B. Tech. Programme from the Department of Computer Science and Engineering

B. Tech. in Computer Science and Engineering and Minor in Computer Science and Engineering.

Program Learning Objectives:	**Program Learning Outcomes (PLO):**
Program Goal 1: **Fundamental Understanding:** To impart knowledge and proficiency in an advanced level of theoretical and practical aspects in the major fields of Computer Science and Engineering.	**Program Learning Outcome 1:** **PLO-1:** Students will acquire and demonstrate a comprehensive understanding of core concepts in computing principles, data structure, algorithms, programming languages. **Program Learning Outcome 2:** **PLO-2:** Students will be able to understand problems computationally, design efficient algorithms, and implement software solutions.
Program Goal 2: **Basic Training for Research and Industry:** To provide quality training for conducting fundamental and advanced research in Computer Science and Engineering and software development.	**Program Learning Outcome 3:** **PLO-3:** Students will develop the ability to apply the scientific method to computer science problems, including formulating hypotheses, designing experiments, and analyzing results. **Program Learning Outcome 4:** **PLO-4:** Students will demonstrate proficiency in software development, including the use of modern programming environments, operating systems, computer networks, version control, and collaborative development practices.
Program Goal 3: **Skill Enhancement:** To focus on skill enhancement in system development and security.	**Program Learning Outcome 5:** **PLO-5:** Students will be able to design, implement, and manage complex systems, computer architecture, networking, ensuring quality, and security.
Program Goal 4: **Communication Skill:** To develop various communication skills such as reading, listening, speaking, etc. This will help in expressing ideas and views clearly and effectively.	**Program Learning Outcome 6:** **PLO-6:** Students will develop the ability to communicate technical information effectively through written reports, oral presentations, and collaborative projects.
Program Goal 5: **Social Awareness:** To understand societal issues related to computer science and allied areas and develop methods and means to abate and create awareness in society.	**Program Learning Outcome 7:** **PLO-7:** Students will develop an awareness of ethical, social, and environmental issues related to computing, applying responsible practices in their professional activities. **Program Learning Outcome 8:** **PLO-8:** Students will learn to work effectively in teams, demonstrating leadership, collaboration, and project management skills.

Sl. No.	Subject Code	SEMESTER I	L	T	P	C
1.	MA1101	Calculus and Linear Algebra	3	1	0	4.0
2.	CS1101	Foundations of Programming	3	0	3	4.5
3.	PH1101/PH1201	Physics	3	1	3	5.5
4.	CE1101/CE1201	Engineering Graphics	1	0	3	2.5
5.	EE1101/EE1201	Electrical Sciences	3	0	3	4.5
6.	HS1101	English for Professionals	2	0	1	2.5
TOTAL			**15**	**2**	**13**	**23.5**

Sl. No.	Subject Code	SEMESTER II	L	T	P	C
1.	MA1201	Probability Theory and Ordinary Differential Equations	3	1	0	4
2.	CS1201	Data Structure	3	0	3	4.5
3.	CH1201/CH1101	Chemistry	3	1	3	5.5
4.	ME1201/ME1101	Mechanical Fabrication	0	0	3	1.5
5.	ME1202/ME1102	Engineering Mechanics	3	1	0	4
6.	IK1201	Indian Knowledge System (IKS)	3	0	0	3
TOTAL			**15**	**3**	**9**	**22.5**

Sl. No.	Subject Code	SEMESTER III	L	T	P	C
1.	CS2101	Algorithm	3	0	3	4.5
2.	CS2102	Digital Logic and Computer Organization	3	0	3	4.5
3.	CS2103	Artificial Intelligence Concepts	2	0	2	3
4.	CS2104	Discrete Mathematics	3	0	0	3
5.	CS2105	Optimization Techniques	3	0	0	3
6.	HS21XX	HSS Elective I	3	0	0	3
		TOTAL	**17**	**0**	**8**	**21**

Sl. No.	Subject Code	SEMESTER IV	L	T	P	C
1.	CS2201	Formal Language and Automata Theory	3	0	0	3
2.	CS2202	Database and Warehousing	3	0	2	4
3.	CS2203	Artificial Intelligence	3	0	3	4.5
4.	CS2204	IT Workshop	0	2	2	3
5.	CS2206	Computer Architecture	3	0	3	4.5
6.	XX22PQ	IDE - I	3	0	0	3
		TOTAL	**15**	**2**	**10**	**22**

Sl. No.	Subject Code	SEMESTER V	L	T	P	C
1.	CS3101	Operating System	3	0	3	4.5
2.	CS3102	Computer Network	3	0	3	4.5
3.	CS3103	Machine Learning	3	0	3	4.5
4.	CS3104	Compiler	3	0	3	4.5
5.	XX31PQ	IDE - II	3	0	0	3
		TOTAL	**15**	**0**	**12**	**21**

Sl. No.	Subject Code	SEMESTER VI	L	T	P	C
1.	CS3201	Cyber Security	3	0	2	4
2.	CS3202	Deep Learning	3	0	3	4.5
3.	CS3203	Internet of Things	3	0	3	4.5
4.	CS32XX	DE-I (CS ELECTIVES LIST)	3	0	0	3
5.	CS3299	Capstone Project	0	0	6	3
		TOTAL	**12**	**0**	**14**	**19**
		Minor - IV	**3**	**0**	**2**	**4**

Sl. No.	Subject Code	SEMESTER VII	L	T	P	C
1.	CS41XX	DE-II (CS ELECTIVES LIST)	3	0	0	3
2.	CS41XX	DE-III (CS ELECTIVES LIST)	3	0	0	3
3.	XX41PQ	IDE-III	3	0	0	3
4.	HS41XX	HSS Elective II	3	0	0	3
5.	CS4198	Summer Internship*/ Summer Project	0	0	12	3
6.	CS4199	Project – I	0	0	12	6
		TOTAL	**12**	**0**	**24**	**21**

*** For specific cases of internship after 6th Semester, the performance evaluation would be made on joining the VIIth Semester and graded accordingly in the VIIth Semester:**

Note :

a) (i) Summer internship (*) period of at least 60 days' (8 weeks) duration begins in the intervening vacation between semester VI and VII that may be done in industry / R&D / Academic Institutions including IIT Patna. The evaluation would comprise **combined grading based on host supervisor evaluation, project internship report after plagiarism check and seminar presentation at the Department (DAPC to coordinate)** with equal weightage of each of the three components stated herein.

a) (ii) Further, on return from internship, students will be evaluated for internship work through combined grading based on host supervisor evaluation, project internship report after plagiarism check, and presentation evaluation by the parent department with equal weightage of each component.

b) (i) In the VIIth semester, students can opt for a semester long internship on recommendation of the DAPC and approval of the Competent Authority.

b) (ii) On approval of semester long internship, at the maximum two courses (properly mapped/aligned syllabus) at par with institute electives may be opted from NPTEL and / or SWAYAM and the other two more should be done at the institute through course overloading in any other semester (either before or after the internship) and/or during following summer semester.

b) (iii) The candidates opting two courses from NPTEL and / or SWAYAM would be required to appear in the examination at the Institute as scheduled in the Academic Calendar.

