CO5	3	3	3	3	3	3	3	3	3	3	3	3	3	3

3. 1= LOW, 3= HIGH

Course Name: Probability and Statistics
Course Code: M(DS)401
Contacts: 3:0:0
Total Contact Hours: 36
Credits: 4

Prerequisite:
The students to whom this course will be offered must have the concept of (10+2) standard algebra and calculus.

Course Outcome(s):
After completion of the course students will be able to
CO1: Recall the distinctive principles of probability and statistics.
CO2: Understand the theoretical workings of theory of probability and tests of hypotheses.
CO3: Apply statistical methods to compute and explain point estimators and interval estimators for mean, variance and proportion.
CO4: Analyze statistical data from engineering experiments.

Course Content

Module 1: (Probability and Random Variables) (15 Hours)
The axioms of probability, Conditional probability, Baye_s theorem, Discrete and continuous random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, Moments, Moment generating functions, Binomial, Poisson, Geometric, Uniform, Exponential and Normal distributions.

Module 2 (Two dimensional random variables) (6 Hours)
Joint distributions, Marginal and conditional distributions, Covariance, Correlation and linear regression, Transformation of random variables, Central limit theorem (for independent and identically distributed random variables).

Module 3 (Sampling Distribution) (4 Hours)
Distributions of the sample mean and the sample variance for a normal population, Chi-Square, t and F distributions, problems

Module 4 (Estimation) (11 Hours)
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, problems.

Module 5 (Testing of Hypotheses) (12 Hours)
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,

problems.

Project Domains:

1. Construction of Univariate and Bivariate Frequency Tables
2. Diagrammatic and Graphical representation of data.
3. Fitting of discrete and Continuous distributions
4. Regression Analysis
5. Curve Fitting
6. Tests of significance about Single Mean, Two Means
7. Construction of Confidence Intervals for Mean, Variance, and Proportion

Text and Reference Books:

1. Sheldon M. Ross, ―Introduction to Probability and Statistics for Engineers and Scientists‖, Academic Press, (2009).
2. D. C. Montgomery and G.C. Runger, ―Applied Statistics and Probability for Engineers‖, 5th edition, John Wiley & Sons, (2009).
3. Robert H. Shumway and David S. Stoffer, ―Time Series Analysis and Its Applications with R Examples‖, Third edition, Springer Texts in Statistics (2006).
4. N. G. Das: Statistical Methods, TMH.
5. Sancheti, D. S. &Kapoor, V.K.: Statistics Theory, Method & Application, Sultan chand & sons, New Delhi
6. N.K. Dutta (2004). Fundamentals of Biostatistics, Kanishka Publishers

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	1	1	_	_	_	_	_	_	_	_	_	1	1
CO2	3	2	1	_	_	_	_	_	_	_	_	_	2	2
CO3	3	2	2	_	_	_	_	_	_	_	_	_	1	1
CO4	3	3	2	3	_	_	_	_	_	_	_	_	2	2

1= LOW, 3= HIGH

Course Name: Principles of Management
Course Code: HU(DS)401
Contact (L: T: P): 2:0:0
Total Contact Hours: 24
Credit: 2

Prerequisites: None
Course Objectives:
The objective of the course is to make the students able to -

- To introduce the fundamental concepts and theories of management.
- To understand the functions of planning, organizing, leading, and controlling within an organization.
- To explore different management styles and their impact on organizational effectiveness.
- To grasp the importance of ethical decision-making and social responsibility in management.
- To develop basic managerial skills applicable in various organizational settings.

Course Outcomes (COs):
After completion of the course, students will be able to
CO1: Understand the foundational concepts and historical evolution of management thought.
CO2: Explain the process of planning and strategic decision-making in organizations.
CO3: Describe the principles of organizing, including organizational structure and design.
CO4: Analyze various leadership theories and their application in motivating employees.
CO5: Evaluate the controlling function of management and its role in achieving organizational goals.

Course Content:
Module – 1: Introduction to Management [5L]
Definition and Importance of Management, Functions of Management. [2L]
Evolution of Management Thought: Classical, Human Relations, Quantitative, Contingency Approaches. [2L]
Managerial Roles and Skills. [1L]

Module – 2: Planning and Decision Making [5L]

Nature and Purpose of Planning, Types of Plans, Strategic Planning Process. [2L]
Goal Setting, MBO (Management by Objectives). [1L]

Decision Making: Types of decisions, decision-making process, rational vs. bounded rationality. [2L]

Module – 3: Organizing [5L]

Nature and Purpose of Organizing, Organizational Structure (Line, Staff, Functional, Divisional, Matrix). [2L]
Departmentalization, Delegation of Authority, Centralization vs. Decentralization. [2L]
Span of Control, Formal and Informal Organization. [1L]

Module 4: Leading and Motivation [5L]

Leadership: Theories (Trait, Behavioral, Situational, Transformational), Leadership Styles. [2L]
Motivation: Theories (Maslow, Herzberg, Vroom, Equity), Application of Motivational Theories. [2L]
Communication: Process, types, barriers, and effective communication. [1L]

Module 5: Controlling and Emerging Issues [4L]

Controlling: Nature and Purpose, Control Process, Types of Control. [2L]
Techniques of Control (Budgetary Control, Performance Appraisals). [1L]
Ethical Issues in Management, Corporate Social Responsibility, Global Management. [1L]

Textbooks:

1. **"Management,"** 14th Edition, Stephen P. Robbins and Mary Coulter, Pearson, 2018.
2. **"Principles of Management,"** 10th Edition, Harold Koontz and Heinz Weihrich, McGraw-Hill Education 2012.

Reference Book:

1. **"Fundamentals of Management,"** 11th Edition, Ricky W. Griffin, Cengage Learning 2020.
2. **"Organizational Behavior,"** 18th Edition, Stephen P. Robbins and Timothy A. Judge, Pearson 2022.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3

CO1	1	1	1	_	_	_	_	_	_	_	_	1	1	1
CO2	2	2	1	_	_	_	_	_	_	_	_	1	2	2
CO3	2	2	2	_	_	_	_	_	_	_	_	2	1	1
CO4	3	3	2	2	2	2	_	_	_	_	_	2	2	2
CO5	3	3	2	2	2	2	_	_	_	_	_	2	2	2

1= LOW, 3= HIGH

Course Name: Database Management Systems Lab
Course Code: DS491
Contact (Periods/Week): = 3P/Week
Total Contact Hours: 36
Credits: 1.5

Pre-requisite:
1. Logic of programming language
2. Basic concepts of data structure and algorithms

Course Outcomes: After attending the course students should be able to

CO1 To Understand and Describe the basic concepts and utility of Database management system, different data models of Database management system.

CO2 To Design an Entity Relationship (E-R) Diagram and relational model for any kind of real-life application and able to Apply relational algebra operations, SQL, Neo4j for solving query.

CO3 To Analyze and Create the relational database for any real-life applications based on normalization.

CO4 To Apply the query optimization techniques, different file organization techniques and determine whether the transaction satisfies the ACID properties.

CO5 To Implement and organize the database of an organization as a team.

Course Content:

Module 1

Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.)

Module 2

Converting ER Model to Relational Model (Represent entities and relationships in Tabular form, represent attributes as columns, identifying keys) and apply the normalization techniques.

Module 3

Creation of Tables using SQL- Overview of using SQL tool, Data types in SQL, Creating Tables (along with Primary and Foreign keys), Altering Tables and Dropping Tables

Module 4

Practicing DML commands- Insert, Select, Update, Delete

Module 5

Practicing Queries using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT, CONSTRAINTS etc., Practicing Sub queries (Nested, Correlated) and Joins (Inner, Outer and Equi).

Module 6

Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING, VIEWS Creation and Dropping, Practicing on Triggers - creation of trigger, Insertion using trigger, Deletion using trigger, Updating using trigger.

Module 7

Procedures- Creation of Stored Procedures, Execution of Procedure, and Modification of Procedure, PL/SQL, Cursors- Declaring Cursor, Opening Cursor, Fetching the data, closing the cursor.

CO-PO Mapping

CO's	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	2	2	2	-	-	-	3	-	-	3	3	3

CO2	3	2	2	2	2	-	-	-	3	-	-	3	3	3
CO3	3	3	2	2	2	-	-	-	3	-	-	3	3	3
CO4	3	2	2	2	2	-	-	-	3	-	-	3	3	3
CO5	3	2	2	2	2	-	-	-	3	-	-	3	3	3
	3.0	2.2	2.0	2.0	2.0	-	-	-	3.0	-	-	3.0	3.0	3.0

1= LOW, 3= HIGH

Course Name: Computer Networks Lab
Course Code: DS492
Contact (Periods/Week): = 3P/Week
Total Contact Hours: 36
Credits: 1.5

Prerequisites:

- Familiarity and knowledge of Computer Network and Computer Architecture
- Also require strong knowledge of programming languages like C, Java and UNIX or Linux environment.

Course Outcomes (COs):

After attending the course students should be able to

CO1 To design and implement small size network and to understand various networking commands.

CO2 To provide the knowledge of various networking tools and their related concepts.

CO3 To understand various application layer protocols for its implementation in client/server environment

CO4 Understand the TCP/IP configuration for Windows and Linux

CO5 Learn the major software and hardware technologies used on computer networks

Course Contents:

- Familiarization of UNIX or Linux environment, UNIX or Linux general Commands specially Network Commands. Familiarization of Internetworking - Network Cables - Color coding - Crimping. Internetworking Operating Systems - Configurations. [6L]
- Socket Programming using TCP and UDP [18L] Implementing routing protocols such as RIP, OSPF. [2L]

- Familiarization of advanced simulators like Packet Tracer, NS2/NS3, OMNET++, TinyOS[4L] Server Configuration: only web server (If time permit, Instructor can do more than that) [6L]

Textbooks:

1. TCP sockets in C Programs-Practical guide for Programmers ByMicheal, J Donahoo and Kenneth L calvert. Socket Programming by Raj Kumar Buyaa.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	2	3	-	-	-	-	-	2	3	3	3
CO2	3	3	3	3	3	-	-	-	-	-	2	3	3	3
CO3	3	3	3	3	3	-	-	-	-	-	2	3	3	3
CO4	3	3	3	3	3	-	-	-	-	-	2	3	3	3
CO5	2	3	2	2	3	-	-	-	-	-	2	3	3	3

1= LOW, 3= HIGH

Course Name: Object Oriented Programming Lab
Course Code: DS493
Contact (Periods/Week): 3P/Week
Total Contact Hours: 36
Credits: 1.5

Pre-requisite:

1. Computer Fundamentals
2. Basic understanding of Computer Programming and related Programming Paradigms
3. Problem-Solving Techniques with proper logic Implementation.

Course Outcomes:

After attending the course, students should be able to

CO1 Create the procedure of communication between Objects, classes & methods.

CO2 Understand the elementary facts of Object Orientation with various characteristics as well as several aspects of Java.

CO3 Analyze distinct features of different string handling functions with various I/O operations.

CO4	Discuss simple Code Reusability notion w.r.t. Inheritance, Package and Interface.
CO5	Apply Exception handling, Multithreading and Applet (Web program in java) programming concept in Java.

Course Content:

Module 1: Java Basics

Simple Java programming using operators, control statements & loops, array.
Programming on class, object, and method, access specifier.
Programming on constructor, method/constructor overloading.
Programming on this keyword, call by value & call by reference, static variables & methods, inner classes.

Module 2: Basic String handling & I/O

Programming to show the use of String class methods - charAt() ,compareTo(),equals(), equalsIgnoreCase(), indexOf(), length() ,substring(),toCharArray(),toLowerCase(),toString(),toUpperCase(), trim(), valueOf() methods.
Programming to show the use of StringBuffer class methods - append(),capacity(),charAt(),delete(),deleteCharAt(),ensureCapacity(), getChars(), indexOf(), insert(), length(), setCharAt(), setLength(), substring(), toString() methods.
Programming on Command line arguments.
Programming using keyboard input by implementing BufferedReader& Scanner classes.

Module 3: Inheritance, Interface and Java Packages

Programming on Simple Inheritance, super and final keywords, super() method.

Programming on method overriding, dynamic method dispatch, abstract classes & methods, multiple inheritance by using interface.
Programming on importing system package, creating user-defined package, importing user- defined package, using protected access specifier, subclassing an imported class of a package, using same names for classes of different packages, adding multiple public classes to a package.

Module 4: Exception handling, Multithreading and Applet Programming

Programming on exception handling using try-catch block, implementing throw and throws keywords, using finally block, creating user-defined exception.
Programming on creating child threads i) by extending thread class ii) by implementing runnable interface, creating child threads by assigning thread priorities.
Programming on creating simple applet to display some message, creating applet two add 2 integers, creating applet to do GUI based programming.

Textbooks:

1. Herbert Schildt – **"Java: The Complete Reference "** – 9th Ed. – TMH
2. E. Balagurusamy – **" Programming With Java: A Primer "** – 3rd Ed. – TMH.

Reference Books:

1. R.K Das – **" Core Java for Beginners "** – VIKAS PUBLISHING.
2. Rambaugh, James Michael, Blaha – **" Object Oriented Modelling and Design "** – Prentice Hall, India

CO-PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	3	2	-	-	-	3	-	2	3	2	2
CO2	3	3	3	3	2	-	-	-	3	-	2	3	2	2
CO3	3	3	3	3	2	-	-	-	3	-	2	3	2	3
CO4	3	3	3	3	3	-	-	-	3	-	2	3	2	3
CO5	3	3	3	3	3	-	-	-	3	-	2	3	3	3
	3.0	3.0	2.8	3.0	2.4	-	-	-	3.0	-	2.0	3.0	2.2	2.6

1= LOW, 3= HIGH

Course Name: Numerical Methods Lab
Course Code: M(DS)491
Contact (Periods/Week): 3P/Week
Total Contact Hours: 36
Credits: 1.5

Prerequisite:
Any introductory course on a programming language (for example, C/ MATLAB).

Course Outcomes (COs):
After attending the course, students should be able to

CO1 Describe and explain the theoretical workings of numerical techniques with the help of C
CO2 Compute basic commands and scripts in a mathematical programming language
CO3 Apply the programming skills to solve the problems using multiple numerical approaches.
CO4 Analyze if the results are reasonable, and then interpret and communicate the
CO5 Apply the distinctive principles of numerical analysis and the associated error measures.

Course Content:

1. Assignments on Newton forward /backward, Lagrange's interpolation. Assignments on numerical integration using the Trapezoidal rule, Simpson's 1/3 rule, and Weddle's rule.
2. Assignments on numerical solution of a system of linear equations using Gauss elimination, Tridiagonal matrix algorithm, and Gauss-Seidel iterations. LU Factorization method.
3. Assignments on numerical solution of Algebraic equations by the Bisection method, the Regula-Falsi method, the Secant Method, and the Newton-Raphson method
4. Assignments on ordinary differential equations: Euler's method, Euler's modified method, Runge-Kutta methods, Taylor series method, and Predictor-Corrector method.

5. Implementation of numerical methods on computers through C/C++ and commercial Software Packages:MATLAB/Scilab/ LabVIEW/Mathematica/NAG(Numerical Algorithms Group)/Python.

CO-PO Mapping

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	2	-	-	3	-	2	2	2	2
CO2	3	3	2	2	3	2	-	-	3	-	2	2	2	2
CO3	3	3	2	2	3	2	-	-	3	-	2	2	2	2
CO4	3	3	2	2	3	2	-	-	3	-	2	2	2	2
CO5	3	3	2	2	3	2	-	-	3	-	2	2	2	2

1= LOW, 3= HIGH

Course Name: Soft Skill & Aptitude
Course Code: HU(DS)491
Contact (L: T: P): 2:0:0
Total Contact Hours: 24
Credit: 2

Prerequisites:

None

Course Objectives:

The objective of the course is to make the students able to -

- To enhance communication skills, both verbal and non-verbal, are essential for professional success.
- To develop critical thinking and problem-solving abilities.
- To improve interpersonal skills, teamwork, and leadership qualities.
- To build confidence and develop a positive attitude for career development.
- To prepare students for aptitude tests commonly encountered in recruitment processes.

Course Outcomes (COs):

After completion of the course, students will be able to

CO1: Communicate effectively in professional settings, both orally and in writing.
CO2: Apply critical thinking and problem-solving approaches to various situations.
CO3: Demonstrate effective teamwork and leadership skills in group activities.
CO4: Develop self-awareness, positive attitude, and professional etiquette.
CO5: Solve quantitative aptitude, logical reasoning, and verbal ability problems efficiently.

Course Content:

Module – 1: Communication Skills [6L]
Verbal Communication: Public speaking, presentation skills, active listening. [3L]
Non-Verbal Communication: Body language, gestures, facial expressions, proxemics. [2L]
Written Communication: Email etiquette, report writing basics, professional correspondence. [1L]

Module – 2: Interpersonal Skills and Teamwork [6L]
Building Rapport, Empathy, Conflict Resolution. [2L]
Teamwork: Role of a team player, collaboration, managing diversity in teams. [2L]
Negotiation Skills: Principles and strategies. [2L]

Module – 3: Critical Thinking and Problem Solving [6L]
Logical Reasoning: Deductive and inductive reasoning, analytical thinking. [3L]
Problem Solving: Steps in problem-solving, creative thinking, decision-making. [3L]

Module 4: Professional Etiquette and Personal Development [3L]
Time Management, Stress Management. [1L]
Professional Grooming, Interview Skills (introduction). [1L]
Goal Setting and Self-Motivation. [1L]

Module 5: Aptitude Training [3L]
Quantitative Aptitude: Number system, percentages, profit & loss, ratio & proportion. [1L]
Logical Reasoning: Blood relations, seating arrangements, coding-decoding. [1L]

Verbal Ability: Synonyms, antonyms, sentence correction, reading comprehension. [1L]

Textbooks:

1. **"Soft Skills: An Integrated Approach to Maximise Personality,"** 1st Edition, Gajendra Singh Chauhan and Et al., Vikas Publishing House, 2010.

2. **"A Modern Approach to Verbal & Non-Verbal Reasoning,"** 1st Edition, R.S. Aggarwal, S. Chand & Company Ltd., 2017.

Reference Book:

1. **“Quantitative Aptitude for Competitive Examinations,"** R.S. Aggarwal, S. Chand & Company Ltd., 2017.
2. **"The 7 Habits of Highly Effective People,"** Stephen Covey, Free Press, 1989.

CO-PO Mapping

CO’s	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	1	2	-	-	1	1	-	2	3	-	2	-	-
CO2	1	1	2	2	-	3	3	-	2	3	-	3	2	1
CO3	2	2	2	2	-	3	3	2	2	3	-	3	2	-
CO4	2	2	-	-	-	3	3	2	2	3	-	3	-	-
CO5	2	2	-	-	-	3	3	2	2	3	-	3	-	-

1= LOW, 3= HIGH

3rd Year 5th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credits**
A. THEORY					L	T	P	Total	
1	ENGG	Major	DS501	Advanced Artificial Intelligence	3	0	0	3	**3**
2	ENGG	Major	DS502	Formal Language and Automata Theory	3	0	0	3	**3**
3	ENGG	Major	DS503	Machine Learning	3	0	0	3	**3**
4	ENGG	Major	DS504A	Software Engineering	3	0	0	3	**3**
			DS504B	Optimization Techniques					
			DS504C	Blockchain Solutions for Crypto Security					
			DS504D	Big Data Analytics					
5	HUM	Minor	HU(DS)505	Research Methodology	3	0	0	3	**2**
B. PRACTICAL									
1	ENGG	Major	DS591	Advanced Artificial Intelligence Lab	0	0	3	3	**1.5**
2	ENGG	Major	DS592	Machine Learning Lab	0	0	3	3	**1.5**
3	ENGG	Major	DS595	Programming for Data Science using R Lab	0	0	3	3	**1.5**
4	Skill Enhancement Course	Internship	DS581	Internship/ Industrial Training	0	0	2	2	**2**
5	PRJ	Project	DS582	Project -I	0	0	0	4	**2**
		TOTAL CREDIT						**30**	**22.5**

Course Name: ADVANCED ARTIFICIAL INTELLIGENCE
Course Code: DS501
Contact (Periods/Week): = 3L/Week
Total Contact Hours: 36
Credits: 3

Pre-requisite: Data Structure, Design and Analysis of Algorithms, Statistics

Course Objectives: The objective of the course is to enable students to

- Comprehend the fundamental concepts of Knowledge Representation and Inferencing in Artificial Intelligence and its utilitarian importance in current technological context
- Formulate a problem as State-Space Exploration Framework or an Inferencing Framework of Artificial Intelligence.
- Use the strategies of AI-Heuristics to find acceptable solutions avoiding brute-force techniques.
- Design AI-Frameworks for Inferencing based on knowledge base.
- Analyze the effectiveness of an AI-Inferencing Modeling offering solutions to the respective problem

Course Outcome(s): After the completion of the course students will be able to

Course Outcomes	Name of Course Outcomes
CO1	To understand and explain the fundamental concepts of knowledge representation and inferencing in Artificial Intelligence and its utilitarian importance in current technological context for further exploration leading towards lifelong learning.
CO2	To Identify and formulate an engineering problem primarily to fit a State-Space Exploration Framework or an Inferencing Agent Design Framework within the scope of Artificial Intelligence paradigm.
CO3	To Explore relevant literature and apply the concept of Heuristic Techniques or Inferencing Models of Artificial Intelligence to solve problems.
CO4	To Develop Inferencing Models for proposing solutions to the problems of Artificial Intelligence.

CO5	To Implement Inferencing Models of Artificial Intelligence through developing feasible algorithms and investigate their effectiveness by analyzing their performances in solving the relevant problems.

CO-PO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	3	-	-	-
CO2	2	3	-	-	-	-	-	-	-	-	-	-	-	-	-
CO3	2	2	3	2	-	-	-	-	-	-	-	-	-	-	-
CO4	2	2	2	3	-	-	-	-	-	-	-	-	-	-	-
CO5	2	2	3	3	-	-	-	-	-	-	-	-	-	-	-
	2.20	2.20	1.60	1.60	-	-	-	-	-	-	-	0.60	-	-	-

Course Content:

Module-1: Introduction to Artificial Intelligence [2L]
Basic Concepts, History of Artificial Intelligence, Architecture of an Artificial Intelligent Agent, Applications of Artificial Intelligence.

Module-2: Artificial Intelligence Problem Formulation as State-Space Exploration Problem for Goal Searching [8L]
Basic Concepts, State-Space Exploration Formulation for Water Jug Problem, Missionaries and Cannibals Problems, Farmer-Wolf-Goat-Cabbage Problem, 8-Puzzle Problem, Constraint Satisfaction Problem and Production System for Goal Searching. Blind Search Techniques for Goal Searching: Breadth First Search, Depth First Search, Depth Limited Search, Iterative Deepening Search, Uniform Cost Search, Bi-directional Search.

Module-3: Heuristic Techniques for Goal Searching [8L]
Basic Concepts of Heuristic Techniques and Properties of Heuristic Functions, Hill Climbing Search. Best First Search, A* Search, AO* Search Genetic Algorithm Based Evolutionary Search, Ant Colony Optimization, Particle Swarm

Optimization.

Module-4: Adversarial Search for Game Playing [3L]
Basic Concepts, Minimax Search, Alpha-Beta Pruning.

Module-5: Knowledge Representation and Inference using Propositional Logic and Predicate Logic [5L]
Propositional Logic: Knowledge Representation and Inference using Propositional Logic Predicate Logic: Knowledge Representation, Inference and Answer Extraction using First Order Predicate Logic.

Module-6: Reasoning under Uncertainty [5L]
Bayesian Inferencing and Bayesian Belief Network, Dempster-Shafer Theory, Overview of Fuzzy Logic and Inferencing.

Module–7: Introduction to Natural Language Processing [2L]
Basic Concepts, Steps of Natural Language Processing, Morphological, Syntactic and Semantic Analysis, Discourse Integration and Pragmatic Analysis, Applications of Natural Language Processing.

Module-8: Introduction to Machine Learning [3L]
Basic concepts of Machine Learning Model, Supervised Learning, Unsupervised Learning, and Reinforced Learning, Overview of Artificial Neural Network.

Text books:
1. Russell, S. and Norvig, P. 2015. **Artificial Intelligence - A Modern Approach, 3rd edition**, Prentice Hall.
2. Rich, E., Knight, K and Shankar, B. 2009. **Artificial Intelligence, 3rd edition,** Tata McGrawHill.
Reference Books:
1. Padhy, N.P. 2009. **Artificial Intelligence and Intelligent Systems,** Oxford University Press.
2. Deepak Khemani, **"A First Course in Artificial Intelligence",** McGraw Hill.

Course Name: Formal Language and Automata Theory
Course Code: DS502
Contacts: 3:0:0
Total Contact Hours: 36
Credits: 3

Prerequisites:
1. Digital Logic
2. Computer organization
3. Computer Fundamentals

Course Outcomes (COs): After attending the course students should be able to

CO1	Understand the fundamental concepts of Finite State Automata to Explain or Illustrate and Identify problems where students can Apply the concept appropriately to Solve them.
CO2	Understand the fundamental concepts of Regular Expressions and its relation with DFA so that they can Develop regular expression for a specified language and Validate it.
CO3	Understand the fundamental concepts of Context Free Grammar so that they can Design gramma for a specified language and Validate it.
CO4	Explain or Illustrate the fundamental operating principles of Push Down Automata and Use it appropriately to Solve problems.
CO5	Understand the operating principles of Turing Machine and Design Turing Machines to Propose solutions to the related problems appropriately and validate the effectiveness as well as limitations of computation making the students aware of its utilitarian importance for further explorations leading towards life long learning.

CO-PO Mapping

CO-PSO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3								3
CO2	2	2	2	2								3
CO3	3	3	3	3								3
CO4	3	3	3	3								3
CO5	3	3	3	3					2	2		3

Cos	PSO1	PSO2	PSO3
CO1	3	3	3
CO2	3	3	3
CO3	3	3	3
CO4	3	3	3

Course Contents:

Module-1:[9L]
Fundamentals: Basic definition of sequential circuit, block diagram, mathematical representation, concept of transition table and transition diagram,[1L] Introduction to Finite State Model (FSM), Design of sequence detector, Finite State Machine, Finite Automata, Deterministic Finite Automation (DFA) and Non-deterministic Finite Automation (NFA), Transition diagrams, Transition tables and Language recognizers.[3L] NFA with empty transitions, Equivalence between NFA with and without empty transitions. NFA to DFA conversion.[2L], Minimization of FSM: Minimization Algorithm for DFA, Introduction to Myhill-Nerode Theorem [2L], Limitations of FSM, Application of Finite Automata[1L]

Module-2:[7L]
Finite Automata with output − Moore & Mealy machine. Representation of Moore & Mealy Machine, Processing of the String through Moore & Mealy Machine, Equivalence of Moore & Mealy Machine – Inter- conversion.[2L], Equivalent states and Distinguishable States, Equivalence and k equivalence, Minimization of, Mealy Machine[1L], Minimization of incompletely specified machine−Merger Graph, Merger Table, Compatibility Graph [2L], Lossless and Lossy Machine – Testing Table, Testing Graph [2L]

Module-3:[5L]
Regular Languages, Regular Sets, Regular Expressions, Algebraic Rules for Regular Expressions, Arden‘sTheorem statement and proof[1L]. Constructing Finite Automata (FA) for given regular expressions, Regular string accepted by FA[2L], Constructing Regular Expression for a given Finite Automata[1L], Pumping Lemma of Regular Sets. Closure properties of regular sets[1L]

Module-4:[9L]
Grammar Formalism-Context Free Grammars, Derivation trees, sentential forms. Rightmost and leftmost derivation of strings, Parse Tree, Ambiguity in context free grammars.[1L], Minimization of Context Free Grammars. [1L],Removal of

null and unit production[1L] Chomsky normal form and Greibach normal form.[1L], Pumping Lemma for Context Free Languages.[1L], Enumeration of properties of CFL, Closure property of CFL, Ogden‘s lemma & its applications [1L], Regular grammars−right linear and left linear grammars[1L], Pushdown Automata: Pushdown automata, definition. Introduction to DCFL, DPDA, NCFL, NPDA[1L] Acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence.[1L] Equivalence of CFL and PDA, inter-conversion.1L]

Module-5:[5L]
Turing Machine: Turing Machine, definition, model[1L] Design of TM, Computable functions [1L], Church‘s hypothesis, counter machine [1L] Types of Turing machines [1L]Universal Turing Machine, Halting problem[1L]

Textbook:
1. **Introduction to Automata Theory Languages and Computation**‖, Hopcroft.E. and Ullman J.D., Pearson Education.
Reference Books:
1**. Formal Languages and Automata Theory**‖**,**C. K. Nagpal, Oxford
2. **Switching and Finite Automata Theory**‖, Zvi Kohavi, 2nd Edition, Tata McGraw Hill

Course Name: Machine Learning
Course Code: DS503
Contact (Periods/Week): 3L/Week
Credit Point: 3 No. of Lectures: 36

Prerequisite:
1. Basic programming skills, Algorithm design.
2. Probability, Axioms of Probability, Conditional Probability, Bernoulli Distribution, Binomial Distribution, Multinomial Distribution, Uniform Distribution, Normal (Gaussian) Distribution, Chi Square Distribution, t Distribution, F Distribution. Probability Distribution and Density Functions, Joint Distribution and Density Functions, Conditional Distributions, Bayes' Rule, Expectation, Variance, Weak Law of Large Numbers.
3. Linear Algebra; Convex Optimization ; Statistics; Calculus.

Course Outcomes (COs): After attending the course students should be able to

CO-PO Mapping

COs	PO1	P O 2	P O 3	P O 4	P O 5	P O 6	P O 7	P O 8	P O 9	P O 10	P O 11	PO 12
CO1	3	3	3	3	3							3
CO2	3	3	3	3	3							3
CO3	3	3	3	3	3							3
CO4	3	3	3	3	3							3
CO5	3	3	3	3	3							3

CO1	Understand the fundamental concepts of basics of machine learning to Explain or Illustrate and Identify problems where students can Apply the concept appropriately to Solve them.
CO2	Understand the fundamental concepts of regression problems so that they can propose models for predicting values based on exemplary data and Analyze their performances.
CO3	Explain or Illustrate the fundamental strategies of unsupervised machine learning paradigm to solve clustering problems and Analyze their performances.
CO4	Explain or Illustrate the concepts of Mining Frequent Patterns, Associations and Data Streams and Apply them to solve the relevant problems and Analyse their performances.
CO5	Develop ideas to Propose solutions to the problems of supervised learning and Identify problem where students can Apply the concept appropriately and Analyze the effectiveness as well a limitations of solutions making the students aware of its utilitarian importance for further explorations leading towards lifelong learning.

CO-PSO Mapping

Cos	PSO1	PSO2	PSO3
CO1	3	3	3
CO2	3	3	3

CO3	3	3	3
CO4	3	3	3
CO5	3	3	3

Course Content

Module 1: [8L]

Supervised Learning (Regression/Classification) Basic methods: Distance-based methods, Nearest- Neighbours, Decision Trees, Naive Bayes Linear models: Linear Regression, Logistic Regression, Generalized Linear Models Support Vector Machines, Nonlinearity and Kernel Methods Beyond Binary Classification: Multi-class/Structured Outputs, Ranking

Module 2:[5L]

Unsupervised Learning Clustering: K-means/Kernel K-means
Dimensionality Reduction: PCA and kernel PCA Matrix Factorization and Matrix Completion Generative Models (mixture models and latent factor models)

Module 3:[4L]

Evaluating Machine Learning algorithms and Model Selection, Introduction to Statistical Learning Theory, Ensemble Methods (Boosting, Bagging, Random Forests)

Module 4: [7L]

Sparse Modeling and Estimation, Modeling Sequence/Time-Series Data, Deep Learning and Feature Representation Learning

Module 5: [7L]

Scalable Machine Learning (Online and Distributed Learning) A selection from some other advanced topics, e.g., Semi-supervised Learning, Active Learning, Reinforcement Learning, Inference in Graphical Models, Introduction to Bayesian Learning and Inference

Module6: [4L]

Recent trends in various learning techniques of machine learning and classification methods.

Text Books:

1. Kevin Murphy, **Machine Learning: A Probabilistic Perspective**, MIT Press, 2012

2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, **The Elements of Statistical Learning, Springer**

Reference Books:

1. Christopher Bishop, **Pattern Recognition and Machine Learning, Springer**, 2007
2. Dr. Rajiv Chopra, Machine Learning, Khanna Publishing House, 2018

Course Name: Software Engineering
Course Code: DS504A
Contact: 3:0:0
Total Contact Hours: 36
Credits: 3

Prerequisites: Programming for Problem Solving

Course Objectives:
The objective of the course is to make the students able to -
1. Gain knowledge of fundamental concepts, models, and methodologies used in software engineering.
2. Design, develop, test, and maintain software systems using structured and agile approaches.
3. Develop proficiency in software requirement analysis, architectural design, and use of modeling tools.
4. Apply testing, debugging, and validation techniques to ensure software reliability and performance.
5. Cultivate collaboration, communication, and project management skills for successful software
development in team environments.
Course Outcomes (COs): After attending the course students should be able to

CO1	Understand the basic concept of Software Engineering and mathematical knowledge and apply them in designing solution to engineering problem including the specification, design, implementation, and testing of software systems that meet specification, performance, maintenance and quality requirements
CO2	Analyze, elicit and specify software requirements through a productive working relationship with various stakeholders of the project
CO3	Design applicable solutions in one or more application domains using software engineering approaches that integrates ethical, social, legal and economic concerns.
CO4	Develop the code from the design and effectively apply relevant standards and perform testing, and quality management and practice team work.
CO5	Identify and Use modern engineering tools necessary for software project management time management and software reuse, and an ability to engage in life-long learning.

CO-PO Mapping

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	3	3	2								3
CO2	3	3	3	2								3
CO3	3	3	3	2								3
CO4	3	3	3	2								3
CO5	3	3	3	2								3

CO-PSO Mapping

Cos	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	3
CO2	3	3	3
CO3	3	3	3
CO4	3	3	3
CO5	3	3	3

Course Content:

Module-1:[6L]
Introduction: Software Engineering, Characteristics, Components, Application, Definitions. Software Project Planning-Feasibility Analysis, Technical Feasibility, Cost-Benefit Analysis, Basics of estimation : COCOMO(Basic, intermediate, Complete) model.

Module- 2: [6L]
Evolution and impact of Software engineering, software life cycle models: Waterfall, prototyping, Evolutionary, and Spiral models. Feasibility study, Functional and Non-functional requirements, Requirements gathering, Requirements analysis and specification.

Module -3:[8L]
Basic issues in software design, modularity, cohesion, coupling and layering, function-oriented software design: DFD and Structure chart, object modeling using UML, Object-oriented software development, user interface design. Coding standards and Code review techniques.