Sl. No.	**Subject Code**	**SEMESTER VIII**	**L**	**T**	**P**	**C**
1.	CS42XX	DE-IV (CS Elective List)	3	0	0	3
2.	CS42XX	DE-V (CS Elective List)	3	0	0	3
3.	CS42XX	DE-VI (CS Elective List)	3	0	0	3
4.	CS4299	Project – II	0	0	16	8
		TOTAL	**9**	**0**	**16**	**17**
		GRAND TOTAL (including Semester I & II)	**167**			

ELECTIVE GROUPS

Department Elective - I						
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1.	CS3205	Object-Oriented Programming	3	0	0	3
2.	CS3206	Agile Computing	3	0	0	3
3.	CS3207	Software Engineering	3	0	0	3
4.	CS3208	Bayesian Data Analysis	3	0	0	3
5.	CS3209	Data Mining	3	0	0	3
6.	CS3210	Information Retrieval	3	0	0	3

Department Elective - II						
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1.	CS4101	Pattern Recognition	3	0	0	3
2.	CS4102	Principles of Programming Languages	3	0	0	3
3.	CS4103	Social Networks	3	0	0	3
4.	CS4104	Multimedia System	3	0	0	3
5.	CS4105	Program Analysis and Verification	3	0	0	3

Department Elective - III						
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1.	CS4106	Graph Machine Learning	3	0	0	3
2.	CS4107	Bioinformatics	3	0	0	3
3.	CS4108	Time Series Analysis	3	0	0	3
4.	CS4109	Computational Data Analysis	3	0	0	3
5.	CS4110	Blockchain Technology	3	0	0	3
6.	CS4112	Graph Theory	3	0	0	3

Department Elective - IV						
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1.	CS4201	Multivariate Analysis	3	0	0	3
2.	CS4202	Generative AI	3	0	0	3
3.	CS4203	Statistical Machine Learning	3	0	0	3
4.	CS4204	Text Mining	3	0	0	3
5.	CS4214	Combinatorial Optimization	3	0	0	3

Department Elective - V						
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1.	CS4205	Cloud Computing	3	0	0	3
2.	CS4206	Quantum Computing	3	0	0	3
3.	CS4207	Drone Data Processing	3	0	0	3
4.	CS4208	Edge Computing	3	0	0	3
5.	CS4209	Wireless Networks	3	0	0	3
6.	CS4215	Distributed Computing	3	0	0	3
7.	CS4216	Parallel Computing	3	0	0	3

Department Elective - VI						
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1.	CS4210	Computer Security	3	0	0	3
2.	CS4211	Cryptography	3	0	0	3
3.	CS4212	Big Data Analytics	3	0	0	3
4.	CS4213	Computer Forensics	3	0	0	3

IDE from CSE (Available to student other than Dept. of CSE)						
	Course Code	**Course Name**	**L**	**T**	**P**	**C**
IDE- I	CS2207	Introduction to Data Science	3	0	0	3
IDE –II	CS3106	Computer Graphics	3	0	0	3
IDE -III	CS4113	Data Analysis and Visualization	3	0	0	3

Minor in Computer Science & Engineering						
Minors	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
Minor-I	CS2101	Algorithm	3	0	3	4.5
Minor-II	CS2202	Database and Warehousing	3	0	2	4
Minor-III	CS3101	Operating System	3	0	3	4.5
Minor-IV	CS3201	Cyber Security	3	0	2	4
		Total Credits	**17**			

Sl. No.	Subject Code	SEMESTER I	L	T	P	C
1.	MA1101	Calculus and Linear Algebra	3	1	0	4.0
2.	CS1101	Foundations of Programming	3	0	3	4.5
3.	PH1101/PH1201	Physics	3	1	3	5.5
4.	CE1101/CE1201	Engineering Graphics	1	0	3	2.5
5.	EE1101/EE1201	Electrical Sciences	3	0	3	4.5
6.	HS1101	English for Professionals	2	0	1	2.5
TOTAL			**15**	**2**	**13**	**23.5**

Course Number	**MA1101**
Course Credit (L-T-P-C)	**3-1-0-4**
Course Title	**Calculus and Linear Algebra**
Learning Mode	Lectures and Tutorials
Learning Objectives	To provide the essential knowledge of basic tools of Differential Calculus, Integral Calculus, Vector spaces and Matrix Algebra.
Course Description	This course provides a foundation for Calculus and Linear Algebra. Topics related to properties of single and two variable functions along with their applications will be discussed. In addition fundamentals of linear algebra and matrix theory with applications will also be discussed.
Course Content	**Differential Calculus (12 Lectures)**: Limit and continuity of one variable function (including ε-δ definition). Limit, continuity and differentiability of functions of two variables, Tangent plane and normal, Change of variables, chain rule, Jacobians, Taylor's Theorem for two variables, Extrema of functions of two or more variables, Lagrange's method of undetermined multipliers. **Integral Calculus (10 Lectures)**: Riemann integral for one variable functions, Double and Triple integrals, Change of order of integration. Change of variables, Applications of Multiple integrals such as surface area and volume. **Vector Spaces (12 Lectures)**: Vector spaces (over the field of real numbers), subspaces, spanning set, linear independence, basis and dimension. Linear transformations, range and null space, rank-nullity theorem, matrix of a linear transformation. **Matrix Algebra (8 Lectures)**: Elementary operations and their use in getting the rank, inverse of a matrix and solution of linear simultaneous equations, Orthogonal, symmetric, skew-symmetric, Hermitian, skew-Hermitian, normal and unitary matrices and their elementary properties, Eigenvalues and Eigenvectors of a matrix, Cayley-Hamilton theorem, Diagonalization of a matrix.
Learning Outcome	Students completing this course will be able to: 1. Understand various properties of functions such as limit, continuity and differentiability. 2. Learn about integrations in various dimension and their applications. 3. learn about the concept of basis and dimension of a vector space. 4. define Linear Transformations and compute the domain, range, kernel, rank, and nullity of a linear transformation. 5. compute the inverse of an invertible matrix. 6. solve the system of linear equations. 7. Apply linear algebra concepts to model, solve, and analyze real-world problems.
Assessment Method	Quiz /Assignment/ MSE / ESE

Textbooks:

1. Thomas, G. B., Hass, J., Heil, C. and Weir M. D., "Thomas' Calculus", 14th Ed., Pearson Education, 2018
2. Kreyszig, E., "Advanced Engineering Mathematics", 10th Ed., Wiley India Pvt. Ltd, 2015

Reference Books:

1. Jain, R. K. and Iyenger, S. R. K., "Advanced Engineering Mathematics", 5th Ed., Narosa Publishing House, 2017
2. Axler, S., "Linear Algebra Done Right", 3rd Ed., Springer Nature, 2015
3. Strang, G., "Linear Algebra and Its Applications" 4th Ed., Cengage India Private Limited, 2005

Course Number	**CS1101**
Course Credit	**3-0-3-4.5**
Course Title	**Foundations of Programming**
Learning Mode	Offline
Learning Objectives	• To understand the fundamental concepts of programming • To develop the basic problem-solving skills by designing algorithms and implementing them. • To learn about various data types, control statements, functions, arrays, pointers, and file handling. • To achieve proficiency in debugging and testing a C program
Course Description	This introductory course provides a solid foundation in programming principles and techniques. Designed for students with little to no prior programming experience, it covers fundamental concepts such as variables, data types, control structures, functions, and basic data structures. Students will learn to write, debug, and execute programs using a high-level programming language. Emphasis is placed on developing problem-solving skills, logical thinking, and the ability to write clear and efficient code. By the end of the course, students will be equipped with the essential skills needed to pursue more advanced studies in computer science and software development.
Course Outline	Introduction and Programming basics, Expressions Control and Iterative statements, Functions, Arrays, Recursion vs. Iteration Pointers, 2D-Array with pointers, Structures, String, Dynamic memory allocation, File handling, Contemporary programming languages, and applications **Practical component**: Lab to be conducted on a 3-hour slot weekly. It will be conducted with the theory course so the topics for problems given in the lab are already initiated in the theory class.
Learning Outcome	• Understanding of Basic Syntax and Structure in C language • Proficiency in Data Types, Operators, and Control Structures • Function Implementation and learn to use them appropriately • Efficient Use of Arrays and Strings • Pointer Utilization • Ability to perform dynamic memory allocation and deallocation using malloc (), calloc (), realloc (), and free () functions. • Structured data management with structures and unions • Exposure of file Handling • Learning debugging and error Handling
Assessment Method	Internal (Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Knuth, Donald E. The art of computer programming, volume 4A: combinatorial algorithms, part 1. Pearson Education India, 2011.
- P.J. Deitel and H.M. Deitel, C How To Program, Pearson Education (7th Edition)
- Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice–Hall
- A. Kelley and I. Pohl, A Book on C, Pearson Education (4th Edition)
- K. N. King, C PROGRAMMING A Modern Approach, W. W. Norton & Company