Module -4:[7L]
Fundamentals of testing, White-box, and black-box testing, Test coverage analysis and test case design techniques, mutation testing, Static and dynamic analysis, Software reliability metrics, reliability growth modeling

Module -5: [9L]
Software project management, Project planning and control, cost estimation, project scheduling using PERT and GANTT charts, cost-time relations: Rayleigh-Norden results, quality management, ISO and SEI CMMI, PSP and Six Sigma. Computer aided software engineering, software maintenance, software reuse, Component- based software development.

Text Books:

1. **Fundamentals of Software Engineering** by Rajib Mall, –PHI-3rd Edition, 2009.
2. **Software Engineering**- Pankaj Jalote(Wiley-India)

Reference Books:

1. **Software Engineering** – AgarwalandAgarwal(PHI)
2. **Software Engineering,** by Ian Sommerville, Pearson Education Inc., New Delhi, (2009).

3. **Software Engineering: A Practitioner‟s Approach‖**, by Roger S. Pressman, McGraw-Hill.(2005)

Course Name: Optimization Techniques
Course Code: D504B
Contact: 3:0:0
Total Contact Hours: 36
Credits: 3

Prerequisites:
1. Engineering Mathematics
2. Programming Fundamentals
3. Data Structures and Algorithms
4. Data Handling and Visualization

Course Objectives:
The objective of the course is to make the students able to -
1. Introduce mathematical foundations of optimization.
2. Develop skills in formulating and solving optimization problems in Data Science.
3. Familiarize with linear, nonlinear, and discrete optimization techniques.
4. Apply optimization algorithms to real-world computational problems.

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

CO1	**Explain** fundamental concepts, classifications, and applications of optimization in Computer Science and Data Science.
CO2	**Formulate** real-world problems as linear, nonlinear, or discrete optimization models.
CO3	**Apply** classical optimization methods such as Simplex, gradient-based techniques, and dynamic programming.
CO4	**Implement** and analyze meta-heuristic algorithms for complex optimization problems.
CO5	**Evaluate** and select suitable optimization techniques for solving multi-objective and large-scale data-driven problems.

Course Content:
Module – 1:[6L]
Introduction to Optimization: Basic concepts- optimality, constraints, objective

functions, Applications in Computer Science & Data Science, Classification: linear vs. nonlinear, unconstrained vs. constrained, deterministic vs. stochastic.

Module – 2:[6L]

Linear Programming (LP): Formulation of LP problems, Graphical method, Simplex method, Duality and sensitivity analysis, Applications of LP in data-driven decision-making.

Module – 3:[8L]

Nonlinear Optimization: Unconstrained optimization: gradient descent, Newton's method, Constrained optimization: Lagrange multipliers, Karush-Kuhn-Tucker (KKT) conditions, Quadratic programming.

Module 4:[8L]

Discrete & Combinatorial Optimization: Integer programming and branch-and-bound, Dynamic programming, Greedy algorithms, Applications in machine learning model selection, scheduling, and routing.

Module 5:[8L]

Meta-heuristic & Advanced Techniques: Genetic algorithms, Particle Swarm Optimization, Ant Colony Optimization, Convex optimization in machine learning, Multi-objective optimization.

Text Book:

1. S. S. Rao – *Engineering Optimization: Theory and Practice*
2. Kalyanmoy Deb – *Optimization for Engineering Design*

Reference Book:

1. Dimitris Bertsimas & John Tsitsiklis – *Introduction to Linear Optimization*

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	1	1	1	1	-	1	-	2	3	2	2
CO2	3	3	2	2	2	1	1	-	1	1	2	3	3	2
CO3	3	3	2	3	3	1	1	-	1	1	2	3	3	2
CO4	2	2	2	3	3	1	1	-	1	1	2	3	3	3
CO5	2	3	2	2	3	2	2	1	2	1	3	3	3	3

1= LOW, 3= HIGH

Course Name: Blockchain and Cryptocurrency Technologies
Course Code: D504C
Contact: 3:0:0
Total Contact Hours: 36
Credits: 3

Prerequisites:
1. Basics of Cryptography
2. Data Structures & Algorithms
3. Fundamentals of Computer Networks

Course Objectives:
The objective of the course is to make the students able to -
1. Introduce the principles, architecture, and operation of blockchain technology.
2. Explain cryptographic techniques and consensus mechanisms used in blockchain security.
3. Analyze security vulnerabilities and threat models in blockchain and cryptocurrency systems.
4. Study real-world blockchain applications, case studies, and compliance frameworks.
5. Equip students with skills for designing secure blockchain-based solutions in data-driven
environments.

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

CO1	**Explain** blockchain architecture and underlying cryptographic foundations
CO2	**Evaluate** consensus algorithms and their security trade-offs.
CO3	**Identify** and mitigate blockchain and cryptocurrency security vulnerabilities.
CO4	**Apply** blockchain for secure data sharing, privacy preservation, and decentralized analytics.
CO5	**Design** blockchain-based security models for real-world data science applications.

Course Content:
Module – 1:[6L]
Introduction to Blockchain Technology: History and evolution of blockchain, Centralized vs. decentralized systems, Blockchain architecture: blocks, nodes, ledgers, Public, private, and consortium blockchains, Smart contracts basics.

Module – 2:[6L]
Cryptography for Blockchain Security: Hash functions (SHA-256, Keccak), Public key cryptography (RSA, ECC), Digital signatures & PKI in blockchain, Merkle trees & Patricia tries, Wallets and key management.

Module – 3:[6L]
Consensus Mechanisms & Security Models: Proof of Work (PoW), Proof of Stake (PoS) & its variants (DPoS, NPoS), Byzantine Fault Tolerance (PBFT, Tendermint), Blockchain fork attacks and mitigation strategies.

Module 4:[6L]
Blockchain Vulnerabilities & Attack Vectors: Double spending attacks, 51% attacks, Smart contract vulnerabilities (Reentrancy, Integer overflow), Sybil attacks, Eclipse attacks, Privacy leakage in public blockchains.

Module 5:[12L]
Cryptocurrency Security: Cryptocurrency protocols (Bitcoin, Ethereum, Monero), Cold wallets vs. hot wallets, Exchange security & hacks, Stablecoins and DeFi security risks, Regulatory compliance (AML, KYC); Blockchain for Data Science Security: Secure data sharing via blockchain, Decentralized identity management, Blockchain-based federated learning, Integration with IoT, AI, and Big Data systems, Blockchain in healthcare, supply chain, and finance security.

Text Book:
1. Narayanan, Arvind et al., *Bitcoin and Cryptocurrency Technologies*, Princeton University Press.
2. Antonopoulos, Andreas M., *Mastering Bitcoin*, O'Reilly Media.
3. Bashir, Imran, *Mastering Blockchain*, Packt Publishing.

Reference Books & Resources:
1. Nakamoto, Satoshi, *Bitcoin: A Peer-to-Peer Electronic Cash System* (Whitepaper).
2. Ethereum Foundation Docs (https://ethereum.org/en/developers/).
3. Hyperledger Fabric Documentation.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	1	1	1	3	2	2	3	3	3
CO	3	3	3	3	3	1	1	1	3	2	2	3	3	3

2														
CO 3	3	3	3	3	3	1	1	1	3	2	2	3	3	3
CO 4	3	3	3	3	3	1	1	1	3	2	2	3	3	3
CO 5	3	3	3	3	3	1	1	1	3	2	2	3	3	3

1= LOW, 3= HIGH

Course Name: BIG DATA ANALYSIS
Course Code: DS504D
Contact (Periods/Week):=3L/Week

Total Contact Hours: 36
Credits: 3

Pre-requisite:

1. Data Structure, Design and Analysis of Algorithms, Database Management Systems, Statistics, Artificial Intelligence, Programming skills of Python.

Course Outcomes: After attending the course students should be able to

CO-PO-PSO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
DS504C. 1	3	2	2	1	-	-	-	-	-	-	-	3	3	2	2
DS504C. 2	3	3	3	2	-	-	-	-	-	-	-	3	3	3	2
DS504C. 3	3	3	2	3	-	-	-	-	-	-	-	3	3	2	3
DS504C. 4	2	3	2	2	-	-	-	-	-	-	-	3	2	2	3
DS504C. 5	3	2	2	3	-	-	-	-	-	-	-	3	2	3	3
	2.8	2.6	2.2	2.2	-	-	-	-	-	-	-	3.0	2.6	2.8	2.6

Course Outcomes	Name of Course Outcomes
DS504C.1	Understand and explain the fundamental concepts of the Big Data Analytics which are primarily explored for making automated decisions using machine learning strategies on analyzing large scale structured as well as unstructured data distributed across multiple locations (Map Reduce, Hadoop and No SQL Framework) underscoring the utilitarian importance in current technological context for further exploration leading towards lifelong learning.
DS504C.2	Identify and formulate an engineering problem of analyzing large scale data distributed across multiple locations to make automated meaningful decisions within the scope of Big Data Analytics Frameworks.
DS504C.3	Explore relevant literature and apply the concepts of Big Data Analytics to solve problems of making automated decisions dealing with large scale structured as well as unstructured data using Map Reduce, Hadoop and advanced SQL Frameworks.
DS504C.4	Excogitate ideas for proposing solutions to the challenging problems of Big Data Analytics.
DS504C.5	Implement ideas of Big Data Analytics through developing feasible algorithms or frameworks and investigate their effectiveness in solving the relevant problems by analyzing the performances using proper techniques.

Course Content:

Module – 1: Introduction to Basic Analytics [10L]:
Introduction: Big data overview, Analyst‘s perspective on data repositories, Current analytical architecture, Drivers of big data, Examples of big data analytics. Life Cycle of Data Analytics: Phase 1: Discovery, Phase 2: Data preparation, Phase 3: Model planning, Phase 4: Model building, Phase 5: Communication of results, Phase 6: Making operational. Basic Analytic Methods: Visualization, Dirty data, Data exploration versus presentation, Statistical methods for evaluation – hypothesis testing, difference of means, rank sum test, type I and type II errors, ANOVA.

Module - 2: Advanced Analytic Methods I [8L]:
Clustering: Overview, K-means, Determining the number of clusters, Diagnostics. Association Rules: Overview, Apriori algorithm, Evaluation of candidate rules, Application of association rules, Validation and testing, Diagnostics. Regression: Linear regression - model description, Logistic regression – model description, Other regression models. Classification: Decision trees – overview, General algorithm, Decision tree algorithms, Evaluating a decision tree, Naïve Bayes – Bayes theorem, Naïve Bayes classifier, Diagnostics of classifiers.

Module – 3: Advanced Analytic Methods II [8L]:
Time Series Analysis: Overview, Box-Jenkins methodology, Autocorrelation function (ACF), Autoregressive model, Moving average model, ARMA and ARIMA model, Building and evaluating an ARIMA model. Text Analysis: Steps in text analysis, Collecting raw text, Representing text, Term Frequency-Inverse Document Frequency (TFIDF), Categorizing documents by types, Determining sentiments. Map Reduce and Hadoop: Analytics for unstructured data – map reduce, Apache Hadoop, Hadoop Ecosystem – Pig, Hive, Hbase, Mahout.

Module – 4: Advanced Analytic Methods III [10L]:
Technology and Tools: SQL essentials - Join, Set, Grouping extensions, Advanced SQL – Window functions, User-defined functions, Ordered aggregates, MADlib, NoSQL. Integration of Techniques: Communicating and operationalizing an analytic project. Creating final deliverables – Developing core materials, project goals, Main findings, Approach, Model description and model details, Recommendations, Providing technical specifications and code. Data visualization basics - Key points, evolution of a graph, common representation methods, how to clean up a graphic.

Text book:
1. EMC Education Services (Editor), Data Science and Big Data Analytics. John

Wiley & Sons, 2015. 2. Mike Barlow, Real-Time Big Data Analytics: Emerging Architecture. O'Reilly, 2013.

Reference Books:

1. Nathan Marz and James Warren, Big Data: Principles and Best Practices for Scalable Real-time Data Systems. Manning Publications, 2015.

2. Venkat Ankam, Big Data Analytics. Packt Publishing Ltd., UK, 2016

Course Name: Research Methodology
Course Code: HU(DS)505
Contact (L: T: P): 3:0:0
Total Contact Hours: 36
Credit: 3

Prerequisites:

1. Engineering Mathematics
2. Introduction to Programming and Problem Solving
3. Data Structures and Algorithms
4. Database Management Systems
5. Data Handling and Visualization

Course Objectives:

The objective of the course is to make the students able to -

1. Understand research concepts, processes, and ethics
2. Equip with skills for problem formulation, literature survey, and hypothesis development
3. Be trained in experimental design, data analysis, and interpretation
4. Be familiarized with technical writing and research dissemination in Data Science.

Course Outcomes (COs):

After completion of the course students will be able to

CO1	**Explain** fundamental concepts, types, and processes of research, with a focus on applications in Computer Science and Data Science.
CO2	**Conduct** systematic literature reviews using scholarly databases and identify research gaps.
CO3	**Formulate** clear research problems, hypotheses, and select appropriate research methodologies.
CO4	**Apply** statistical and computational tools for data collection, preprocessing, and analysis.

CO5	**Prepare** structured research reports and technical papers, following ethical guidelines and proper citation standards.

Course Content:

Module – 1:[6L]

Introduction to Research: Meaning, objectives, and types of research (Basic, Applied, Exploratory, Descriptive, Analytical), Research process and problem identification, Research in Computer Science & Data Science contexts.

Module – 2:[6L]

Literature Review & Problem Formulation: Sources of information: Journals, conference proceedings, patents, technical reports, Tools for literature search (Google Scholar, IEEE Xplore, Scopus, Web of Science), Research gap identification and problem statement formulation.

Module – 3:[8L]

Research Design & Methods: Qualitative and quantitative research approaches, Experimental design and case studies in Data Science, Sampling techniques, surveys, and observational studies, Hypothesis formulation and testing.

Module 4:[10L]

Data Collection & Analysis: Data types, sources, and collection methods, Preprocessing, cleaning, and validation, Statistical tools for analysis (t-test, ANOVA, correlation, regression), Use of software tools (R, Python, SPSS, MATLAB).

Module 5:[6L]

Technical Writing & Research Ethics: Structure of research reports, theses, and technical papers, Referencing styles (APA, IEEE), Plagiarism and ethical issues in research, Intellectual Property Rights (IPR) and patents.

Text Book:

1. C.R. Kothari & Gaurav Garg – *Research Methodology: Methods and Techniques*.
2. Ranjit Kumar – *Research Methodology: A Step-by-Step Guide for Beginners*

Reference Book:

1. Wayne C. Booth et al. – *The Craft of Research*.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	1	1	1	1	-	1	-	2	2	1	2
CO2	2	3	1	2	2	1	1	-	1	-	2	2	1	3

CO 3	2	3	2	2	2	1	1	-	1	1	2	2	2	3
CO 4	2	2	1	3	3	1	1	-	1	1	2	2	3	2
CO 5	1	2	1	1	1	1	3	1	3	1	2	1	2	3

1= LOW, 3= HIGH

Course Name: Advanced Artificial Intelligence Lab
Course Code:DS591
Contact (L: T: P): 0:0:3
Total Contact Hours: 36
Credit: 1.5

Prerequisites:
1. Artificial Intelligence
2. Machine Learning
3. Data Structures and Algorithms
4. Python Programming
5. Probability & Statistics

Course Objectives:
The objective of the course is to make the students able to -
1. Acquire practical exposure to advanced AI techniques beyond the fundamentals
2. Implement AI algorithms for real-world problem-solving in Data Science domains
3. develop proficiency in using AI frameworks and tools for research and applications

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

CO1	**Implement** advanced AI search, planning, and reasoning techniques for complex problem-solving
CO2	**Apply** machine learning and deep learning methods for intelligent decision-making
CO3	**Utilize** AI frameworks (e.g., TensorFlow, PyTorch, scikit-learn) for practical applications in Data Science
CO4	**Design** and execute AI experiments for real-world datasets
CO5	**Analyze** ethical and societal impacts of AI solutions in diverse domains

Course Content:

Module – 1:[6L]

Advanced Search and Planning : Implementation of informed search algorithms (A*, IDA*, heuristic search), Game playing with adversarial search (minimax, alpha-beta pruning), AI-based path planning problems.

Module – 2:[6L]

Knowledge Representation & Reasoning: Logic-based reasoning with Prolog/Python, Ontology creation and semantic reasoning, Constraint satisfaction problems.

Module – 3:[8L]

Advanced Machine Learning in AI: Ensemble methods (Random Forest, XGBoost), Dimensionality reduction (PCA, t-SNE, LDA), AI-based feature engineering.

Module 4:[8L]

Deep Learning for AI Applications: Convolutional Neural Networks (CNNs) for image tasks, Recurrent Neural Networks (RNNs) and LSTMs for sequence modeling, Transformer-based models for NLP.

Module 5:[8L]

AI for Complex Systems & Ethics: Multi-agent systems simulation, AI in decision-making and optimization, Case study on ethical AI applications.

Text Book:

1. Stuart Russell & Peter Norvig – *Artificial Intelligence: A Modern Approach* (3rd or 4th Edition),
Pearson.
2. Ian Goodfellow, Yoshua Bengio & Aaron Courville – *Deep Learning*, MIT Press.
3. Christopher M. Bishop – *Pattern Recognition and Machine Learning*, Springer.
4. Ethem Alpaydin – *Introduction to Machine Learning* (3rd Edition), MIT Press.

Reference Book:

1. Richard S. Sutton & Andrew G. Barto – *Reinforcement Learning: An Introduction* (2nd Edition), MIT Press.
2. Aurélien Géron – *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd or 3rd Edition), O'Reilly.
3. Steven Bird, Ewan Klein & Edward Loper – Natural Language Processing with Python, O'Reilly.
4. Francois Chollet – Deep Learning with Python, Manning Publications.
5. David L. Poole & Alan K. Mackworth – Artificial Intelligence: Foundations of

Computational Agents (2nd Edition), Cambridge University Press.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	2	1	1	1	1	1	2	3	3	2
CO2	3	3	3	2	3	1	1	1	1	1	3	3	3	3
CO3	3	2	2	2	3	1	1	1	1	1	3	3	3	2
CO4	2	3	3	3	3	1	1	2	2	2	3	3	3	3
CO5	1	2	2	1	1	3	3	2	3	2	2	2	2	3

1= LOW, 3= HIGH

Course Name: Machine Learning Lab
Course Code: DS592
Contact: 0:0:3
Total Contact Hours: 36
Credits: 1.5

Prerequisite: Familiarity with JAVA/ Python Programming

Course Outcomes (COs): After attending the course students should be able to

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3	3							3
CO2	3	3	3	3	3							3
CO3	3	3	3	3	3							3
CO4	3	3	3	3	3							3
CO5	3	3	3	3	3							3

CO1	Understand the fundamental concepts of basics of machine learning to Explain or Illustrate and Identify problems where students can Apply the concept appropriately to Solve them.
CO2	Understand the fundamental concepts of regression problems so that they can propose models for predicting values based on exemplary data and Analyze their performances.
CO3	Explain or Illustrate the fundamental strategies of unsupervised machine learning paradigm to solve clustering problems and Analyze their performances.
CO4	Explain or Illustrate the concepts of Mining Frequent Patterns, Associations and Data Streams and Apply them to solve the relevant problems and Analyse their performances.
CO5	Develop ideas to Propose solutions to the problems of supervised learning and Identify problems where students can Apply the concept appropriately and Analyze the effectiveness as well as limitations of solutions making the students aware of its utilitarian importance for further explorations leading towards lifelong learning.

CO-PO Mapping

CO-PSO Mapping

Cos	PSO1	PSO2	PSO3
CO1	3	3	3
CO2	3	3	3
CO3	3	3	3
CO4	3	3	3
CO5	3	3	3

List of Lab Experiments:

1. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
2. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate- Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.
3. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
4. Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.
5. Write a program to implement the naÔve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6. Assuming a set of documents that need to be classified, use the naÔve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.
7. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library classes/API.
8. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.
9. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.
10. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

Course Name: Programing for Data Science Using R Lab
Course Code: DS595
Contact Hours (per week): 3
Credit: 1.5
Allotted Hours: 36L

Pre-requisites:
Data Structure, Design and Analysis of Algorithms, Statistics, Artificial Intelligence, Concept of any Programming Language.

Course Objectives:
1. Identify the appropriate patterns in the data.
2. Obtain, clean/process, and transform data.
3. Analyze the data using several statistical techniques.

4. Construction of data-driven decision models.
5. Use business data to evaluate various marketing strategies. 6. Data analytics to make predictions based on available data.

Course Outcomes:
After completion of the course students will be able to
CO1: Identify the appropriate patterns in the data.
CO2: Summarize the information content in a large data set.
CO3: Extract knowledge from the data sets.
CO4: Construct data-driven decision models.
CO5: Use historical data to predict future outcomes.

COURSE CONTENT
WEEK-1: Write an R program to find Principal Components using KL-Transform of a given data set.
WEEK-2: Write an R program to find discriminatory components using Multiple Discriminant Analysis for a given data set.

WEEK -3: Regression Implementation and Analysis of Linear and Nonlinear Regression using R programming.
WEEK -4: Classification 1 Implementation and Analysis of k-Nearest-Neighbor Classifier
WEEK -5: Classification 2 Implementation and analysis of Decision Tree Classifier, Naïve Bayes Classifier using R programming.

WEEK -6: Classification 3 Implementation and analysis of ANN-Backpropagation using R programming.
WEEK -7: Classification 4 Implementation and analysis of Linear SVM Based Classifier using R programming.
WEEK-8: Classification Implementation and analysis of Linear SVM Based Classifier using R programming.
WEEK -9: Write an R program to find solution of an over determined system of equations using Least Square Method. WEEK -10: Write an R program to construct a ensembled decision tree or Random Forest.
WEEK -11: Clustering Implementation and Analysis of k-Means and k-Medoids using R programs.
WEEK -12: Write an R program for time series forecasting using ARIMA method.

Textbook:
1. **'R for Data Science'**, Hadley Wickham, 1st edition, O′Reilly
2. **'The Book of R'**, Tilman M. Davies, 1st edition, No Strach Press
Reference Books:
1. Discovering Statistics Using R, Andy Field, 1st edition, SAGE Publications Ltd.
2. The Art of R Programming, Norman Matloff, 1st edition, No Starch Press.

Course Name: Internship/Industrial Training
Course Code: DS581
Contact Hours (per week): 3
Credit: 1.5
Allotted Hours: 36L

Students will do Internship and Industrial training.

3rd Year 6th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credits**
A. THEORY					L	T	P	Total	
1	ENGG	Major	DS601	Neural Networks and Deep Learning	3	1	0	4	**4**
2	ENGG	Major	DS602	Data Warehousing & Data Mining	3	0	0	3	**3**
3	ENGG	Major	DS603	Natural Language Processing	3	0	0	3	**3**
4	ENGG	Minor	DS604A	Compiler Design	3	0	0	3	**3**
			DS604B	Data Science Use Cases					
			DS604C	Mobile Computing					
5	ENGG	Major	DS605	Generative AI & Explainable AI	3	0	0	3	3
B. PRACTICAL									
1	ENGG	Major	DS691	Neural Networks and Deep Learning Lab	0	0	3	3	**1.5**
2	ENGG	Major	DS693	Natural Language Processing Lab	0	0	3	3	**1.5**
3	ENGG	Major	DS695	Generative AI & Explainable AI Lab	0	0	3	3	**1.5**
4	PRJ	Project	DS681	Project -II	0	0	12	12	**6**
		TOTAL CREDIT		37					**26.5**

***Note:** Course from Skill India to be done in consultation with the department. Certificate of completion is to be submitted to the department/COE Section by the end of the 5th semester as the mark of completion of MC681.

Course Name: Neural Networks and Deep Learning

Course Code: DS601

Contact (L: T: P): 3:1:0

Total Contact Hours: 52

Credit: 4

Prerequisites:

This course assumes that you have mature programming abilities. You will not be taught how to make algorithms or write code. You should know that coming in. If you're unsure about your programming abilities, you should wait and take this class later

Course Objectives:

The objective of the course is to make the students able to -

Introduce major deep learning algorithms, the problem settings, and their applications to solve real world problems.

The course covers feedforward networks, convolutional networks, recurrent and recursive networks, as well as general topics such as input encoding and training techniques.

Course Outcomes (COs):

After completion of the course students will be able to

CO1	**Understand** the learning components of neural networks and apply standard neural network models to learning problems**.**
CO2	**Analyse** the learning strategies of deep learning – regularization, generalization, optimization, bias and variance.
CO3	**Analyze** regular deep learning models for training, testing and validation in standard datasets.
CO4	**Apply** neural networks for deep learning using standard tools.
CO5	**Understand** the mathematics for Deep learning.

Course Content:

Module – 1:

Introduction: Various paradigms of earning problems, Perspectives and Issues in deep learning framework, review of fundamental learning techniques**.**

Module – 2:

Feedforward neural network: Artificial Neural Network, activation function, multi-layer neural network.

Module – 3:

Training Neural Network: Risk minimization, loss function, backpropagation, regularization, model selection, and optimization.

Module 4:

Conditional Random Fields: Linear chain, partition function, Markov network, Belief propagation, Training CRFs, Hidden Markov Model, Entropy.

Module 5:

Deep Learning: Deep Feed Forward network, regularizations, training deep models, dropouts, Convolutional Neural Network, Recurrent Neural Network, Deep Belief Network.

Module 6:

Probabilistic Neural Network: Hopfield Net, Boltzman machine, RBMs, Sigmoid net, Autoencoders.

Module 7:

Deep Learning research: Object recognition, sparse coding, computer vision, natural language processing.

Module 8:

Deep Learning Tools: Caffe, Theano, Torch.

Text Book:

1. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016..

2. Bishop, C. ,M., Pattern Recognition and Machine Learning, Springer, 2006.

Reference Book:

1. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
2. Golub, G.,H., and Van Loan,C.,F., Matrix Computations, JHU Press,2013.
3. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	3	0	1	1	1	3	2	2	3	2	2
CO2	3	2	3	2	2	1	1	1	2	2	2	3	2	2
CO3	3	2	3	2	3	1	1	1	2	2	2	3	2	2
CO4	3	1	2	1	2	1	1	1	2	2	2	3	2	1
CO5	3	1	2	1	0	1	1	1	3	2	2	3	2	2

Course Name: Data Warehousing & Data Mining

Course Code: DS602

Contact (L: T: P): 3:0:0

Total Contact Hours: 40

Credit: 3

Prerequisites:

This course assumes that you have mature programming abilities. You will not be taught how to make algorithms or write code. You should know that coming in. If you're unsure about your programming abilities, you should wait and take this class later

Course Objectives:

The objective of the course is to make the students able to -

Introduce major deep learning algorithms, the problem settings, and their applications to solve real world problems.

The course covers feedforward networks, convolutional networks, recurrent and recursive networks, as well as general topics such as input encoding and training techniques.

Course Outcomes (COs):

After completion of the course students will be able to

CO1	**Study of different sequential pattern algorithms**
CO2	**Study the technique to extract patterns from time series data and it application in real world.**
CO3	**Can extend the Graph mining algorithms to Web mining**
CO4	**Help in identifying the computing framework for Big Data**
CO5	**Understand** Data mining algorithms

Course Content:

Unit 1: Introduction to Data Warehousing; Data Mining: Mining frequent patterns, association and correlations; Sequential Pattern Mining concepts, primitives, scalable methods;

Unit 2: Classification and prediction; Cluster Analysis – Types of Data in Cluster Analysis, Partitioning methods, Hierarchical Methods; Transactional Patterns and other temporal based frequent patterns,

Unit 3: Mining Time series Data, Periodicity Analysis for time related sequence data, Trend analysis, Similarity search in Time-series analysis;

Unit 4: Mining Data Streams, Methodologies for stream data processing and stream data systems, Frequent pattern mining in stream data, Sequential Pattern Mining in Data Streams, Classification of dynamic data streams, Class Imbalance Problem; Graph Mining; Social Network Analysis;modulation for communication, filtering, feedback control systems.

Unit 5: Web Mining, Mining the web page layout structure, mining web link structure, mining multimedia data on the web, Automatic classification of web documents and web usage mining; Distributed Data Mining.

Unit 6: Recent trends in Distributed Warehousing and Data Mining, Class Imbalance Problem; Graph Mining; Social Network Analysis

Text Book:

1. Data Warehousing Fundamentals for IT Professionals, Second Edition by Paulraj Ponniah, Wiley India.
2. Data Warehousing, Data Mining, & OLAP – Second Edition by Alex Berson and Stephen J. Smith, Tata McGraw Hill Education
3. Data warehouse Toolkit by Ralph Kimball, Wiley India

Reference Book:

1. Data Mining & Warehousing by Ikvinderpal Singh, Khanna Publishing House
2. Jiawei Han and M Kamber, Data Mining Concepts and Techniques,, Second Edition, Elsevier Publication, 2011.

3. Vipin Kumar, Introduction to Data Mining - Pang-Ning Tan, Michael Steinbach, Addison Wesley,2006

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	3	0	1	1	1	3	2	2	3	2	2
CO2	3	2	3	2	2	1	1	1	2	2	2	3	2	2
CO3	3	2	3	2	3	1	1	1	2	2	2	3	2	2
CO4	3	1	2	1	2	1	1	1	2	2	2	3	2	1
CO5	3	1	2	1	0	1	1	1	3	2	2	3	2	2

1= LOW, 3= HIGH

Course Name: Natural Language Processing

Course Code: DS603

Contact (L: T: P): 3:0:0

Total Contact Hours: 40

Credit: 3

Prerequisites:

Data Structures, Probability & Statistics, Machine Learning, Python Programming

Course Objectives:

Understand the fundamentals of human language processing.

Apply algorithms for linguistic data processing.

Build NLP applications like text classification, machine translation, and question answering.

Explore modern NLP with machine learning and deep learning models.

Course Outcomes (COs):

After completion of the course students will be able to

CO1	**Understand** the fundamentals and challenges of NLP
CO2	**Analyze and apply** linguistic concepts to NLP problems
CO3	**Implement** traditional and statistical NLP algorithms
CO4	**Apply** deep learning techniques to NLP applications
CO5	**Design** and evaluate real-world NLP systems considering ethics

Course Content:

Module – 1:

Introduction to NLP and Linguistics: What is NLP? Goals, History, and Applications

NLP tasks: Classification, Parsing, Translation, Information Retrieval

Linguistic essentials: Phonology, Morphology, Syntax, Semantics, Pragmatics, Language ambiguity and challenges, Overview of NLP pipeline

Module – 2:

Text Processing and Morphology: Regular Expressions, Tokenization, Stemming, Lemmatization, Stop word removal, normalization, Morphological analysis: Inflectional and Derivational morphology Finite State Automata (FSA) for token recognition

Module – 3:

Syntax and Parsing: Part-of-Speech (POS) Tagging: Rule-based, HMM-based, CRF-based, Context-Free Grammar (CFG), Top-down and Bottom-up parsing, Shift-reduce parsing, Dependency parsing.

Module 4:

Semantics and Discourse Analysis: Word Sense Disambiguation (WSD), Lexical semantics: WordNet, semantic roles, Compositional semantics, Discourse structure and coherence, Coreference resolution.

Module 5:

Statistical NLP & Language Modeling: Probability and statistics for NLP, N-gram Language Models and Smoothing, Perplexity, Zipf's Law, Text classification using: Naive Bayes, Logistic Regression, Support Vector Machines (SVM).

Module 6:

Sequence Models & Information Extraction: Hidden Markov Models (HMM), Conditional Random Fields (CRF), Named Entity Recognition (NER), Chunking and shallow parsing, Information Extraction and Retrieval.

Module 7:

Deep Learning for NLP: Word Embeddings: Word2Vec, GloVe, FastText, Recurrent Neural Networks (RNN), LSTM, GRU, Attention mechanism, Transformers and BERT overview, Transfer learning in NLP

Module 8:

Applications and Case Studies: Machine Translation (Statistical vs Neural), Text Summarization: Extractive and Abstractive, Question Answering systems, Chatbots and Dialogue Systems, Ethical considerations in NLP: Bias, Fairness, Privacy.

Text Book:

1. **Neural Network Methods in NLP** – Yoav Goldberg
2. **Natural Language Processing with Python** – Steven Bird, Ewan Klein, and Edward

Reference Book:

1. **Foundations of Statistical NLP** – Christopher D. Manning & Hinrich Schütze
2. **Speech and Language Processing** – Daniel Jurafsky & James H. Martin

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	1	3	0	1	1	1	3	1	2	2	2	2
CO2	3	1	3	2	3	1	1	1	1	1	1	2	2	2
CO3	3	3	3	2	3	3	1	1	2	1	1	3	2	2
CO4	1	3	3	1	3	1	1	1	2	2	2	3	2	1
CO5	1	1	3	1	0	3	3	1	3	2	3	3	2	2

Course Name: Compiler Design

Course Code: DS604A

Contact (L: T: P): 3:0:0

Total Contact Hours: 40

Credit: 3

Prerequisites:

Data Structures, Formal Languages and Automata Theory, Computer Architecture

Course Objectives:

Understand the phases of a compiler.

Learn lexical, syntax, and semantic analysis.

Apply code generation and optimization techniques.

Design a mini-compiler using tools like Lex and Yacc.

Course Outcomes (COs):

After completion of the course students will be able to

CO1	**Understand** the structure and phases of a compiler**.**
CO2	**Apply** lexical and syntax analysis techniques using formal methods**.**
CO3	**Analyze** semantic rules and generate intermediate code.
CO4	**Implement** code optimization and generation strategies.
CO5	**Design** and build a working mini-compiler.

Course Content:

Module – 1:

Introduction to Compilers: Compiler vs Interpreter, Phases of Compiler, Compiler Construction Tools Bootstrapping and Cross Compilation

Module – 2:

Lexical Analysis: Role of Lexical Analyzer, Regular Expressions NFA and DFA, Lex tool

Module – 3:

Syntax Analysis: Context-Free Grammars, Parse Trees Top-down parsing (Recursive Descent, Predictive) Bottom-up parsing (LR, SLR, LALR), Yacc tool

Module 4:

Semantic Analysis and Intermediate Code: Syntax-Directed Definitions and Translation Type Checking, Intermediate Code Generation Three-Address Code, Quadruples

Module 5:

Runtime Environments: Activation Records, Symbol Table Management, Heap and Stack Memory Allocation

Module 6:

Code Optimization: Basic Block Optimization - Loop Optimization, Peephole Optimization

Module 7:

Code Generation: Instruction Selection and Scheduling, Register Allocation, DAG for Basic Blocks

Module 8:

Advanced Topics: Just-In-Time Compilation, LLVM Introduction to Garbage Collection and Parallelizing Compilers

Text Book:

1. Aho, Sethi, Ullman - “Compiler Principles, Techniques and Tools” - Pearson Education.
2. Holub - “Compiler Design in C” - PHI.