Course Number	**PH1101/PH1201**
Course Credit	**3-1-3-5.5**
Course Title	**Physics**
Learning Mode	Lectures and Tutorials
Learning Objectives	Complies with Program Goals 1 and 2
Course Description	This course deals with fundamentals in Classical mechanics, Waves and Oscillations and Quantum Mechanics. As a prerequisite, the mathematical preliminaries such as coordinate systems, vector calculus etc will be discussed in the beginning.
Course Outline	Orthogonal coordinate systems (Plane polar, Spherical, Cylindrical), concept of generalised coordinates, generalised velocity and phase space for a mechanical system, Introduction to vector operators, Gradient, divergence, curl and Laplacian in different co-ordinate systems. Central force problem and its applications. Rigid body rotation, vector nature of angular velocity, Finding the principal axes, Euler's equations; Gyroscopic motion and its application; Accelerated frame of reference, Fictitious forces. Potential energy and concept of equilibrium, Lennard-Jones and double-well potentials, Small oscillations, Harmonic oscillator, damped and forced oscillations, resonance and its different examples, oscillator states in phase space, coupled oscillations, normal modes, longitudinal and transverse waves, wave equation, plane waves, examples two- and three-dimensional waves. Michelson-Morley experiment, Lorentz transformation, Postulates of special theory of relativity, Time dilation and length contraction, Applications of special theory of relativity.
Learning Outcome	Complies with PLO 1a, 2a, 3a
Assessment Method	Quiz, Assignments and Exams

Suggested Readings:

Textbooks:

1. Engineering Mechanics, M. K. Harbola, 2nd ed., Cengage, 2012
2. D. Kleppner and R. J. Kolenkow, An introduction to Mechanics, Tata McGraw-Hill, New Delhi, 2000.
3. I. G. Main, Oscillations and Waves
4. H. G. Pain, The Physics of Vibrations and Waves, 1968
5. Frank S. Crawford, Berkeley Physics Course Vol 3: Waves and Oscillations, McGraw Hill, 1966.

References:

1. R. P. Feynman, R. B. Leighton and M. Sands, The Feynman Lecture in Physics, Vol I, Narosa Publishing House, New Delhi, 2009.
2. David Morin, Introduction to Classical Mechanics, Cambridge University Press, NY, 2007.
3. P. C. Deshmukh, Foundations of Classical Mechanics, Cambridge University Press, 2019

Course code	**CE1101/CE1201**
Course Credit (L-T-P-C)	**1-0-3-2.5**
Course Title	**Engineering Graphics**
Learning Mode	Lectures and Practical
Learning Objectives	Complies with PLO-1a 1. The course on engineering drawing is designed to introduce the fundamentals of technical drawing as an important form of conveying information. 2. Apply principles of engineering visualization and projection theory to prepare engineering drawings, using conventional and modern drawing tools. 3. Practice drawing orthographic projections, isometric views, and sectional views, of simple and combined solids in different orientations.
Course Description	This course will introduce drawing as a tool to represent a complex three-dimensional object on two-dimensional paper through methods of projections. The course explains the use of different drafting tools and the importance of conventions for uniformity and standardization of the interpretation of the drawings.
Course Outline	Fundamental of engineering drawing, line types, dimensioning, and scales. Conic sections: ellipse, parabola, hyperbola; cycloidal curves. Principle of projection, method of projection, orthographic projection, plane of projection, first angle of projection, Projection of points, lines, planes and solids. Section of solids: Sectional views of simple solids- prism, pyramid, cylinder, cone, sphere; the true shape of the section. Methods of development, development of surfaces. Isometric projections: construction of isometric view of solids and combination of solids from orthographic projections. Introduction to AutoCad and solving isometric problems.
Learning Outcome	After attending this course, the following outcomes are expected: a) The student will understand the basic concepts of engineering drawing. b) The student will be able to use basic drafting tools, drawing instruments, and sheets. c) The student will be able to represent three-dimensional simple and combined solid objects on two-dimensional paper. d) The student will be able to visualize and interpret the orientation of simple and combine solid objects.
Assessment Method	Laboratory Assignments (30%), Mid-semester examination (25%) and End-semester examination (45%).

Suggested Readings:

Textbooks:

1. N.D. Bhatt, Engineering Drawing, Charotar Publishing House.
2. Agrawal & Agrawal, Engineering Drawing, McGraw Hill.
3. Jolhe, Engineering Drawing.

References:

1. Engineering Drawing and Design by David Madsen

Course Number	**EE1101/EE1201**
Course Credit	**3-0-3-4.5**
Course Title	**Electrical Sciences**
Learning Mode	Lectures and Experiments
Learning Objectives	Complies with Program goals 1, 2 and 3
Course Description	The course is designed to meet the requirements of all B. Tech programmes. The course aims at giving an overview of the entire electrical engineering domain from the concepts of circuits, devices, digital systems and magnetic circuits.
Course Outline	Circuit Analysis Techniques, Circuit elements, Simple RL and RC Circuits, Kirchoff's law, Nodal Analysis, Mesh Analysis, Linearity and Superposition, Source Transformations, Thevenin's and Norton's Theorems, Time Domain Response of RC, RL and RLC circuits, Sinusoidal Forcing Function, Phasor Relationship for R, L and C, Impedance and Admittance, Instantaneous power, Real, reactive power and power factor. Semiconductor Diode, Zener Diode, Rectifier Circuits, Clipper, Clamper, UJT, Bipolar Junction Transistors, MOSFET, Transistor Biasing, Transistor Small Signal Analysis, Transistor Amplifier and their types, Operational Amplifiers, Op-amp Equivalent Circuit, Practical Op-amp Circuits, Power Opamp, DC Offset, Constant Gain Multiplier, Voltage Summing, Voltage Buffer, Controlled Sources, Instrumentation Amplifier, Active Filters and Oscillators. Number Systems, Logic Gates, Boolean Theorem, Algebraic Simplification, K-map, Combinatorial Circuits, Encoder, Decoder, Combinatorial Circuit Design, Introduction to Sequential Circuits. Magnetic Circuits, Mutually Coupled Circuits, Transformers, Equivalent Circuit and Performance, Analysis of Three-Phase Circuits, Power measurement in three phase system, Electromechanical Energy Conversion, Introduction to Rotating Machines (DC and AC Machines). Laboratory: Experiments to verify Circuit Theorems; Experiments using diodes and bipolar junction transistor (BJT): design and analysis of half -wave and full-wave rectifiers, clipping and clamping circuits and Zener diode characteristics and its regulators, BJT characteristics (CE, CB and CC) and BJT amplifiers; Experiment on MOSFET characteristics (CS, CG, and CD), parameter extraction and amplifier; Experiments using operational amplifiers (op-amps): summing amplifier, comparator, precision rectifier, Astable and Monostable Multivibrators and oscillators; Experiments using logic gates: combinational circuits such as staircase switch, majority detector, equality detector, multiplexer and demultiplexer; Experiments using flip-flops: sequential circuits such as non-overlapping pulse generator, ripple counter, synchronous counter, pulse counter and numerical display; Power Measurement by two Wattmeter method; Open and Short Circuit Tests of Transformer.
Learning Outcomes	Complies with PLO 1a, 2a and 3a
Assessment Method	Quiz, Assignments and Exams

Texts/References

1. C. K. Alexander, M. N. O. Sadiku, Fundamentals of Electric Circuits, 3rd Edition, McGraw-Hill, 2008.
2. W. H. Hayt and J. E. Kemmerly, Engineering Circuit Analysis, McGraw-Hill, 1993.
3. R. L. Boylestad and L. Nashelsky, Electronic Devices and Circuit Theory, 6th Edition, PHI, 2001.
4. M. M. Mano, M. D. Ciletti, Digital Design, 4th Edition, Pearson Education, 2008.
5. Floyd, Jain, Digital Fundamentals, 8th Edition, Pearson.
6. David V. Kerns, Jr. J. David Irwin, Essentials of Electrical and Computer Engineering, Pearson, 2004.
7. Donald A Neamen, Electronic Circuits; analysis and Design, 3rd Edition, Tata McGraw-Hill Publishing Company Limited.
8. Adel S. Sedra, Kenneth C. Smith, Microelectronic Circuits, 5th Edition, Oxford University Press, 2004.
9. A. E. Fitzgerald, C. Kingsley Jr., S. D. Umans, Electric Machinery, 6th Edition, Tata McGraw-Hill, 2003.
10. D. P. Kothari, I. J. Nagrath, Electric Machines, 3rd Edition, McGraw-Hill, 2004.
11. Del Toro, Vincent. "Principles of electrical engineering." (No Title) (1972).

Course Number	**HS1101**
Course Credit	**L-T-P-W: 2-0-1-2.5**
Course Title	**English for Professionals**
Learning Mode	Offline
Learning Objectives	This course aims to help the students **(a)** attain proficiency in written English through the construction of grammatically correct sentences, utilization of subject-verb agreement principles, mastery of various tenses, and effective deployment of active and passive voice to ensure coherent and impactful written expression; **(b)** enhance oral communication skills by honing public speaking abilities, acquiring strategies to deliver persuasive presentations, and cultivating a polished telephone etiquette, enabling confident and articulate verbal communication; **(c)** foster active listening capabilities by recognizing different types of listening, and applying proven methods and strategies to improve active listening skills; **(d)** strengthen reading skills, including comprehension, interpretation, and critical analysis, to grasp diverse written materials and derive meaning from various types of texts encountered in academic and professional contexts; **(e)** develop adeptness in written communication for business purposes, encompassing the understanding of essential writing elements, mastery of appropriate writing styles thereby enhancing prospects for successful job interviews and subsequent professional endeavors.
Course Description	This academic course on communication skills aims to equip students with fluency in spoken and written English for effective expression in both academic and professional settings. By focusing on essential communication principles and providing practical experiences, students develop clarity, precision, and confidence in their communication. Through interactive discussions and exercises, students enhance critical thinking and adaptability in diverse contexts. Upon completion, students will excel in formal presentations, group discussions, and persuasive writing, enhancing their overall communication proficiency.
Course Outline	**Unit I:** Introduction to professional communication – LSRW - Phonetics and phonology Sounds in English Language – production and articulation – rhythm and intonation – connected speech - Basic Grammar and Advanced Vocabulary Sounds in English Language – production and articulation – rhythm and intonation – connected speech – persuading and negotiating – brevity and clarity in language. Unit II: Characteristics of Technical Communication: Types of communication and forms of communication - Formal and informal communication Verbal and non-Verbal Communication – Communication barriers and remedies Intercultural communication – neutral language Unit III: Comprehension and Composition – summarization, precis writing Business Letter Writing CV/ Resume – E-Communication Unit IV: Statement of Purpose, Writing Project Reports, Writing research proposal, writing abstracts, developing presentations, interviews – combating nervousness Tutorial: Listening Exercises, Speaking Practice (GDs, and Presentations), and Writing Practice Learning Outcome • Attain proficiency in written English, enabling the construction of grammatically correct sentences and coherent written expression through the use of appropriate grammar, tenses, and voice. • Enhance oral communication skills, including public speaking, persuasive presentation, and polished telephone etiquette, fostering confident and articulate verbal expression. • Cultivate active listening abilities, recognizing different listening types, overcoming obstacles, and employing strategies for attentive and effective communication. • Develop proficient written communication skills for business purposes, demonstrating understanding of essential writing elements, appropriate styles, and the creation of reports, notices, agendas, and minutes that effectively convey information.
Assessment Method	Class test + Quiz = 20%; Mid-semester = 25%; Assignment = 15%; End semester = 40%

Suggested Reading

1. Balzotti, Jon. Technical Communication: A Design-Centric Approach. Routledge, 2022.
2. Kaul, Asha, Business Communication. PHI Learning Pvt. Ltd. 2009
3. Laplante, Phillip A. Technical Writing: A Practical Guide for Engineers, Scientists, and Nontechnical Professionals. CRC Press, 2018.
4. Lawson, Celeste, et al. Communication Skills for Business Professionals, Second Edition. CUP, 2019.
5. Sharon Gerson and Steven Gerson. Technical Writing: Process and Product (8th Edition), London: Longman, 2013
6. Rentz, Kathryn, Marie E. Flatley & Paula Lentz. Lesikar's Business Communication Connecting in a Digital world, McGraw-Hill, Irwin.2012
7. Allan & Barbara Pease. The Definitive Book of Body Language, New York, Bantam,2004
8. Jones, Daniel. The Pronunciation of English, New Delhi, Universal Book Stall.2010
9. Savage, Alice. Effective Academic Writing. OUP. 2014
10. Swan and Alter. Oxford English grammar course. OUP. 201

Sl. No.	**Subject Code**	**SEMESTER II**	**L**	**T**	**P**	**C**
1.	MA1201	Probability Theory and Ordinary Differential Equations	3	1	0	4
2.	CS1201	Data Structure	3	0	3	4.5
3.	CH1201/CH1101	Chemistry	3	1	3	5.5
4.	ME1201/ME1101	Mechanical Fabrication	0	0	3	1.5
5.	ME1202/ME1102	Engineering Mechanics	3	1	0	4
6.	IK1201	Indian Knowledge System (IKS)	3	0	0	3
TOTAL			**15**	**3**	**9**	**22.5**

Course Number	**MA1201**
Course Credit (L-T-P-C)	**3-1-0-4**
Course Title	**Probability Theory and Ordinary Differential Equations**
Learning Mode	Lectures and Tutorials
Learning Objectives	To introduce the basic concepts of probability, statistics, and Differential equations.
Course Description	This course aims to cover basic concepts of probability, statistics and ordinary differential equations. In particular, popular distributions, random sampling, various estimators and hypothesis testing will be discussed. Students will also get exposure to the linear ordinary differential equations and their solution techniques.
Course Content	**Probability (12 Lectures)**: Random variables and their probability distributions, Cumulative distribution functions, Expectation and Variance, probability inequalities, Binomial, Poisson, Geometric, negative binomial distributions, Uniform, Exponential, beta, Gamma, Normal and lognormal distributions. **Statistics (10 Lectures)**: Random sampling, sampling distributions, Parameter estimation, Point estimation, unbiased estimators, maximum likelihood estimation, Confidence intervals for normal mean, Simple and composite hypothesis, Type I and Type II errors, Hypothesis testing for normal mean. **Ordinary Differential Equations (20 Lectures)**: First order ordinary differential equations, exactness and integrating factors, Picard's iteration, Ordinary linear differential equations of n-th order, solutions of homogeneous and non-homogeneous equations (Method of variation of parameters). Systems of ordinary differential equations, Power series methods for solutions of ordinary differential equations. Legendre equation and Legendre polynomials, Bessel equation and Bessel functions.
Learning Outcome	Students will get exposure and understanding of: 1. Random variables and their probability distributions 2. Understand popular distributions and their properties 3. Sampling, estimation and hypothesis testing 4. Solution of ordinary differential equations 5. Solution of system of ordinary differential equations 6. Special functions arising as power series solutions of ordinary differential equations
Assessment Method	Quiz /Assignment/ MSE / ESE

Text Books:

1. Hogg, R. V., Mckean, J. and Craig, A. T., “Introduction to Mathematical Statistics”, 8th Ed., Pearson Education India, 2021
2. S.M. Ross “An introduction to Probability Models, Academic Press INC, 11th edition.
3. Miller, I. and Miller, M., “John E. Freund's Mathematical Statistics with Applications”, 8th Ed., Pearson Education India, 2013
4. S. L. Ross, Differential equations, 3rd Edition, Wiley, 1984
5. W. E. Boyce and R. C. Di Prima, Elementary Differential equations and Boundary Value Problems, 7th Edition, Wiley, 2001.