Reference Book:

3. Modern Compiler Implementation- Appel
4. Compiler Construction– Kenneth Louden

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	1	2	1	1	2	2	1	1	2	2	1	2
CO2	3	2	3	1	3	1	2	2	1	1	2	3	1	1
CO3	2	3	3	3	1	1	2	1	2	1	1	3	2	2
CO4	1	2	3	1	3	2	1	1	2	1	2	2	2	1
CO5	1	1	2	3	3	3	1	1	2	1	1	3	2	2

Course Name: Data Science Use Cases

Course Code: DS604B

Contact (L: T: P): 3:0:0

Total Contact Hours: 40

Credit: 3

Prerequisites:

Introduction to Data Science, Python Programming, Basic Statistics

Course Objectives:

Understand real-world applications of data science across different domains.

Analyze and interpret structured and unstructured data.

Develop problem-solving strategies using ML and analytics tools.

Build and present case studies using industry datasets.

Course Outcomes (COs):

After completion of the course students will be able to

CO1	**Understand and describe real-world data science applications across domains**
CO2	**Analyze structured and unstructured data using data science techniques.**
CO3	**Apply ML algorithms to solve domain-specific problems**
CO4	**Evaluate and interpret results from case studies and projects**
CO5	**Design and present data science solutions with ethical and social considerations**

Course Content:

Module – 1:

Introduction to Data Science Use Cases: Revisiting Data Science Pipeline**,** CRISP-DM Model and Business Understanding, Data Sources: APIs, Web, Databases, Understanding Problem Statements in Industry.

Module – 2:

Use Cases in Retail and E-Commerce: Recommendation Systems**,** Price Optimization, Inventory Forecasting, Customer Segmentation

Module – 3:

Use Cases in Healthcare: Disease Prediction and Risk Analytics, Medical Image Analysis (intro), Patient Data Classification, Healthcare Fraud Detection.

Module 4:

Use Cases in Finance and Banking: Credit Scoring and Risk Analysis**,** Fraud Detection using Anomaly Detection, Stock Price Prediction (Intro to Time Series), Loan Default Prediction.

Module 5:

Use Cases in Social Media and Entertainment: Sentiment Analysis using NLP, Trend Prediction and Influencer Detection, Content Personalization (OTT Platforms), Fake News Detection.

Module 6:

Government and Smart City Applications: Traffic and Public Transport Optimization, Pollution and Climate Monitoring, Disaster Prediction and Alert Systems, Open Government Data Analysis.

Module 7:

Manufacturing and IoT Analytics: Predictive Maintenance, Quality Control and Defect Detection, Supply Chain Optimization, Sensor Data Analytics.

Module 8:

Use Case for the Above Topics: Predicting Customer Churn (Telecom), Market Basket Analysis (Retail), Building a Collaborative Filtering System, Forecasting product demand using historical sales data, Diabetes Prediction using ML, Detecting anomalies in insurance claims, Credit risk modeling using logistic regression, Visualizing financial trends using time series, Twitter Sentiment Analysis, Movie recommendation using content-based filtering, Predicting traffic congestion using geospatial data, Analyzing air quality using IoT sensor data, Predictive maintenance using sensor logs, Supply chain bottleneck detection,

Text Book:

3. Data Science for Business by Provost & Fawcett
4. Practical Statistics for Data Scientists by Peter Bruce

Reference Book:

5. Hands-On Machine Learning by Aurélien Géron

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	2	2	2	1	1	1	2	1	2
CO2	2	3	3	2	3	1	1	1	1	1	2	3	1	1
CO3	1	2	3	3	3	1	1	2	1	1	3	3	2	2
CO4	2	3	3	3	2	2	1	2	2	1	3	2	2	1
CO5	2	1	2	2	2	2	2	3	3	3	3	3	2	2

Course Name: Mobile Computing

Course Code: DS604C

Contact (L: T: P): 3:0:0

Total Contact Hours: 40

Credit: 3

Prerequisites:

Computer Networks, Operating Systems

Course Objectives:

Understand the principles of mobile computing and wireless communication.

Learn mobile network architectures and protocols.

Explore mobile application development and security concerns.

Gain exposure to real-time issues in mobile environments.

Course Outcomes (COs):

After completion of the course students will be able to

CO1	**Describe mobile computing principles, GSM, and wireless technologies**
CO2	**Analyze and apply mobile and wireless networking protocols including Mobile IP and TCP**
CO3	**Compare traditional computing and mobile computing constraints**
CO4	**Demonstrate mobile application development process using Android SDK**
CO5	**Evaluate issues and solutions for mobility management, performance, and mobile security**

Course Content:

Module – 1:

Introduction to Mobile Computing: Mobile vs Wireless, History and evolution of mobile communication, Types of mobile communication: GSM, CDMA, 3G, 4G, LTE, 5G, Mobile computing applications and limitations.

Module – 2:

Wireless Communication Fundamentals: Wireless Transmission: Frequencies, Signals, Modulation, Multiplexing (FDM, TDM, CDM), Cellular concepts: Frequency reuse, handoff, roaming, GSM architecture and protocols, MAC protocols and types.

Module – 3:

Mobile Network Layer: Mobile IP: Agent discovery, Registration, Tunneling, Dynamic Host Configuration Protocol (DHCP), Routing in mobile environments, Wireless LAN: IEEE 802.11 standards, architecture, MAC layer, Bluetooth and ZigBee.

Module 4:

Mobile Transport Layer & Databases: TCP in Mobile Networks: Indirect TCP, Snooping TCP, Mobile TCP, Performance issues in mobile networks, Mobile Databases: Transaction models, query processing, Data Synchronization and caching.

Module 5:

Mobile Application Development: Android architecture and components, iOS overview, Introduction to mobile SDKs (Android Studio), Location-based services, Mobile security: Authentication, encryption, malware.

Module 6:

Emerging Topics: Mobile cloud computing, Mobile Edge Computing (MEC), Mobile Payments and Mobile Commerce, IoT and Mobility.

Text Book:

5. "Mobile Communications" by Jochen Schiller (Pearson)
6. "Wireless Communications and Networks" by William Stallings

Reference Book:

1. Beginning Android Programming" by Jerome DiMarzio

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	1	1	1	1	1	1	2	1	2
CO2	2	3	3	3	2	1	1	1	1	1	1	3	1	1
CO3	2	2	3	3	3	1	1	1	1	1	2	3	2	2
CO4	1	2	3	2	3	1	1	1	2	3	2	2	2	1
CO5	2	2	2	2	2	2	2	2	2	2	2	3	2	2

Course Name: Generative AI & Explainable AI

Course Code: DS605

Contact (L: T: P): 3:0:0

Total Contact Hours: 40

Credit: 3

Prerequisites:

Machine Learning, Deep Learning, Python Programming

Course Objectives:

Understand principles of generative models and explainability in AI systems.

Learn popular architectures like GANs, VAEs, and diffusion models.

Apply XAI techniques to interpret AI/ML models for ethical and safe decision-making

Gain practical exposure to state-of-the-art tools and real-world applications.

Course Outcomes (COs):

After completion of the course students will be able to

CO1	**Understand and explain core concepts of generative AI and XAI.**
CO2	**Design and implement generative architectures like GANs, VAEs, and transformers.**
CO3	**Evaluate the role of XAI in building transparent, interpretable, and ethical AI systems.**
CO4	**Use XAI techniques like SHAP, LIME, and Grad-CAM in real-world models.**
CO5	**Build end-to-end AI systems that integrate generation and explainability.**

Course Content:

Module – 1:

Foundations of Generative AI: Introduction to Generative AI: Overview and Applications, Probability Distributions & Data Modeling, Generative vs Discriminative Models, Introduction to Latent Space and Sampling.

Module – 2:

Generative Models: Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs): Vanilla, DCGAN, CycleGAN, Diffusion Models and Text-to-Image Generation, Applications: Image synthesis, style transfer, audio generation.

Module – 3:

Transformers and Foundation Models: Transformers Basics (Self-Attention, Positional Encoding), GPT Architecture (GPT-2, GPT-3, GPT-4, ChatGPT

overview), BERT, T5, LLaMA, and Multimodal Models, Prompt Engineering & Fine-Tuning.

Module 4:

Introduction to Explainable AI (XAI): Need for Explainability in AI/ML, Types of XAI: Global vs Local, Model-agnostic vs Model-specific, Regulatory & Ethical Implications (EU AI Act, Bias, Fairness), Real-life failures due to non-transparent AI.

Module 5:

Techniques in XAI: LIME (Local Interpretable Model-Agnostic Explanations), SHAP (SHapley Additive exPlanations), Saliency Maps, Grad-CAM for CNNs, Explainability for NLP models: Attention Visualization, LLM Explainability.

Module 6:

Applications and Case Studies: Medical AI: Generating medical images, explaining predictions, Financial AI: Fraud detection explainability, Legal and Defense AI: Trust in high-stakes systems, Case Studies: DeepFake Detection, Explainability in Chatbots, AI Art

Text Book:

7. "Deep Learning" – Ian Goodfellow, Yoshua Bengio
8. "Interpretable Machine Learning" – Christoph Molnar (open-source)

Reference Book:

1. Transformers for Natural Language Processing, Denis Rothman

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	2	2	1	1	2	1	1	2	2	1	2
CO2	3	3	3	3	3	3	1	1	1	1	2	3	1	1

CO 3	2	3	2	2	1	2	2	3	1	3	2	3	2	2
CO 4	2	3	3	3	3	1	1	1	2	2	2	2	2	1
CO 5	3	2	3	2	3	1	2	2	3	2	3	3	2	2

Course Name: Neural Networks and Deep Learning Lab

Course Code: DS691

Contact (L: T: P): 0:0:3

Total Contact Hours: 13P

Credit: 1.5

Prerequisites:

Python Programming, Basic Machine Learning

Course Objectives:

To implement and train neural networks from scratch and using frameworks.

To understand the internal working of perceptron, activation functions, and optimizers.

To explore CNNs, RNNs, and DNNs for real-world problems.

To get hands-on experience in model evaluation and tuning.

Course Outcomes (COs):

After completion of the course students will be able to

CO1	**Implement basic neural network architectures using NumPy and PyTorch/TensorFlow.**

CO2	**Experiment with different activation functions and loss functions.**
CO3	**Apply CNNs and RNNs for classification and sequence modeling tasks.**
CO4	**Evaluate and fine-tune models using performance metrics and visualization.**
CO5	**Develop mini-projects using deep learning models for real-world datasets.**

Course Content:

1. Introduction to TensorFlow and PyTorch: Tensors, Autograd, and Modules
2. Implement Perceptron Model from Scratch using NumPy
3. Build and Train a Feedforward Neural Network on MNIST dataset using TensorFlow/Keras
4. Experiment with Different Activation Functions (ReLU, Sigmoid, Tanh)
5. Train a Deep Neural Network with Backpropagation and Mini-batch Gradient Descent
6. Build and Evaluate a Convolutional Neural Network (CNN) for Image Classification
7. Implement Recurrent Neural Networks (RNN) or LSTM for Text Sentiment Analysis
8. Use Transfer Learning with Pretrained Models (e.g., ResNet, VGG, MobileNet)
9. Apply Dropout, Batch Normalization, and Early Stopping
10. Perform Model Evaluation: Accuracy, Confusion Matrix, Precision, Recall, F1 Score
11. Hyperparameter Tuning with KerasTuner or Grid Search
12. Mini Project: Build an End-to-End DL Application (face recognition, image classifier, etc.)

Software & Tools Required

Python 3.x, Anaconda or Google Colab, TensorFlow / Keras / PyTorch, OpenCV / NLTK / scikit-learn, Matplotlib / Seaborn

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	2	2	2	3	1	1	1	1	1	2	3	2	2
CO2	3	2	2	2	3	1	1	1	1	2	2	3	2	2
CO3	3	3	3	3	3	1	1	1	1	2	3	3	2	2
CO4	2	2	2	3	2	1	1	1	1	2	3	3	2	1
CO5	3	3	3	3	3	1	1	1	1	3	3	3	2	2

Course Name: Natural Language Processing Lab

Course Code: DS693

Contact (L: T: P): 0:0:3

Total Contact Hours: 13P

Credit: 1.5

Prerequisites:

Python, Basic ML

Course Objectives:

To familiarize students with text processing, tokenization, and language modeling techniques.

To provide hands-on experience with NLP libraries such as NLTK, spaCy, and Hugging Face.

To apply NLP techniques to tasks like sentiment analysis, summarization, and text classification.

To expose students to state-of-the-art language models (e.g., BERT, GPT).

Course Outcomes (COs):

After completion of the course students will be able to

CO1	Apply fundamental NLP techniques such as tokenization, POS tagging, stemming.
CO2	Perform text classification, sentiment analysis, and NER using Python libraries.
CO3	Evaluate language models using statistical and neural methods.
CO4	Implement and fine-tune pre-trained transformer models like BERT and GPT.
CO5	Design and deploy NLP-based mini-projects using real-world datasets.

Course Content:

1. Introduction to NLP libraries: NLTK, spaCy, Transformers
2. Text preprocessing: tokenization, stemming, lemmatization
3. POS tagging and Named Entity Recognition using spaCy
4. Bag of Words, TF-IDF vectorization for document representation
5. Sentiment Analysis using Scikit-learn or Hugging Face Datasets
6. Language Modeling using N-grams and Perplexity
7. Word Embeddings: Word2Vec, GloVe, FastText
8. Build a text classifier using Naive Bayes or Logistic Regression
9. Introduction to Transformers and Fine-tuning BERT on text classification

10. Text Summarization using extractive and abstractive methods
11. Machine Translation using pre-trained T5 or MarianMT models
12. Mini Project: NLP-based chatbot, resume filter, or FAQ bot

Software & Tools Required

Python 3.x, Jupyter Notebooks / Google Colab, NLTK, spaCy, Scikit-learn, Transformers (Hugging Face), TensorFlow / PyTorch, Matplotlib, Seaborn.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	2	2	1	1	1	1	1	1	3	2	2
CO2	3	3	3	3	3	1	1	1	1	2	2	3	2	2
CO3	3	3	3	2	2	1	1	1	1	2	2	3	2	2
CO4	3	3	3	3	3	1	1	1	1	3	2	3	2	1
CO5	3	2	3	3	3	1	1	1	1	3	3	3	2	2

Course Name: Generative AI & Explainable AI Lab
Course Code: DS695
Contact (L: T: P): 0:0:3
Total Contact Hours: 13P
Credit: 1.5

Prerequisites:
Python, Deep Learning, Machine Learning

Course Objectives:

To implement and evaluate generative models like GANs and VAEs.
To understand and apply Explainable AI (XAI) techniques.
To integrate generative and interpretable AI for real-world tasks.
To work with real datasets using industry-standard tools and libraries.

Course Outcomes (COs):

After completion of the course students will be able to

CO1	Understand and implement basic generative models like VAE and GAN.
CO2	Analyze and apply advanced generative models (e.g., StyleGAN, Diffusion Models).
CO3	Apply XAI techniques like LIME, SHAP, and Grad-CAM to interpret model predictions.
CO4	Evaluate generative and interpretable models using real-world datasets.
CO5	Design a project that integrates generation and explainability in AI systems.

Course Content:

1. Introduction to Generative AI and Explainable AI Tools (TensorFlow, PyTorch, Captum, SHAP)
2. Implement a Variational Autoencoder (VAE) on MNIST or Fashion-MNIST
3. Build and Train a Basic GAN for Digit Generation
4. Conditional GAN for Class-Specific Image Generation
5. Introduction to Diffusion Models (Use Hugging Face Datasets or pretrained models)
6. Use Pre-trained Text-to-Image models like DALL·E / Stable Diffusion
7. Implement LIME to explain predictions of a text/image classifier
8. Use SHAP to interpret tabular classification/regression model predictions
9. Use Grad-CAM to visualize and explain decisions made by a CNN on an image classification task
10. Explain Transformer model predictions using attention visualization (e.g., BERT, GPT)
11. Compare model outputs before and after explainability enhancement
12. Mini Project: Create a generative model with built-in interpretability.

Software & Tools Required

Python 3.x, Jupyter / Google Colab, PyTorch / TensorFlow / Keras, Hugging Face Transformers & Datasets, Captum, SHAP, LIME, Grad-CAM, Matplotlib, Seaborn, Streamlit (for app demo)

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	1	1	1	1	1	2	3	2	2
CO2	3	3	3	3	3	1	1	1	1	2	3	3	2	2
CO3	2	3	3	3	2	2	2	2	1	2	3	3	2	2
CO4	3	2	3	3	3	1	1	1	1	2	3	3	2	1
CO5	3	3	3	3	3	2	2	2	2	3	3	3	2	2

4th Year 7th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credits**
					L	T	P	Total	
1	ENGG	Major	DS701	Reinforcement Learning	3	0	0	3	**3**
2	ENGG	Major	DS702A	Quantum Computing	3	0	0	3	**3**
			DS702B	Bio Informatics					
			DS702C	Parallel and Distributed Computing					
3	ENGG	Major	DS703A	Information Theory and Coding	3	0	0	3	**3**
			DS703B	IoT and Data Science Synergy					
			DS703C	Nature-Inspired Computing in Data Science					
4	HUM	Minor	HU(DS)701	Human Resource Development and Organizational Behavior	2	0	0	2	**2**
B. PRACTICAL									
1	ENGG	Major	DS791	Reinforcement Learning Lab	0	0	3	3	**1.5**
2	ENGG	Major	DS792A	Quantum Computing Lab	0	0	3	3	**1.5**
			DS792B	Bio Informatics Lab					
			DS792C	Parallel and Distributed Computing Lab					
			DS793A	Information Theory and Coding Lab					
			DS793B	IoT and Data Science Synergy Lab					
			DS793C	Nature-Inspired Computing in Data Science Lab					
5	PRJ	Project	DS781	Project III	0	0	1	16	**8**

							2		
TOTAL CREDIT								**33**	**22**

Course Name: Reinforcement Learning
Course Code: DS701
Contact (L: T: P): 3:0:0
Total Contact Hours: 36
Credit: 3
Prerequisites:

1. Linear Algebra
2. Probability and Statistics
3. Programming in Python
4. Basic concepts of Machine Learning

Course Objectives:

The objective of the course is to make the students able to -

1. To introduce the fundamentals of Reinforcement Learning (RL) and Markov Decision Processes (MDPs).
2. To explore algorithms like Dynamic Programming, Monte Carlo, and Temporal Difference learning.
3. To build knowledge of policy optimization and value function approximation.
4. To equip students with the practical skills to design intelligent agents for sequential decision-making problems.