Course Number	**CS1201**
Course Credit	**3-0-3-4.5**
Course Title	**Data Structure**
Learning Mode	Offline
Learning Objectives	• Understand the principles and concepts of data structures and their importance in computer science. • Learn to implement various data structures and understand how different algorithms works. • Develop problem-solving skills by applying appropriate data structures to different computational problems. • Achieving proficiency in designing efficient algorithms.
Course Description	This course provides a comprehensive study of data structures and their applications in computer science. It focuses on the implementation, analysis, and use of various data structures such as arrays, linked lists, stacks, queues, trees, and graphs. Through theoretical concepts and practical programming exercises, this course aims to develop problem-solving and algorithmic thinking skills essential for advanced topics in computer science and software development.
Course Outline	• Introduction to Data Structure, • Time and space requirements, Asymptotic notations • Abstraction and Abstract data types • Linear Data Structure: stack, queue, list, and linked structure • Unfolding the recursion • Tree, Binary Tree, traversal • Search and Sorting, • Graph, traversal, MST, Shortest distance • Balanced Tree **Practical component**: Lab to be conducted on a 3-hour slot weekly. It will be conducted with the theory course so the topics for problems given in the lab are already initiated in the theory class.
Learning Outcome	• Understand Data Structure Fundamentals • Implement Basic Data Structures using a programming language • Analyse and Apply Algorithms • Design and Analyse Tree Structures • Understand the usage of graph and its related algorithms • Design and Implement Sorting and Searching Algorithms • Debug and Optimize Code
Assessment Method	Internal (Quiz/Assignment/Project), Mid-Term, End-Term

<u>Suggested Reading:</u>

- Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Data Structures and Algorithms, Published by Addison-Wesley
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein., Introduction to Algorithms,
- Mark Allen Weiss, Data Structures and Algorithm Analysis in Java
- Robert Sedgewick and Kevin Wayne, Algorithms
- Narasimha Karumanchi, Data Structures and Algorithms Made Easy

Course Number	**CH1201/CH1101**
Course Credit	**3-1-3-5.5**
Course Title	**Chemistry**
Learning Mode	Offline
Learning Objectives	The course aims to lay a foundation for all three branches of chemistry, viz. Organic, Inorganic, and Physical Chemistry. The course aims to nurture knowledge to appreciate the interface of chemistry with other science and Engineering branches by combining theoretical concepts and experimental studies.
Course Description	This course introduces basic organic chemistry, inorganic chemistry and Physical chemistry to understand fundamental laws that governs various reactions, reaction rates, equilibrium, and their applications in daily life through relevant experimentation.
Course Outline	**Module 1:** Thermodynamics: The fundamental definition and concept, the zeroth and first law. Work, heat, energy and enthalpies. Second law: entropy, free energy and chemical potential. Change of Phase. Third law. Chemical equilibrium. Conductance of solutions, Kohlrausch's law-ionic mobilities, Basic Electrochemistry. **Module 2:** Coordination chemistry: Crystal field theory and consequences color, magnetism, J.T distortion. Bioinorganic chemistry: Trace elements in biology, heme and non-heme oxygen carriers, haemoglobin and myoglobin; Organometallic chemistry. **Module 3:** Stereo and regio-chemistry of organic compounds, conformational analysis and conformers, Molecules devoid of point chirality (allenes and biphenyls); Significance of chirality in living systems, organic photochemistry, Modern techniques in structural elucidation of compounds (UV–Vis, IR, NMR). **Module 4 (Lab Component):** Experiments based on redox and complexometric titrations; synthesis and characterization of inorganic complexes and nanomaterials; synthesis and characterization of organic compounds; experiments based on chromatography; experiments based on pH and conductivity measurement; experiment related to chemical kinetics and spectroscopy.
Learning Outcome	Students will be able to 1**.** identify organic and inorganic molecules and relate them to daily life applications through experiments. 2. understand important hypothesis, laws and their derivations to intercept physical phenomenon of chemical reactions and apply them in hands-on experiments. 3. understand the importance of organic and inorganic molecules in our body and environment. 4. know important analytical techniques to intercept chemical entity. 5. approach organic and inorganic synthesis as a skillset for drug manufacturing, calculate limiting reagents and yields, use various analytical tools to characterize organic compounds, interpret and ascertain data related to Physical chemistry aspects and know laboratory safety measures, risk factors and scientific report writing skills.
Assessment Method	**Theory**: 20% Quiz and assignment, 30% Mid sem and 50% End semester exams for theory part (4 credits). **Lab**: 60% lab report, lab performance and assignment, 20% End semester exam for practical part, 20% viva/quiz (1.5 credits). **Overall Weightage**: Theory (70%), Lab (30%).

Suggested Reading:

Text books:

1. Vogel's Qualitative Inorganic Analysis, G. Svehla, 7th Edition, Revised, Prentice Hall, 1996.
2. A. J. Elias, S. S. Manoharan and H. Raj, "Experiments in General Chemistry", Universities Press (India) Pvt. Ltd., 1997.
3. A. J. Elias, A Collection of Interesting General Chemistry Experiments, revised edition, Universities Press (India) Pvt. Ltd., 2007.
4. F. Albert Cotton, G. Wilkinson, C. A. Murillo, M. Bochmann, Advanced Inorganic Chemistry - 6th Edition New Delhi: Wiley India, 2008.
5. K. Mukkanti, Practical Engineering Chemistry, B.S. Publications, Hyderabad, 2009.
6. Shriver and Atkins inorganic chemistry / Peter Atkins, Tina Overton, Jonathan Rourke, Mark Weller, Fraser Armstrong-5th Edition – Oxford: UOP. 2012.
7. Atkins' Physical Chemistry, Peter Atkins, Julio de Paula, James Keeler, Oxford University Press, 11th Edition

2017.

8. K. L. Kapoor, A Textbook of Physical Chemistry, Vol: 1, 2 (6th Edition, 2019), Vol: 3 (5th Edition, 2020) MaGraw Hill.
9. G. R. Chatwal, S. K. Anand, Instrumental Methods of Chemical Analysis, 5th Edition, Himalaya Publications, 2023.

	PLO-1	PLO-2	PLO-3	PLO-4	PLO-5	PLO-6	PLO-7	PLO-8
CLO-1	X	X	X	X	X	X	X	X
CLO-2	X	X		X	X			
CLO-3	X	X	X	X		X	X	
CLO-4	X	X		X	X	X	X	X
CLO-5			X	X	X			X