Course Outcomes (COs):

After completion of the course students will be able to

CO1	**Understand** the fundamentals and theoretical basis of reinforcement learning.
CO2	**Formulate** problems using the RL framework and Markov Decision Processes.
CO3	**Apply** Dynamic Programming and Monte Carlo methods for policy evaluation and improvement.
CO4	**Analyze** and implement TD learning algorithms and model-free methods like Q-learning and Sarsa.
CO5	**Implement** function approximation techniques, policy gradient methods, and evaluate RL applications.

Course Content:

Module – 1: [6L]

Basics of probability and linear algebra, Stochastic multi-armed bandits, Regret and sublinear regret, Upper Confidence Bound (UCB), KL-UCB, Thompson

Sampling. [6L]

Module – 2: [6L]

Markov Decision Problem (MDP), MDP formulation: States, actions, policies, value functions, Reward models: infinite discounted, finite horizon, average rewards. [3L]

Episodic vs continuing tasks, Bellman's optimality equations, Value iteration and policy iteration. [3L]

Module – 3: [8L]

Prediction and control in RL, Model-based RL, Monte Carlo prediction and control, Online MC policy evaluation, First-visit vs every-visit MC. [8L]

Module 4: [8L]

Temporal Difference (TD) Learning, Bootstrapping and TD(0), Monte Carlo vs TD learning, Batch TD(0). [5L]

Q-learning, Sarsa, Expected Sarsa, Convergence and practical issues. [3L]

Module 5: [8L]

Function Approximation, n-step returns, TD(λ), Linear function approximation, Tile coding, Policy search & gradient methods, Experience replay, Fitted Q Iteration, Case studies and applications. [8L]

Text Book:

1. "Reinforcement learning: An introduction," First Edition, Sutton, Richard S., and Andrew G. Barto, MIT press 2020.
2. "Statistical reinforcement learning: modern machine learning approaches," First Edition, Sugiyama, Masashi. CRC Press 2015.

Reference Book:

1. "Bandit algorithms," First Edition, Lattimore, T. and C. Szepesvári. Cambridge University Press. 2020.
2. "Reinforcement Learning Algorithms: Analysis and Applications," Boris Belousov, Hany Abdulsamad, Pascal Klink, Simone Parisi, and Jan Peters First Edition, Springer 2021.
3. Alexander Zai and Brandon Brown "Deep Reinforcement Learning in Action," First Edition, Manning Publications 2020.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	1	2	1	1	1	1	1	2	2	2	1
CO2	3	3	2	2	3	1	1	1	1	1	2	3	2	2
CO3	3	3	2	3	3	1	1	1	1	1	2	3	3	2
CO4	3	3	3	3	3	1	1	1	1	1	2	3	3	2
CO5	3	3	3	3	3	2	2	2	2	2	3	3	3	3

Course Name: Quantum Computing
Course Code: DS702A
Contact (L: T: P): 3:0:0
Total Contact Hours: 36
Credit: 3

Prerequisites:

- Linear Algebra
- Basic concepts of Probability and Statistics
- Discrete Mathematics
- Basic understanding of classical computing paradigms

Course Objectives:

The objective of the course is to make the students able to -

- To introduce the fundamental principles of quantum mechanics relevant to quantum computing.
- To understand the concepts of qubits, superposition, entanglement, and quantum gates.
- To explore various quantum algorithms and their potential computational advantages.
- To learn about different quantum computing models and architectures.
- To grasp the challenges and future prospects of quantum computing.

Course Outcomes (COs):

After completion of the course, students will be able to

CO1: Understand the basic principles of quantum mechanics and their application in quantum computing.
CO2: Explain the concepts of qubits, quantum superposition, and quantum entanglement.
CO3: Analyze and construct elementary quantum circuits using quantum gates.
CO4: Describe and apply fundamental quantum algorithms, such as Deutsch-Jozsa and Grover's algorithms.
CO5: Discuss the challenges and potential impact of quantum computing on various fields.

Course Content:

Module – 1: Foundations of Quantum Mechanics for Computing [8L]

Introduction to quantum mechanics: States, vectors, inner products, operators, eigenvalues, eigenvectors. [3L]
Postulates of quantum mechanics: State space, evolution, measurement. [3L]
Dirac notation, Pauli matrices, Bloch sphere representation of a qubit. [2L]

Module – 2: Qubits, Quantum Gates, and Circuits [8L]

Qubits vs. classical bits, single-qubit gates (Hadamard, Pauli-X, Y, Z, phase gates). [4L]
Multi-qubit systems, tensor products, quantum entanglement, Bell states. [2L]
Controlled gates (CNOT, Toffoli), universal gate sets, quantum circuit model. [2L]

Module – 3: Quantum Algorithms I [7L]

Deutsch-Jozsa algorithm: Problem statement, classical vs. quantum solution, circuit implementation. [3L]
Quantum Fourier Transform (QFT): Properties, circuit implementation. [2L]
Phase Estimation Algorithm (PEA): Introduction and applications. [2L]

Module 4: Quantum Algorithms II [7L]

Shor's algorithm: Overview of factoring, order-finding problem (conceptual). [3L]
Grover's search algorithm: Problem statement, amplitude amplification, circuit (conceptual). [4L]

Module 5: Quantum Computing Architectures and Challenges [6L]

Physical realizations of qubits: Superconducting circuits, trapped ions, photonic qubits (overview). [3L]
Quantum error correction: Introduction to noise, basic concepts of quantum error correction (conceptual). [2L]
Current challenges and future prospects of quantum computing, quantum supremacy, and applications. [1L]

Textbooks:

1. "Quantum Computation and Quantum Information," 10th Anniversary Edition, Michael A. Nielsen and Isaac L. Chuang, Cambridge University Press, 2010.

2. "Quantum Computing for Computer Scientists," 1st Edition, Noson S. Yanofsky and Mirco A. Mannucci, Cambridge University Press, 2008.

Reference Book:

1. "Introduction to Quantum Mechanics," 3rd Edition, David J. Griffiths and Darrell F. Schroeter, Cambridge University Press, 2018.
2. "Dancing with Qubits: How quantum mechanics creates a new kind of computing," 1st Edition, Robert S. Sutor, Packt Publishing, 2019.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	2	1	1	1	1	1	1	1	1	1	2	1	1
CO2	1	3	2	2	2	1	1	2	2	1	1	3	2	1
CO3	1	3	2	3	3	1	1	3	3	1	1	3	2	2
CO4	2	3	3	3	3	2	1	3	3	1	2	3	3	2
CO5	2	3	3	3	3	2	2	3	3	2	2	3	3	3

Course Name: Bioinformatics
Course Code: DS702B
Contact (L: T: P): 3:0:0
Total Contact Hours: 36
Credit: 3

Prerequisites:

- Basic concepts of Biology/Molecular Biology (recommended)
- Programming in Python (familiarity with data structures)
- Linear Algebra
- Probability and Statistics

Course Objectives:
The objective of the course is to make the students able to -

- To introduce the fundamental concepts and challenges in bioinformatics.
- To understand various biological databases and data formats.
- To learn algorithms for sequence alignment and analysis.
- To explore methods for phylogenetic analysis and gene prediction.

- To grasp the principles of protein structure prediction and functional genomics.

Course Outcomes (COs):
After completion of the course, students will be able to
CO1: Understand the scope and fundamental concepts of bioinformatics and its applications.

CO2: Utilize various biological databases and interpret different biological data formats.

CO3: Apply sequence alignment algorithms for DNA and protein sequences.

CO4: Construct phylogenetic trees and perform basic gene prediction.

CO5: Describe methods for protein structure prediction, functional genomics, and proteomics.

Course Content:
Module – 1: Introduction to Bioinformatics and Biological Databases [7L]
Introduction to Bioinformatics: Definition, scope, applications, major research areas. [3L]
Biological Data Types: DNA, RNA, Protein sequences and structures. [2L]
Biological Databases: NCBI, GenBank, EMBL, DDBJ, UniProt, PDB. [2L]

Module – 2: Sequence Alignment [8L]
Pairwise Sequence Alignment: Global (Needleman-Wunsch) and Local (Smith-Waterman) algorithms. [4L]
Scoring matrices (PAM, BLOSUM), Gap penalties. [2L]
Heuristic alignment algorithms: BLAST and FASTA. [2L]

Module – 3: Multiple Sequence Alignment and Phylogenetics [7L]
Multiple Sequence Alignment (MSA): CLUSTAL, T-Coffee. [3L]
Applications of MSA: Conserved regions, functional domains. [2L]
Phylogenetic Analysis: Basic concepts, types of trees, methods (distance-based, character-based). [2L]
Module 4: Gene Prediction and Genome Annotation [7L]
Gene Prediction in Prokaryotes and Eukaryotes: Open Reading Frames (ORFs), gene-finding algorithms. [3L]
Genome Annotation: Functional and structural annotation. [2L]
Comparative Genomics: Synteny, gene duplication. [2L]

Module 5: Protein Structure Prediction and Functional Genomics [7L]

Protein Structure: Primary, secondary, tertiary, and quaternary structures. [2L]
Protein Structure Prediction: Homology modeling, threading, de novo methods (conceptual). [3L]
Functional Genomics and Proteomics: Microarrays, RNA-Seq, Mass Spectrometry (introduction). [2L]

Textbooks:

1. "Bioinformatics: Sequence and Genome Analysis," 3rd Edition, David W. Mount, Cold Spring Harbor Laboratory Press, 2017.
2. "Bioinformatics and Functional Genomics," 3rd Edition, Jonathan Pevsner, Wiley-Blackwell, 2015.

Reference Book:

"Computational Molecular Biology: An Algorithmic Approach," 1st Edition, Pavel Pevzner, MIT Press, 2000.
"Essential Bioinformatics," 1st Edition, Jin Xiong, Cambridge University Press, 2006.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	1	1	1	2	1	1	2	1	1	2	1	2
CO2	3	2	2	2	2	2	2	1	2	2	1	3	2	2
CO3	3	2	2	3	2	2	1	2	1	2	2	3	2	2
CO4	2	3	3	3	3	3	2	2	2	2	2	3	3	2
CO5	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Course Name: Parallel and Distributed Computing
Course Code: DS702C
Contact (L: T: P): 3:0:0
Total Contact Hours: 36
Credit: 3

Prerequisites:

- Data Structures and Algorithms
- Operating Systems
- Computer Architecture
- Programming in C/C++ or Java

Course Objectives:
The objective of the course is to make the students able to -

- To understand the fundamental concepts and models of parallel and distributed computing.
- To explore various parallel architectures and programming paradigms.
- To learn about techniques for designing and analyzing parallel algorithms.
- To grasp the challenges and solutions related to communication, synchronization, and consistency in distributed systems.
- To gain insights into contemporary distributed computing systems and their applications.

Course Outcomes (COs):

After completion of the course, students will be able to
CO1: Understand the basic principles and models of parallel and distributed computing.
CO2: Analyze different parallel architectures and their impact on algorithm design.
CO3: Design and implement parallel algorithms using shared-memory and message-passing paradigms.
CO4: Explain the concepts of consistency, consensus, and fault tolerance in distributed systems.
CO5: Evaluate the performance and scalability of parallel and distributed applications.
Course Content:
Module – 1: Introduction to Parallel Computing [7L]
Introduction to Parallel Computing: Concepts, motivations, parallelism vs.

concurrency. [3L]
Parallel Computer Models: Shared-memory, distributed-memory, hybrid models. [2L]
Performance Metrics: Speedup, efficiency, Amdahl's Law, scalability. [2L]

Module – 2: Parallel Architectures and Programming Paradigms [8L]
Shared Memory Architectures: Multicore processors, GPUs, cache coherence. [3L]
Programming with Threads: POSIX Threads (pthreads), OpenMP directives. [3L]
Distributed Memory Architectures: Interconnection networks, message passing interface (MPI). [2L]

Module – 3: Parallel Algorithm Design [7L]

Design Methodologies: Partitioning, communication, agglomeration, mapping. [3L]
Parallel Algorithms for Basic Tasks: Sorting, matrix multiplication. [2L]
Synchronization and Load Balancing: Critical sections, barriers, locks. [2L]

Module 4: Distributed Computing Fundamentals [7L]
Introduction to Distributed Systems: Characteristics, goals, challenges. [2L]
Communication in Distributed Systems: Remote Procedure Calls (RPC), Message Queues. [2L]
Distributed File Systems: DFS concepts, consistency models. [3L]
Module 5: Consistency, Fault Tolerance, and Advanced Topics [7L]
Consistency Models: Sequential, causal, eventual consistency. [2L]
Distributed Consensus: Paxos, Raft (conceptual overview). [2L]
Fault Tolerance and Replication: Techniques for handling failures. [1L]
Cloud Computing and Big Data Systems (e.g., MapReduce, Spark) - Introduction. [2L]

Textbooks:

1. "Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers," 2nd Edition, Barry Wilkinson and Michael Allen, Prentice Hall, 2005.
2. "Distributed Systems: Concepts and Design," 5th Edition, George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, John Wiley & Sons, 2011.

Reference Book:

1. "An Introduction to Parallel Programming," Peter S. Pacheco, Morgan Kaufmann, 2011.

2. "Designing Data-Intensive Applications," 1st Edition, Martin Kleppmann, O'Reilly Media, 2017.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	2	1	1	1	1	1	1	1	1	1	2	1	1
CO2	1	3	2	2	2	1	1	2	2	1	1	3	2	1
CO3	1	3	2	3	3	1	1	3	3	1	1	3	2	2
CO4	2	3	3	3	3	2	1	3	3	1	2	3	3	2
CO5	2	3	3	3	3	2	2	3	3	2	2	3	3	3

Course Name: Information Theory and Coding
Course Code: DS703A
Contact (L: T: P): 3:0:0
Total Contact Hours: 36
Credit: 3
Prerequisites:

1. Probability and Random Processes
2. Basic knowledge of Digital Communication Systems
3. Linear Algebra
4. Discrete Mathematics

Course Objectives:
The objective of the course is to make the students able to -

1. To understand the fundamental concepts of information theory, including entropy, mutual information, and channel capacity.
2. To learn various source coding techniques for data compression.
3. To grasp the principles of channel coding for error detection and correction.
4. To analyze and apply different types of error control codes, including linear block codes, cyclic codes, and convolutional codes.
5. To design and evaluate coding schemes for reliable data transmission over noisy channels.

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

CO1	**Understand** the basic concepts of information theory, including entropy, mutual information, and data processing inequality.
CO2	**Apply** various source coding algorithms for efficient data

	compression.
CO3	**Analyze** the capacity of different communication channels and the implications of Shannon's theorems.
CO4	**Design** and **implement** linear block codes and cyclic codes for error detection and correction.
CO5	**Evaluate** the performance of convolutional codes and other advanced coding techniques.

Course Content:

Module – 1: Introduction to Information Theory [7L]
Introduction to Information Theory, Uncertainty, Information and Entropy, Joint and Conditional Entropy, Relative Entropy and Mutual Information. [4L]
Chain Rules for Entropy and Mutual Information, Data Processing Inequality, Fano's Inequality. [3L]

Module – 2: Source Coding [7L]
Kraft's Inequality, Uniquely Decodable Codes, Prefix Codes. [2L], Huffman Coding, Shannon-Fano-Elias Coding, Arithmetic Coding. [3L], Lempel-Ziv-Welch (LZW) Algorithm, Universal Source Coding. [2L]

Module – 3: Channel Capacity [7L]
Discrete Memoryless Channels (DMC), Symmetric Channels. [2L], Channel Capacity, Noisy Channel Coding Theorem, Shannon Limit. [3L], Differential Entropy, Gaussian Channel, Rate Distortion Theory. [2L]
Module 4: Error Control Coding - Linear Block Codes [8L]
Introduction to Error Control Coding, Types of Errors. [2L], Linear Block Codes, Generator and Parity Check Matrices. [3L], Syndrome Decoding, Hamming Codes, Minimum Distance. [3L]

Module 5: Cyclic Codes and Convolutional Codes [7L]
Cyclic Codes, Generator Polynomials, Encoding and Decoding of Cyclic Codes. [3L], BCH Codes, Reed-Solomon Codes (Introduction). [2L], Convolutional Codes, Tree, Trellis, and State Diagrams, Viterbi Algorithm (Concept). [2L]

Textbook:

1. "Elements of Information Theory," 2nd Edition, Thomas M. Cover and Joy A. Thomas, Wiley 2006.

2. "Digital Communications," 5th Edition, John G. Proakis and Masoud Salehi, McGraw-Hill Education 2008.

Reference Book:

1. "Information Theory, Coding and Cryptography," 3rd Edition, Ranjan Bose, Tata McGraw-Hill 2016.
2. "Error Control Systems for Digital Communication and Storage," 2nd Edition, Stephen B. Wicker, Prentice Hall 1995.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	1	1	1	2	1	1	2	1	1	2	1	2	2
CO2	1	1	1	1	2	1	1	2	1	1	2	1	2	2
CO3	1	1	1	1	2	1	1	2	1	1	2	1	2	2
CO4	1	1	1	1	2	1	1	2	1	1	2	1	2	2
CO5	1	1	1	1	2	1	1	2	1	1	2	1	2	2

Course Name: IoT and Data Science Synergy
Course Code: DS703B
Contact (L: T: P): 3:0:0
Total Contact Hours: 36

Credit: 3
Prerequisites:

1. Basic concepts of IoT
2. Fundamentals of Data Science and Machine Learning
3. Programming in Python
4. Database management concepts

Course Objectives:
The objective of the course is to make the students able to -

1. To understand the synergistic relationship between IoT and Data Science.
2. To learn how data is collected, processed, and analyzed from IoT devices.
3. To explore various data science techniques applicable to IoT data, including machine learning and deep learning.
4. To gain insights into real-world applications of IoT and Data Science across different domains.
5. To address challenges related to data security, privacy, and scalability in IoT-Data Science ecosystems.

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

CO1	**Understand** the architectural components of IoT systems and the role of data in them.

CO2	**Identify** and apply appropriate data collection, pre-processing, and storage techniques for IoT data.
CO3	**Analyze** IoT data using various machine learning and statistical methods to extract valuable insights.
CO4	**Develop** and **implement** data-driven solutions for specific IoT applications.
CO5	**Address** and **propose** solutions for security, privacy, and ethical challenges in IoT data ecosystems.

Course Content:

Module – 1: Introduction to IoT and Data Science Integration [7L]
Introduction to Internet of Things (IoT) architecture, key components, and communication protocols. [3L], Overview of Data Science concepts, stages of data analysis, and their relevance to IoT. [2L]
Synergy between IoT and Data Science, use cases, and emerging trends. [2L]

Module – 2: IoT Data Acquisition and Management [7L]
Sensors and Actuators, Data acquisition methods from IoT devices. [3L], Edge computing, Fog computing, and Cloud computing in IoT data processing. [2L], IoT data storage solutions: Time-series databases, NoSQL databases. [2L]

Module – 3: Data Pre-processing and Feature Engineering for IoT [7L]
Handling noisy, missing, and inconsistent IoT data. [3L], Data cleaning, normalization, and transformation techniques for time-series data. [2L], Feature extraction and selection from raw sensor data. [2L]

Module 4: Machine Learning and Deep Learning for IoT Data [8L]
Supervised learning for IoT applications (e.g., predictive maintenance, anomaly detection). [3L]
Unsupervised learning for IoT (e.g., clustering, pattern recognition). [3L]
Introduction to deep learning for complex IoT data analysis (e.g., CNNs for image data, LSTMs for time series). [2L]

Module 5: IoT Data Analytics Applications and Challenges [7L]
Case studies in Smart Cities, Healthcare, Industrial IoT, and Smart Homes. [3L]
Security and Privacy concerns in IoT data. [2L]
Scalability, interoperability, and ethical considerations in IoT-Data Science solutions. [2L]

Textbook:

1. "IoT and Data Science: The Ultimate Guide for Beginners and Experts," 1st Edition, Andrew Mason, Independently published 2023.

2. "Machine Learning with R: Expert techniques for predictive modeling," 3rd Edition, Brett Lantz, Packt Publishing 2019 (Focus on data analysis principles applicable to IoT).

Reference Book:

1. "Data Science for IoT: Principles and Applications," 1st Edition, Arshdeep Bahga and Vijay Madisetti, Arshdeep Bahga and Vijay Madisetti 2021.
2. "Practical IoT Hacking: The Definitive Guide to Attacking the Internet of Things," 1st Edition, Fotios Chantzis, Ioannis Stais, Evangelos Deirmentzoglou, and Georgios Karopoulos, No Starch Press 2020.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	1	1	1	1	1	1	1	1	2	1	1
CO2	3	3	2	2	2	1	1	1	1	1	1	3	2	1
CO3	3	3	2	3	2	1	1	1	1	1	1	3	2	2
CO4	3	3	3	3	3	2	1	1	1	1	2	3	3	2
CO5	3	3	3	3	3	2	2	2	2	2	2	3	3	3

Course Name: Nature-Inspired Computing in Data Science
Course Code: DS703C
Contact (L: T: P): 3:0:0
Total Contact Hours: 36
Credit: 3

Prerequisites:

1. Basic concepts of Data Science and Machine Learning
2. Programming in Python
3. Fundamentals of Optimization
4. Basic understanding of algorithms and data structures

Course Objectives:

The objective of the course is to make the students able to -

1. To introduce the fundamental concepts of nature-inspired computing algorithms.
2. To explore the application of nature-inspired algorithms in solving complex data science problems.

3. To understand the principles behind various metaheuristics, including evolutionary algorithms, swarm intelligence, and immune-inspired algorithms.
4. To equip students with the practical skills to implement and evaluate nature-inspired algorithms for optimization, feature selection, clustering, and classification tasks in data science.
5. To analyze the strengths and limitations of different nature-inspired approaches for real-world data challenges.

Course Outcomes (COs):

After completion of the course, students will be able to

CO1	**Understand** the core concepts and principles of various nature-inspired computing paradigms.
CO2	**Apply** nature-inspired algorithms to solve optimization problems in data science.
CO3	**Analyze** and **implement** evolutionary algorithms for tasks like feature selection and hyperparameter tuning.
CO4	**Utilize** swarm intelligence techniques for clustering and classification in large datasets.
CO5	**Evaluate** the performance of different nature-inspired algorithms on real-world data science problems.

Course Content:

Module – 1: Introduction to Nature-Inspired Computing [6L]

Introduction to nature-inspired computing, Biological and natural metaphors for computation. [2L]

Overview of optimization problems in data science, Metaheuristics vs. Heuristics. [2L]

Problem representation, Fitness functions, Search space. [2L]

Module – 2: Evolutionary Algorithms [8L]

Genetic Algorithms (GAs): Basic operators (selection, crossover, mutation). [3L]

Genetic Programming (GP), Evolutionary Strategies (ES). [2L]

Applications in data science: Feature selection, Hyperparameter optimization, Model training. [3L]

Module – 3: Swarm Intelligence [8L]

Particle Swarm Optimization (PSO): Principles, variations, and applications. [3L]

Ant Colony Optimization (ACO): Principles and application to combinatorial

problems. [3L]
Other swarm intelligence algorithms: Bee Colony Optimization, Firefly Algorithm. [2L]

Module 4: Immune-Inspired and Other Nature-Inspired Algorithms [8L]
Artificial Immune Systems (AIS): Clonal Selection, Negative Selection. [3L]
Simulated Annealing (SA), Tabu Search (TS). [2L]
Harmony Search (HS), Bat Algorithm, Cuckoo Search (CS). [3L]

Module 5: Applications and Challenges in Data Science [6L]
Nature-inspired algorithms for clustering: K-means variants, density-based clustering. [2L]
Nature-inspired algorithms for classification: Rule extraction, ensemble methods. [2L]
Challenges and future directions: Hybrid approaches, Big Data, Ethical considerations. [2L]

Textbook:

1. "Nature-Inspired Optimization Algorithms," 2nd Edition, Xin-She Yang, Elsevier 2020.
2. "Computational Intelligence: An Introduction," 3rd Edition, Andries P. Engelbrecht, Wiley 2007.

Reference Book:

1. "Swarm Intelligence," 2nd Edition, James Kennedy, Russell C. Eberhart, and Yuhui Shi, Morgan Kaufmann 2001.
2. "Genetic Algorithms in Search, Optimization, and Machine Learning," David E. Goldberg, Addison-Wesley 1989.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	-	-	-	-	-	-	-	-	3	3
CO2	3	3	3	3	-	-	-	-	-	-	-	-	3	2
CO3	3	3	3	3	-	-	-	-	-	-	-	3	3	3
CO4	3	3	3	3	-	-	-	-	-	-	-	-	3	3
CO5	3	3	3	3	-	-	-	-	3	2	-	-	3	2

Course Name: Human Resource Development and Organizational Behavior
Course Code: HU(DS)701
Contact (L: T: P): 2:0:0
Total Contact Hours: 24
Credit: 2
Prerequisites:

1. Basic Understanding of Management Principles
2. Communication and Interpersonal Skills
3. Knowledge of Psychology and Sociology
4. Awareness of Organizational Structure and Culture
5. Interest in People and Development

Course Objectives:

The objective of the course is to make the students able to -

1. To understand the fundamental concepts of Human Resource Development (HRD) and Organizational Behavior (OB).
2. To explore individual, group, and organizational dynamics that influence behavior in the workplace.
3. To learn about key HRD functions such as training, performance management, and career development.
4. To analyze how organizational culture and change impact employee behavior and organizational effectiveness.
5. To develop an appreciation for ethical considerations and contemporary challenges in HRD and OB.

Course Outcomes (COs):

After completion of the course, students will be able to

CO1	**Understand** the basic principles and theories of HRD and Organizational Behavior.
CO2	**Analyze** individual behavioral components, such as perception, motivation, and personality, within an organizational context.
CO3	**Examine** group dynamics, team building, and conflict resolution within organizations.
CO4	**Explain** the core functions of Human Resource Development and their strategic importance.
CO5	**Evaluate** the impact of organizational culture, leadership, and change management on organizational effectiveness.

Course Content:

Module – 1: Introduction to HRD and Organizational Behavior [4L]

Introduction to Human Resource Development (HRD): Concept, objectives, and

scope. [2L] Introduction to Organizational Behavior (OB): Definition, nature, goals, and contributing disciplines. [2L]

Module – 2: Individual Behavior in Organization [5L]
Foundations of Individual Behavior: Biographical characteristics, ability, and learning. [2L] Perception, Personality, and Values: Determinants, theories (e.g., Big Five), and their relevance to OB. [2L] Motivation: Theories (e.g., Maslow, Herzberg, Expectancy theory) and applications. [1L]

Module – 3: Group Dynamics and Teamwork [5L]
Group Dynamics: Types of groups, stages of group development, group properties (roles, norms, status, cohesion). [2L] Teamwork: Concept of teams, types of teams, creating effective teams. [2L] Conflict and Negotiation: Sources of conflict, conflict management styles, negotiation strategies. [1L]

Module 4: Human Resource Development Functions [5L]
Training and Development: Need assessment, methods, and evaluation. [2L] Performance Management: Process, appraisal methods, feedback. [1L] Career Development: Planning, stages, mentoring. [1L] Employee Engagement and Retention Strategies. [1L]

Module 5: Organizational Culture and Change [5L]
Organizational Culture: Components, types, functions, creating and sustaining culture. [2L] Leadership: Theories (e.g., trait, behavioral, transformational), leadership styles. [1L] Organizational Change: Forces for change, resistance to change, managing change. [2L]

Textbook:

1. "Organizational Behavior," 18th Edition, Stephen P. Robbins and Timothy A. Judge, Pearson 2022.
2. "Human Resource Development: A Concise Introduction," 1st Edition, Andrew D. Brown and Sarah E. James, SAGE Publications, 2017.

Reference Book:

1. "Human Resource Management," 16th Edition, Gary Dessler, Pearson 2020.
2. "Understanding and Managing Organizational Behavior," 6th Edition, Jennifer M. George and Gareth R. Jones, Pearson, 2012.

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	2	-	-	-	-	-	2	2	1	1

CO2	3	3	3	3	-	-	2	-	-	-	2	2	1	1
CO3	3	3	-	-	-	-	3	-	-	-	3	2	2	2
CO4	3	3	3	2	-	-	3	-	-	-	3	2	2	2
CO5	3	3	3	3	2	-	-	-	-	-	2	2	3	3

Course Name: Reinforcement Learning Lab (PEC-CS-T-791)

Course Code: DS791
Contact Hours (L: T P): 0:0:3
Credits: 1.5
Prerequisites

• Linear Algebra
• Probability and Statistics
• Python Programming
• Basics of Machine Learning
Course Objectives

• To provide hands-on implementation of Reinforcement Learning (RL) concepts.
• To explore and simulate classical RL algorithms including Dynamic Programming, Monte Carlo methods, and Temporal Difference learning.
• To apply RL for solving Markov Decision Processes (MDPs).
• To implement policy optimization and function approximation in realistic environments.
• To build intelligent agents using Python and OpenAI Gym.

Course Outcomes (COs)

CO1: Understand and simulate multi-armed bandit problems and their strategies.
CO2: Model decision-making problems using the RL framework and solve using MDP solvers.
CO3: Implement dynamic programming and Monte Carlo algorithms for value prediction and policy improvement.
CO4: Develop TD learning agents using Sarsa and Q-learning methods.
CO5: Design and test advanced agents using function approximation, policy gradients, and experience replay

.

Lab Module-Wise Course Content

Module 1: Multi-Armed Bandits and Exploration Strategies

• Implement a basic stochastic multi-armed bandit environment.
• Simulate and compare strategies: ε-greedy, UCB, KL-UCB, and Thompson Sampling.

Module 2: Markov Decision Processes (MDPs)

• Define a grid world MDP with specified rewards and actions.
• Implement value iteration and policy iteration on MDPs.

Module 3: Monte Carlo Methods

• Implement first-visit and every-visit Monte Carlo for policy evaluation.
• Apply Monte Carlo control for epsilon-soft policy improvement.

Module 4: Temporal Difference (TD) Learning

• Implement TD (0) for value estimation.
• Develop Sarsa, Q-learning, and Expected Sarsa agents in OpenAI Gym.

Module 5: Function Approximation and Advanced Techniques

• Use linear function approximation with Q-learning.
• Implement policy gradient and actor-critic methods using neural networks.
• Apply experience replay and Fitted Q-Iteration for training agents.

Textbooks & References

Textbooks

1. Sutton, Richard S., and Andrew G. Barto. Reinforcement Learning: An Introduction, MIT Press, 2020.
2. Sugiyama, Masashi. Statistical Reinforcement Learning: Modern Machine Learning Approaches, CRC Press, 2015.

References

1. Lattimore, T., and C. Szepesvári. Bandit Algorithms, Cambridge University Press, 2020.
2. Boris Belousov et al. Reinforcement Learning Algorithms: Analysis and Applications, Springer, 2021.
3. Alexander Zai and Brandon Brown. Deep Reinforcement Learning in Action, Manning Publications, 2020.

CO-PO and CO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	1	2	1	1	1	1	1	2	2	2	1
CO2	3	3	2	2	3	1	1	1	1	1	2	3	2	2
CO3	3	3	2	3	3	1	1	1	1	1	2	3	3	2
CO4	3	3	3	3	3	1	1	1	1	1	2	3	3	2
CO5	3	3	3	3	3	2	2	2	2	2	3	3	3	3

Course Name: Quantum Computing Lab
Course Code: DS792A
Contact Hours (L: T: P): 0:0:3
Credits: 1.5

Prerequisites:

- Linear Algebra
- Basic concepts of Quantum Mechanics (recommended, but not strictly mandatory for lab focus)
- Programming in Python (familiarity with NumPy)

Course Objectives:

- To provide hands-on experience with fundamental concepts of quantum computing.
- To implement basic quantum circuits and algorithms using quantum programming frameworks.
- To explore quantum phenomena like superposition, entanglement, and quantum gates through simulation.
- To apply quantum algorithms to solve simple computational problems.
- To gain practical skills in using quantum simulators and cloud-based quantum platforms.

Course Outcomes (COs):

CO1: Understand and simulate basic quantum operations, including quantum states and measurements.
CO2: Implement fundamental quantum gates and construct simple quantum circuits.
CO3: Apply quantum algorithms like Deutsch-Jozsa and Grover's algorithm to solve specific problems.
CO4: Experiment with quantum entanglement and its implications for quantum computing.
CO5: Utilize quantum programming frameworks (e.g., Qiskit) for simulating and executing quantum programs.

Lab Module-Wise Course Content:

Module 1: Introduction to Quantum States and Measurements

Experiment with single-qubit states and their representation (Bloch Sphere).

Simulate quantum measurements and observe probabilistic outcomes.
Implement basic state preparation circuits.

Module 2: Quantum Gates and Circuits
Implement Pauli-X, Y, Z gates and Hadamard gate.
Construct controlled-NOT (CNOT) and other controlled gates.
Build simple quantum circuits (e.g., creating superposition and entanglement).
Module 3: Quantum Algorithms - I (Basic Algorithms)

Implement the Deutsch-Jozsa algorithm for a simple function.
Explore the concepts of quantum parallelism and interference.
Simulate and analyze the results of basic quantum oracles.

Module 4: Quantum Algorithms - II (Search and Optimization)

Implement Grover's search algorithm for an unstructured database.
Experiment with quantum approximate optimization algorithm (QAOA) for simple problems.
Understand the role of amplitude amplification.

Module 5: Quantum Entanglement and Introduction to Advanced Topics

Create and verify entangled states (e.g., Bell states).
Explore quantum teleportation (conceptual understanding and simple simulation).
Introduction to quantum error correction (basic concepts).
Running basic circuits on a quantum simulator or a real quantum device (if access available).

Textbooks & References:
Textbooks

1. "Quantum Computation and Quantum Information," 10th Anniversary Edition, Michael A. Nielsen and Isaac L. Chuang, Cambridge University Press, 2010.
2. "Programming Quantum Computers: Essential Algorithms and Code Samples," O'Reilly Media, 2019 (For practical coding focus).