Course Number	**ME1201/ME1101**
Course Credit (L-T-P-C)	**0-0-3-1.5**
Course Title	**Mechanical Fabrication**
Learning Mode	Fabrication work – hands on fabrication work in Workshop
Learning Objectives	Complies with PLOs 3-4. • This course aims to develop the concepts and skills of various mechanical fabrication methods. • Fabrication of metallic and non-metallic components, fabrication using bulk and sheet metals, subtractive and additive manufacturing methods, and assemble the parts
Course Description	This course is designed to fulfil the need of hand on experience about various approaches (conventional and CNC, subtractive and additive) of mechanical fabrication approaches. Prerequisite: NIL
Course Outline	The jobs for various shops should be planned such that they are the parts of an assembled item. The student groups will fabricate different parts in various shops which will involve some amount of their creativeness/input particularly in design and/or planning. Various components as required for the assembled part can be made using the following shops: **Sheet Metal Working:** Development, sheet cutting and fabrication of designated job using sheet metal (ferrous/nonferrous); Joining of required portions by soldering, in case part is desired to be made leak proof. **Pattern Making and Foundry:** Making of suitable pattern (wood); making of sand mould, melting of non-ferrous metal/alloy (Al or Al alloys), pouring, solidification. Observation/identification of various defects appeared on the component. **Joining:** Butt/lap/corner joint job fabrication as required of low carbon steel plates; weld quality inspection by dye-penetration test (non-destructive testing approach)of the component made. Demonstration of semi-automatic Gas Metal Arc welding (GMAW). **Conventional machining:** Operations on lathe and vertical milling to fabricate the required component. The fabrication of the component should cover various lathe operations like straight turning, facing, thread cutting, parting off etc., and operations using indexing mechanism on vertical milling. **CNC centre:** Fundamentals of CNC programming using G and M code; setting and operations of job using CNC lathe or milling, tool reference, work reference, tool offset, tool radius compensation to fabricate the component with a designed profile on Al/Al-alloy plate. **3D printing (Fused Filament Fabrication): (2 weeks)** Create the model, select appropriate slicing and path for fabrication of a 3D job by layer deposition (additive manufacturing approach) using polymeric material. Demonstration on pattern fabrication using 3D printing.
Learning Outcome	• This course would enable the students to develop the concept of design, fabrication (subtractive and additive) for various engineering applications**.** Fabrication of components and assemble them. • The practical skill and hands on experience for various fabrication methods from bulk, sheet metal using conventional as well as CNC machines.
Assessment Method	Fabrication of components in each of the shops required for assembly of the given part; submission of reports for each shop, and quiz assessment.

Text and Reference books:

1. Hajra Choudhury, HazraChoudhary and Nirjhar Roy, 2007, Elements of Workshop Technology, vol. I,Mediapromoters and Publishers Pvt. Ltd.
2. W A J Chapman, Workshop Technology, 1998, Part -1, 1st South Asian Edition, Viva Book Pvt Ltd.
3. P.N. Rao, 2009, Manufacturing Technology, Vol.1, 3rd Ed., Tata McGraw Hill Publishing Company.
4. M.Adithan, B.S. Pabla, 2012, CNC machines, New Age International Publishers

Course Number	**ME1202/ ME1102**
Course Number	**Engineering Mechanics**
Course Credit (L-T-P-C)	**3-1-0-4**
Pre-requisites	Nil
Semester	Spring
Learning Mode	Lectures
Learning Objectives	Complies with PLOs 1, 4 • The objective of this first course in mechanics is to enable engineering students to analyze basic mechanics problems and apply vector-based approach to solve them.
Course Outline	1. **Rigid body statics**: Equivalent force system. Equations of equilibrium, Free body diagram, Reaction, Static indeterminacy. 2. **Structures**: 2D truss, Method of joints, Method of section. Beam, Frame, types of loading and supports, axial force, Bending moment, Shear force and Torque Diagrams for a member. 3. **Friction**: Dry friction (static and kinetic), wedge friction, disk friction (thrust bearing), belt friction, square threaded screw, journal bearings, Wheel friction, Rolling resistance. 4. **Centroid and Moment of Inertia** 5. **Introduction to stress and strain**: Definition of Stress, Normal and shear Stress. Relation between stress and strain, Cauchy formula. 6. **Stress in an axially loaded member and stress due to torsion in axisymmetric section**
Learning Outcomes:	Following learning outcomes are expected after going through this course. • Learn and apply general mathematical and computer skills to solve basic mechanics problems. • Apply the vector-based approach to solve mechanics problems.
Assessment Method	Mid semester examination, End semester examination, Class test/Quiz, Tutorials

Reference Books

1. H. Shames, Engineering Mechanics: Statics and dynamics, 4th Ed, PHI, 2002.
2. F. P. Beer and E. R. Johnston, Vector Mechanics for Engineers, Vol I - Statics, 3rd Ed, Tata McGraw Hill, 2000.
3. J. L. Meriam and L. G. Kraige, Engineering Mechanics, Vol I - Statics, 5th Ed, John Wiley, 2002.
4. E.P. Popov, Engineering Mechanics of Solids, 2nd Ed, PHI, 1998.
5. F. P. Beer and E. R. Johnston, J.T. Dewolf, and D.F. Mazurek, Mechanics of Materials, 6th Ed, McGraw Hill Education (India) Pvt. Ltd., 2012.

Sl. No.	**Subject Code**	**SEMESTER III**	**L**	**T**	**P**	**C**
1.	CS2101	Algorithm	3	0	3	4.5
2.	CS2102	Digital Logic and Computer Organization	3	0	3	4.5
3.	CS2103	Artificial Intelligence Concepts	2	0	2	3
4.	CS2104	Discrete Mathematics	3	0	0	3
5.	CS2105	Optimization Techniques	3	0	0	3
6.	HS21XX	HSS Elective - I	3	0	0	3
TOTAL			**17**	**0**	**8**	**21**

Course Number	**CS2101**
Course Credit (L-T-P-C)	**3-0-3-4.5**
Course Title	**Algorithm**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) to understand and explain fundamental concepts of computational complexity, including time and space complexity, and analyses the efficiency of algorithms; (b) to apply various algorithm design paradigms such as divide-and-conquer, dynamic programming, greedy algorithms, and backtracking to solve computational problems; (c) to develop and implement common algorithms for tasks such as sorting, searching, and graph traversal, and utilize well-known algorithms like Dijkstra's and Kruskal's; (d) to utilize fundamental data structures, including arrays, linked lists, stacks, queues, trees, and graphs, selecting and implementing the most appropriate one for specific problems; and (e) to evaluate the performance and scalability of algorithms and data structures, conducting empirical analysis to understand their practical performance, and enhancing problem-solving skills through theoretical knowledge application in practical scenarios.
Course Description	The course introduces the basics of computational complexity analysis and various algorithm design paradigms. The goal is to provide students with solid foundations to deal with a wide variety of computational problems, and to provide a thorough knowledge of the most common algorithms and data structures.
Course Outline	**Unit I** Role of algorithms in computing and elementary data structures. **Unit II** Analysis framework: Asymptotic notations, Analysis & Master Theorem Unfolding of recursion: review of sorting and searching algorithms, Huffman Encoding, String matching, hashing, Trees, Subset sum **Unit III** Algorithm design paradigm: • Brute force algorithms- Exhaustive search • Greedy algorithms • Divide and conquer algorithms, Branch-and-bound • Backtracking • Dynamic programming: Matrix Chain Multiplication, 0/1 Knapsack problem **Unit IV** Graph based algorithm: MST, Shortest distance, colouring, Vertex cover, TSP **Unit V** Reducibility: P, NP, NP complete, and NP hard **Unit VI** Elements of Randomized and approximation Algorithms **Practical component**: Lab to be conducted on a 3-hour slot weekly. It will be conducted with the theory course so the topics for problems given in the lab are already initiated in the theory class.
Learning Outcome	• Describe how efficiency affects the practical usage of algorithms and data structures. • Identify different algorithmic techniques for running programs at scale. • Construct programs that apply computational concepts as a tool in other domains. • Discuss how computer science interacts with and affects the world.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- T. H. Carmen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to Algorithms, MIT Press, 2001.
- A. Aho, J. E. HopcroŌ and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974.
- M. T. Goodrich and R. Tamassia, Algorithm Design: Foundations, Analysis and Internet Examples, John Wiley & Sons, 2001

Course Number	**CS2102**
Course Credit **(L-T-P-C)**	**3-0-3-4.5**
Course Title	**Digital Logic and Computer Organization**
Learning Mode	Offline
Learning Objectives	This course targets to cover the different number systems, designing of combinational and sequential logic circuits. This course will also expose students to the basic architecture of processing, memory and i/o organization in a computer system.
Course Description	The course covers foundation of digital logic and Computer organization that including number systems, Boolean algebra, optimizing logic gates. Besides this it covers designing of different combinational and sequential circuits, computer organization
Course Outline	Number System and Codes; Combinational logic circuits: Sequential logic circuits; Finite State machines. Basic computer organization and design, Operational concepts, Instruction codes, Computer Registers, Computer Instructions Familiarization with assembly language programming; Execution of a complete instruction. Memory organization: concept of hierarchical memory organization I/O devices – Programmed Input/output -Interrupts – Direct Memory Access – Buses, I/O devices and processors. **Practical component:** Lab to be conducted on a 3-hour slot weekly. It will be conducted with the theory course so the topics for problems given in the lab are already initiated in the theory class.
Learning Outcome	The student will be able to: • Demonstrate an understanding of how data is represented within a computer system. • Appreciate understanding of the basic blocks, key terminology in digital logic and Computer organization • Demonstrate classic components of a computational system (i.e. input, output, memory, data path, control) and understanding their functionality.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Text Books:

- Mano, M. Morris. *Digital logic and computer design*. Pearson Education India, 2017.
- Harris, David, and Sarah Harris. *Digital design and computer architecture*. Morgan Kaufmann, 2010.
- M. Moris Mano, "Computer Systems Architecture", 4th Edition, Pearson/PHI,
- Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization", 5th Edition, McGraw Hill.
- William Stallings, "Computer Organization and Architecture", 6th Edition, Pearson/PHI

Course Number	**CS2103**
Course Credit (L-T-P-C)	**2-0-2-3**
Course Title	**Artificial Intelligence Concepts**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) grasp the fundamental principles and subfields of Artificial Intelligence (AI) and Data Science.(b) Gain expertise in the stages of Data Science from data collection to model evaluation.(c) proficiency in applying supervised and unsupervised learning algorithms.(d) introduced to Deep Learning architectures and their applications.
Course Description	This course offers a comprehensive exploration of foundational principles and advanced techniques in Artificial Intelligence (AI), Data Science, Machine Learning (ML), and Deep Learning (DL). Students will delve into the ethical implications, applications, and future trends of AI, understanding its societal impacts and responsible deployment. The curriculum covers the evolution and stages of Data Science, emphasizing mastery of data collection, pre-processing, exploratory analytics, and rigorous model development and evaluation across various domains. In Machine Learning, students will gain proficiency in supervised and unsupervised learning algorithms, feature selection, dimensionality reduction, and a variety of classification and clustering techniques. Deep Learning concepts will be introduced, focusing on neural networks, Convolutional Neural Networks (CNNs) for image analysis, Recurrent Neural Networks (RNNs) for sequential data processing, attention mechanisms, and training Generative Adversarial Networks (GANs). Through theoretical lectures, practical exercises, and hands-on projects, students will acquire the skills necessary to apply these technologies effectively in solving real-world problems and advancing their careers in AI and Data Science.
Course Outline	Historical evolution of AI, Conceptualization of AI, related terms, and subfields with applications. **Practical component**: Lab to be conducted on a 2-hour slot weekly. It will be conducted with the theory course so the topics for problems given in the lab are already initiated in the theory class.
Learning Outcome	• Understand the basic concept of AI. • Analysis of Data using Data science and data Analytics. • Explore state-of-the-art techniques and applications in machine learning • Compare and contrast various multiple deep learning architectures
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Text Books:

1. Tom M. Mitchell, *2017. Machine Learning*.
2. Andrew-ng. Lecture Series – Deep Learning.ai . (Stanford)
3. Relevant research articles.

Reference Books:

1. Grus, J., 2019. *Data science from scratch: first principles with python*

Course Number	**CS2104**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Discrete Mathematics**
Learning Mode	Offline
Learning Objectives	The objective of the course is to introduce the fundamental concepts in discrete mathematics with emphasis on their applications to computer science.
Course Description	This course covers Fundamentals of logic (the laws of logic, rules of inferences, quantifiers, proofs of theorems), Fundamental principles of counting (permutations, combinations), set theory, relations and functions, graphs, shortest path and minimal spanning trees algorithms. Monoids and Groups.
Course Outline	• Logic and proofs • Elementary set theory • Relations and functions • Recurrence relations • Counting & Combinatorics • Induction and Recursion • Modular arithmetic • Graph theory • Elementary probability theory
Learning Outcomes	• Mathematical formalism of complex computer science problem and identifying their effective solutions. • Improving critical thinking, and recognize valid, logical, mathematical arguments and construct valid arguments/proofs. • Understanding the mathematical foundation behind cryptographic solutions in cryptology and others.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Discrete Mathematics and its Applications - Kenneth H. Rosen 7th Edition -Tata McGraw Hill, 2007
- Elements of Discrete Mathematics, C. L Liu, McGraw-Hill Inc, 1985. Applied Combinatorics, Alan Tucker, 2007.
- Concrete Mathematics, Ronald Graham, Donald Knuth, and Oren Patashnik, 2nd Edition - Pearson Education Publishers - 1996.
- Combinatorics: Topics, Techniques, Algorithms by Peter J. Cameron, Cambridge University Press, 1994 (reprinted 1996).

Course Number	**CS2105**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Optimization Techniques**
Learning Mode	Offline
Learning Objectives	To gain a thorough understanding principles of linear programming including problem formulation, geometric interpretations, and graphical solutions. To explore advanced methods such as the Simplex algorithm, Big M method, and Revised Simplex method for optimizing linear programming problems. To understand duality theory and sensitivity analysis in linear programming, and apply them to real-world scenarios like transportation and assignment problems. To learn integer programming techniques like Branch and Bound and the Gomory cutting plane method for solving integer and mixed integer problems. To understand game theory concepts such as saddle points, matrix games, and strategies, and apply optimization methods to solve game-theoretic problems effectively.
Course Description	This course provides an exploration of essential methods for solving complex problems across various domains, including operations research, engineering, economics, and artificial intelligence. Beginning with foundational concepts in linear programming, students will delve into problem formulation, geometric interpretations, and graphical solutions, progressing to advanced techniques such as the Simplex algorithm, Big M method, and Revised Simplex method. Duality theory in linear programming is extensively covered, alongside integer programming techniques like Branch and Bound and the Gomory cutting plane method for both integer and mixed integer problems. The course also explores game theory applications, focusing on matrix games and two-person zero-sum games, utilizing graphical and simplex methods to derive optimal solutions. Additionally, students will gain insights into optimization techniques tailored for artificial intelligence and machine learning applications, preparing them to tackle real-world optimization challenges effectively.
Course Outline	Linear programming problem (LLP): Introduction and problem formulation, Concepts from Geometry, Geometrical aspects of LPP, Graphical solutions, Linear programming in standard form, Simplex, Big M and Two Phase Methods, Revised simplex method, Special cases of LPP. Duality theory: Dual simplex method, Sensitivity analysis of LP problems, Transportation, Assignment, and Traveling Salesman problems. Integer programming problems: Branch and bound method, Gomory cutting plane method for all integers and for mixed integer LPP. Theory of games: Saddle point, Linear programming formulation of matrix games, Two-person zero-sum games with and without saddle-points, Pure and mixed strategies, Graphical method of solution of a game, Solution of a game by simplex method. Basics of optimization techniques for artificial intelligence and machine learning
Learning Outcome	Upon successful completion of this course, students will: • Demonstrate proficiency in formulating and solving linear programming problems using advanced methods like the Simplex algorithm and its variants. • Apply duality theory and sensitivity analysis to analyze and optimize solutions in linear programming applications, including transportation and assignment problems. • Utilize integer programming techniques, such as Branch and Bound and Gomory cutting plane methods, to solve integer and mixed integer linear programming problems effectively. • Apply game theory concepts to analyze and solve matrix games using linear programming formulations, employing graphical and simplex methods for optimal strategy determination. • Apply optimization techniques relevant to artificial intelligence and machine learning applications, demonstrating the ability to optimize models
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Hamdy A. Taha, Operations Research: An Introduction, 10th edition, PHI, New Delhi (2019).
- F.S. Hillier, G.J. Lieberman, Introduction to Operations Research, 10th edition, McGraw Hill (2017).
- Ravindran, D.T. Phillips, J.J. Solberg, Operations Research, John Wiley and Sons, New York (2005).
- M.S. Bazaraa, J.J. Jarvis and H.D. Sherali, Linear Programming and Network Flows, 3rd Edition, Wiley (2004).
- D.G. Luenberger, Linear and Nonlinear Programming, 2nd Edition, Kluwer (2003).