References

1. IBM Qiskit Documentation: https://qiskit.org/documentation/
2. Microsoft Quantum Development Kit Documentation: https://docs.microsoft.com/en-us/quantum/

CO-PO and CO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	1	2	1	1	1	2	1	2	1	2	1
CO2	3	3	2	2	3	1	1	1	3	1	3	2	2	2
CO3	3	3	2	3	3	1	1	1	3	1	3	2	3	2
CO4	3	3	3	3	3	1	1	1	3	1	3	2	3	2
CO5	3	3	3	3	3	2	2	2	3	2	3	3	3	3

Course Name: Bioinformatics Lab
Course Code: DS792B
Contact Hours (L: T: P): 0:0:3
Credits: 1.5

Prerequisites:

- Basic understanding of Biology/Genetics (recommended)
- Programming in Python (familiarity with data structures and basic scripting)
- Basic statistics concepts

Course Objectives:

- To provide hands-on experience with fundamental bioinformatics tools and databases.
- To perform sequence alignment and analysis using various algorithms.
- To analyze biological data, including DNA, RNA, and protein
- To explore protein structure prediction and visualization techniques.
- To apply computational methods to solve biological problems.

Course Outcomes (COs):
CO1: Understand and utilize common bioinformatics databases and online tools.
CO2: Perform sequence alignment (pairwise and multiple) and interpret the results.
CO3: Analyze DNA and protein sequences for evolutionary relationships and functional motifs.
CO4: Predict and visualize protein structures using computational methods.
CO5: Apply bioinformatics techniques to investigate real-world biological questions.

Lab Module-Wise Course Content:

Module 1: Introduction to Bioinformatics Databases and Tools

Explore NCBI databases (GenBank, PubMed, BLAST).
Use online tools for sequence retrieval and basic information.
Introduction to bioinformatics file formats (FASTA, GenBank).

Module 2: Sequence Alignment (Pairwise and Multiple)

Perform pairwise sequence alignment using BLAST and Needle/Water (local/global alignment).
Interpret alignment scores, E-values, and phylogenetic relationships.
Conduct multiple sequence alignment using Clustal Omega and visualize results.

Module 3: Nucleic Acid Sequence Analysis

Analyze DNA sequences for open reading frames (ORFs).
Identify gene regions, promoters, and terminators.
Perform restriction enzyme mapping.
Basic RNA sequence analysis (e.g., secondary structure prediction).

Module 4: Protein Sequence Analysis and Structure

Analyze protein sequences for domains, motifs, and post-translational modifications.
Use tools for protein property prediction (e.g., molecular weight, isoelectric point).
Introduction to protein structure databases (PDB).
Visualize protein structures using PyMOL or other viewers.

Module 5: Phylogenetic Analysis and Practical Applications

Construct phylogenetic trees using various methods (e.g., neighbor-joining, maximum likelihood).
Interpret phylogenetic trees and evolutionary relationships.
Case studies: Applying bioinformatics tools to disease studies, drug discovery, or agricultural research.
Introduction to scripting for bioinformatics tasks (e.g., parsing FASTA files in Python).

Textbooks & References:

Textbooks

1. "Bioinformatics: Sequence and Genome Analysis," 3rd Edition, David W. Mount, Cold Spring Harbor Laboratory Press, 2017.
2. "Bioinformatics: Principles and Applications," Book by Bibekanand Mallick and Zhumur Ghosh.

References

1. NCBI (National Center for Biotechnology Information) website: https://www.ncbi.nlm.nih.gov/
2. Expasy website: https://www.expasy.org/
3. UniProt website: https://www.uniprot.org/

CO-PO and CO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	1	1	1	1	1	2	1	1	2	1	1
CO2	3	3	2	2	1	1	1	1	2	1	1	3	2	1
CO3	3	3	2	3	1	1	1	1	2	2	1	3	3	2
CO4	3	3	3	3	1	1	1	1	2	2	1	2	3	2
CO5	3	3	3	3	1	2	2	2	3	2	2	3	3	3

Course Name: Parallel and Distributed Computing Lab
Course Code: DS792C
Contact Hours (L:T:P): 0:0:3
Credits: 1.5
Prerequisites:

- Programming in Python/C/C++
- Data Structures and Algorithms
- Basic understanding of Operating Systems

Course Objectives:

- To provide hands-on experience with various parallel programming models and paradigms.
- To implement algorithms for shared-memory and distributed-memory parallel architectures.
- To explore the performance characteristics and challenges of parallel and distributed systems.
- To utilize standard tools and frameworks for developing parallel and distributed applications.
- To apply parallel and distributed computing concepts to solve computationally intensive problems.

Course Outcomes (COs):
CO1: Understand and implement fundamental parallel programming constructs using threads and processes.
CO2: Develop shared-memory parallel programs using OpenMP directives.
CO3: Design and implement distributed-memory parallel programs using MPI.
CO4: Analyze and evaluate the performance and scalability of parallel algorithms.
CO5: Apply parallel and distributed computing techniques to address real-world data-intensive challenges.

Lab Module-Wise Course Content:

Module 1: Introduction to Concurrency and Parallelism

Experiment with multi-threading in Python/C++ (e.g., threading module, pthreads).
Implement basic concurrent tasks (e.g., producer-consumer problem).
Explore process creation and inter-process communication (pipes, shared memory).

Module 2: Shared Memory Parallelism with OpenMP

Set up and compile OpenMP programs (in C/C++).
Implement parallel loops, sections, and tasks using OpenMP.
Practice synchronization primitives (locks, barriers, critical sections).
Analyze speedup and efficiency for simple parallel programs.

Module 3: Distributed Memory Parallelism with MPI

Set up and run MPI programs (in Python/C/C++).
Implement point-to-point communication (send/receive).
Utilize collective communication operations (broadcast, reduce, scatter, gather).
Develop parallel algorithms for distributed arrays.

Module 4: Parallel Algorithms and Performance Analysis

Implement parallel sorting algorithms (e.g., parallel merge sort, odd-even transposition sort).
Implement parallel matrix multiplication.
Measure and analyze performance metrics: speedup, efficiency, scalability.
Identify and mitigate performance bottlenecks in parallel applications.

Module 5: Distributed Data Processing Concepts (Optional/Advanced)

Introduction to distributed file systems (conceptual, no implementation).
Simulate basic concepts of MapReduce paradigm for data processing (e.g., word count).
Explore a simple distributed task scheduling problem.
Hands-on with a simple distributed framework (e.g., basic Dask array operations in Python).

Textbooks & References:

Textbooks

1. "Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers," 2nd Edition, Barry Wilkinson and Michael Allen, Prentice Hall, 2005.
2. "An Introduction to Parallel Programming," Peter S. Pacheco, Morgan Kaufmann, 2011.

References

1. OpenMP Application Program Interface Specifications: https://www.openmp.org/specifications/
2. MPI (Message Passing Interface) Forum: https://www.mpi-forum.org/
3. Python multiprocessing and threading documentation.

CO-PO and CO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	1	2	2	1	1	2	1	1	1	1	1
CO2	3	3	2	2	2	2	1	1	2	1	1	1	2	1
CO3	3	3	2	3	2	2	1	1	2	2	1	2	3	2
CO4	3	3	3	3	3	3	1	1	2	2	1	2	3	2
CO5	3	3	3	3	3	3	2	2	3	2	2	3	3	3

Course Name: Information Theory and Coding Lab
Course Code: DS793A
Contact Hours (L: T: P): 0:0:3
Credits: 1.5

Prerequisites:

- Probability and Random Processes
- Basic programming skills (Python/MATLAB)

Course Objectives:

- To provide hands-on experience with fundamental concepts of information theory.
- To implement various source coding algorithms for data compression.
- To implement and analyze different channel coding techniques for error control.
- To simulate communication channels and evaluate coding performance.
- To gain practical skills in designing and testing coding schemes.

Course Outcomes (COs):
CO1: Implement and evaluate entropy and mutual information for given data sets.
CO2: Develop and test various source coding algorithms (e.g., Huffman, Shannon-Fano-Elias).
CO3: Simulate discrete memoryless channels and analyze their capacity.
CO4: Implement and decode linear block codes, including Hamming codes.
CO5: Design and analyze simple cyclic codes and convolutional codes.

Lab Module-Wise Course Content:
Module 1: Information Measures
Calculate entropy and joint entropy for discrete random variables.
Compute mutual information between two random variables.
Verify basic properties of entropy and mutual information using simulations.
Module 2: Source Coding Techniques
Implement Huffman coding algorithm and evaluate compression ratio.
Implement Shannon-Fano-Elias coding.

Develop a basic Lempel-Ziv (LZ77/LZ78) compression scheme.

Module 3: Channel Simulation and Capacity
Simulate a Binary Symmetric Channel (BSC) with varying error probabilities.
Simulate a Binary Erasure Channel (BEC).
Estimate channel capacity for simple DMCs (e.g., BSC, BEC).

Module 4: Linear Block Codes
Implement encoding and decoding for a simple linear block code (e.g., parity check code).
Implement Hamming code (7,4) for error detection and correction.
Develop a syndrome decoding algorithm for linear block codes.

Module 5: Cyclic and Convolutional Codes
Implement encoding for a simple cyclic code using polynomial division.
Simulate a basic (2, 1, X) convolutional encoder.
Implement the Viterbi algorithm for a simple convolutional code (conceptual/simplified simulation).
Compare the performance of different coding schemes in terms of error rates.

Textbooks & References:
Textbooks

1. "Elements of Information Theory," 2nd Edition, Thomas M. Cover and Joy A. Thomas, Wiley, 2006 (For theoretical background).
2. "Digital Communications," 5th Edition, John G. Proakis and Masoud Salehi, McGraw-Hill Education, 2008 (For coding techniques).

References

1. Online resources and tutorials for Python/MATLAB for signal processing and coding.
2. SciPy and NumPy documentation for numerical operations

CO-PO and CO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	1	2	1	2	2	1	1	2	1	2	1	1	1
CO2	2	2	2	2	2	2	1	1	2	1	2	1	2	2
CO3	2	2	2	3	2	2	1	1	2	2	2	2	3	3
CO4	2	2	3	3	3	3	1	1	2	2	2	2	3	3
CO5	3	3	3	3	3	3	2	2	3	2	3	3	3	3

Course Name: IoT and Data Science Synergy Lab
Course Code: DS793B
Contact Hours (L: T: P): 0:0:3
Credits: 1.5

Prerequisites:

- Basic concepts of IoT and sensors
- Python Programming
- Fundamentals of Data Science and Machine Learning
- Basic database concepts

Course Objectives:

- To provide hands-on experience in collecting, processing, and analyzing data from IoT devices.
- To implement data science techniques on real or simulated IoT datasets.
- To explore various data storage and management solutions for IoT data.
- To develop and evaluate basic data-driven applications for IoT scenarios.
- To understand practical challenges in IoT-Data Science integration.

Course Outcomes (COs):
CO1: Acquire and process sensor data from IoT devices or simulated environments.
CO2: Implement data cleaning, transformation, and feature engineering techniques for IoT data.
CO3: Apply machine learning algorithms for tasks like anomaly detection, prediction, and classification on IoT datasets.
CO4: Utilize basic data visualization tools to present insights from IoT data.
CO5: Develop simple data-driven solutions for real-world IoT use cases.

Lab Module-Wise Course Content:
Module 1: IoT Data Acquisition and Basic Pre-processing
Set up a simulated IoT device (e.g., using Python scripts to generate sensor data).
Read and parse sensor data in various formats (e.g., CSV, JSON).
Implement basic data cleaning: handling missing values, outliers (e.g., using Pandas).

Module 2: IoT Data Storage and Retrieval
Store IoT sensor data into a local time-series database (e.g., InfluxDB or basic file storage).

Retrieve and query historical IoT data for analysis.
Basic exploration of data streams.

Module 3: Feature Engineering and Time-Series Analysis

Extract meaningful features from raw time-series sensor data (e.g., moving averages, standard deviation).
Apply simple time-series analysis techniques (e.g., trend analysis, seasonality detection).
Implement sliding window techniques for feature creation.

Module 4: Machine Learning Applications on IoT Data

Implement a simple anomaly detection algorithm (e.g., Isolation Forest, One-Class SVM) on sensor data.
Develop a predictive model (e.g., linear regression, decision tree) for a sensor output (e.g., temperature prediction).
Classify device states or events based on sensor readings.

Module 5: IoT Data Visualization and Case Study

Visualize IoT data trends, anomalies, and insights using libraries like Matplotlib or Seaborn.
Work on a mini-project applying IoT data science principles to a specific scenario (e.g., smart home energy monitoring, simple health monitoring).
Discuss challenges encountered (e.g., data volume, real-time processing).

Textbooks & References:

Textbooks

1. "INTERNET OF THINGS - A HANDS-ON APPROACH," by Arsheep Bahga, Vijay Madisetti.
2. "Python for Data Analysis," 3rd Edition, Wes McKinney, O'Reilly Media 2022 (For practical data handling).

References

1. Documentation for Python libraries: Pandas, NumPy, Scikit-learn, Matplotlib.
2. Online tutorials for specific IoT platforms or data generation tools.

CO-PO and CO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3

CO1	2	3	2	2	2	2	2	2	2	2	2	3	3	3
CO2	2	3	2	2	2	2	2	2	2	2	2	3	3	3
CO3	2	3	3	2	2	2	3	3	3	2	2	3	3	3
CO4	2	3	3	2	2	2	3	3	3	2	2	3	3	3
CO5	2	3	3	2	3	3	3	3	3	3	3	3	3	3

Course Name: Nature-Inspired Computing in Data Science Lab
Course Code: DS793C
Contact Hours (L: T: P): 0:0:3
Credits: 1.5

Prerequisites:

- Basic concepts of Data Science and Machine Learning
- Programming in Python (familiarity with NumPy, Pandas)
- Understanding of basic optimization concepts

Course Objectives:

- To provide hands-on experience in implementing various nature-inspired computing algorithms.
- To apply nature-inspired algorithms to solve practical data science problems, such as optimization, feature selection, and clustering.
- To analyze and compare the performance of different nature-inspired algorithms on various datasets.
- To gain practical skills in using nature-inspired algorithms for data analysis and model building.
- To evaluate the suitability of nature-inspired approaches for complex real-world data challenges.

Course Outcomes (COs):
CO1: Implement fundamental components of evolutionary algorithms like Genetic Algorithms.
CO2: Develop and apply swarm intelligence algorithms such as Particle Swarm Optimization (PSO) to data problems.
CO3: Utilize nature-inspired algorithms for feature selection and dimensionality reduction in datasets.
CO4: Implement nature-inspired techniques for data clustering and classification.

CO5: Evaluate and compare the performance of different nature-inspired algorithms on benchmark data science problems.

Lab Module-Wise Course Content:

Module 1: Introduction to Evolutionary Algorithms (Genetic Algorithms)

Implement basic genetic algorithm operators (selection, crossover, mutation).
Apply a simple GA to solve a basic optimization problem (e.g., function maximization).
Analyze convergence behavior and parameter tuning.

Module 2: Swarm Intelligence - Particle Swarm Optimization (PSO)

Implement the basic Particle Swarm Optimization (PSO) algorithm.
Apply PSO for continuous function optimization.
Visualize the swarm's movement and best-found solutions.

Module 3: Nature-Inspired Algorithms for Feature Selection

Implement a wrapper-based feature selection method using GA or PSO.
Apply the implemented algorithm to a real-world dataset for feature subset selection.
Evaluate the performance of selected features on a classification task.

Module 4: Nature-Inspired Algorithms for Clustering and Optimization

Implement a clustering algorithm (e.g., K-means) optimized by a nature-inspired algorithm (e.g., GA-Kmeans or PSO-Kmeans).
Apply Simulated Annealing (SA) to solve a combinatorial optimization problem relevant to data (e.g., Traveling Salesperson Problem on data points).

Module 5: Performance Evaluation and Comparative Study

Select and implement another nature-inspired algorithm (e.g., Ant Colony Optimization for pathfinding, Firefly Algorithm for optimization).
Compare the performance (e.g., accuracy, convergence speed) of at least two different nature-inspired algorithms on a common data science problem (e.g., classification, clustering).
Discuss the strengths and weaknesses of each algorithm based on experimental results.

Textbooks & References:

Textbooks

1. "Nature-Inspired Optimization Algorithms," 2nd Edition, Xin-She Yang, Elsevier 2020.
2. "Nature Inspired Computing for Data Science" by Minakhi Rout, Jitendra Kumar Rout, Himansu Das, Springer Nature 2019.

References

1. Documentation for Python libraries: SciPy, NumPy, Scikit-learn, Matplotlib.
2. Online resources and tutorials for specific nature-inspired algorithms.

CO-PO and CO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	1	1	2	3	2	3	2	2	2	2	1	3	3
CO2	2	2	2	2	3	2	3	2	2	2	2	2	3	3
CO3	3	3	3	2	3	3	3	3	3	2	2	3	3	3
CO4	3	3	3	2	3	3	3	3	3	2	2	3	3	3
CO5	3	3	3	2	3	3	3	3	3	3	3	3	3	3

4th Year 8th Semester									
Sl. No.	**Broad Category**	**Category**	**Course Code**	**Course Title**	**Hours per week**				**Credits**
A.	**PRACTICAL**								
1	PRJ	Grand Viva	DS881	Grand Viva	0	0	0	8	**4**
2	ENGG	Ability Enhancement Course	DS882	Internship/Entrepreneurship	0	0	0	8	**4**
		TOTAL CREDIT						**16**	**8**

Course Name: Grand Viva
Course Code: DS881
Contact Hours (L: T: P): 0:0:0
Credits: 4

Students will go through Grand viva

Course Name: Internship/Entrepreneurship
Course Code: DS882
Contact Hours (L: T: P): 0:0:0
Credits: 4

Students will go through Internship/Entrepreneurship