.

Sl. No.	Subject Code	SEMESTER IV	L	T	P	C
1.	CS2201	Formal Language and Automata Theory	3	0	0	3
2.	CS2202	Database and Warehousing	3	0	2	4
3.	CS2203	Artificial Intelligence	3	0	3	4.5
4.	CS2204	IT Workshop	0	2	2	3
5.	CS2206	Computer Architecture	3	0	3	4.5
6.	XX22PQ	IDE-I	3	0	0	3
TOTAL			**15**	**2**	**10**	**22**

Course Number	**CS2201**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Formal Language and Automata Theory**
Learning Mode	Offline
Learning Objectives	This course will introduce Learners about the basic mathematical models of computation, problems that can be solved by computers and problems that are computationally hard. It also introduces basic computation models, their properties and the necessary mathematical techniques to prove more advanced attributes of these models. The learners will be able to express computer science problems as mathematical statements and formulate proofs.
Course Description	This course is designed to cover computability and computational complexity theory. Topics include regular and context-free languages, decidable and undecidable problems, reducibility, time and space measures on computation.
Course Outline	Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages. Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata. Context-free languages and pushdown automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs. Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. Turing machines: The basic model for Turing machines (TM), Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators. Undecidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice's theorem, undecidable problems about languages; Complexity theory: time and space complexity, Classes P, NP, NP-complete.
Learning Outcomes	The student will be able to: • Gain proficiency with mathematical tools and formal methods • Understand various mathematical models of computation and formal languages • Understand Turing machines, decidable languages, and undecidable languages • Design and analyze Turing machines, their capabilities and limitations • Understand the basics of complexity theory, complexity classes and possible unsolved problems in theoretical computer science
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

1. J. E. Hopcroft, R. Motwani and J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Pearson Education India (3rd edition).

2. K. L. P. Mishra, N. Chandrasekaran, Theory of Computer Science: Automata, Languages and Computation, PHI Learning Pvt. Ltd. (3rd edition).

3. D. I. A. Cohen, Introduction to Computer Theory, John Wiley & Sons, 1997.

4. J. C. Martin, Introduction to Languages and the Theory of Computation, Tata McGraw-Hill (3rd Ed.).

5. H. R. Lewis and C. H. Papadimitriou, Elements of the Theory of Computation, Prentice Hall, 1997.

6. Garey, D.S., Johnson, G., Computers and Intractability: A Guide to the Theory of NP- Completeness, Freeman, New York, 1979

7. M. Sipser, Introduction to the Theory of Computation, Thomson, 2004

Course Number	**CS2202**
Course Credit (L-T-P-C)	**3-0-2-4**
Course Title	**Database and Warehousing**
Learning Mode	Offline
Learning Objectives	• Understand the fundamental principles of database systems and data warehousing. • Learn to design, implement, and manage databases using relational database management systems (RDBMS). • Explore the concepts and techniques of data warehousing and data mining. • Develop skills in SQL for querying and managing databases. • Analyze and optimize database performance and ensure data integrity and security.
Course Description	This course provides an in-depth exploration of database systems and data warehousing, covering essential concepts, technologies, and techniques. Students will learn about the design and implementation of relational databases, including data modeling, normalization, and SQL. The course will also introduce data warehousing concepts, focusing on data extraction, transformation, and loading (ETL), as well as data mining techniques. Through practical exercises and projects, students will gain hands-on experience in working with databases and data warehouses, preparing them for real-world applications.
Course Outline	1. Introduction to Databases, Overview of database systems, Types of databases and database models, Database architecture and components 2. Data Modeling, Entity-Relationship (ER) modeling, Relational model and schema design, Normalization and denormalization 3. Structured Query Language (SQL), Basic SQL queries (SELECT, INSERT, UPDATE, DELETE), Advanced SQL (joins, subqueries, indexing) , SQL functions and stored procedures 4. Database Design and Implementation, Database design principles, Creating and managing databases using RDBMS, Data integrity and constraints 5. Database Management and Administration, Database backup and recovery, User management and security, Performance tuning and optimization 6. Introduction to Data Warehousing, Concepts and architecture of data warehousing, Data warehousing vs. databases, Data modeling for data warehousing 7. ETL Processes, Data extraction, transformation, and loading (ETL), ETL tools and techniques, Data cleaning and integration 8. Data Mining and Analytics, Introduction to data mining, Data mining techniques and algorithms, Applications of data mining 9. Advanced Topics in Data Warehousing, Big data and data warehousing, Cloud-based data warehousing solutions, Data governance and data quality management **Practical component:** Lab to be conducted on a 2-hour slot weekly. It will be conducted with the theory course so the topics for problems given in the lab are already initiated in the theory class.
Learning Outcome	The student will be able to: • Demonstrate a thorough understanding of database and data warehousing principles. • Design, implement, and manage relational databases using RDBMS. • Write efficient SQL queries for data manipulation and retrieval. • Implement data warehousing solutions, including ETL processes and data mining techniques. • Analyze and optimize the performance of databases and data warehouses, ensuring data integrity and security.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Textbooks

1. "Database System Concepts" (7th Edition) by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan
2. "Fundamentals of Database Systems" (7th Edition) by Ramez Elmasri and Shamkant B. Navathe
3. "Data Warehousing: The Ultimate Guide to Building a Data Warehouse for Business Intelligence" (1st Edition) by Erik Thomsen
4. "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" (3rd Edition) by Ralph Kimball and Margy Ross

5. "SQL: The Complete Reference" (3rd Edition) by James R. Groff and Paul N. Weinberg

Course Number	**CS2203**
Course Credit (L-T-P-C)	**3-0-3-4.5**
Course Title	**Artificial Intelligence**
Learning Mode	Offline
Learning Objectives	• To understand the core concepts and principles of Artificial Intelligence and intelligent agents. • To learn and apply uninformed and informed search strategies to solve complex problems. • To formulate and solve constraint satisfaction problems and engage in adversarial search. • To represent knowledge using propositional and first-order logic and perform inference and planning. • To utilize various learning techniques and understand their applications in different AI domains.
Course Description	This course provides a comprehensive introduction to the fundamental concepts and techniques of Artificial Intelligence (AI). Students will learn about the design and implementation of intelligent agents, various search strategies, constraint satisfaction problems, knowledge representation, and reasoning. Additionally, the course covers learning techniques and their practical applications, preparing students to apply AI principles in real-world scenarios. The lab component allows students to implement these concepts, reinforcing theoretical knowledge through hands-on experience.
Course Outline	Introduction: Definition and scope of Artificial Intelligence, background and evolution, intelligent agents and environment Problem Solving: Solving problems by searching, uninformed and informed search Uninformed search: Breadth-first search (BFS), Depth-first search (DFS), Uniform-cost search (UCS) Informed search: Heuristic function design and evaluation, A* search Local search: Hill climbing Adversarial search: Min-max, alpha-beta pruning Constraint Satisfaction Problem (CSP): definition and examples of CSPs Knowledge Representation and Reasoning: Propositional Logic, First Order Logic Introduction to Learning Techniques: Bayesian, decision tree, etc. Some applications of AI **Practical component:** Lab to be conducted on a 3-hour slot weekly. It will be conducted with the theory course so the topics for problems given in the lab are already initiated in the theory class.
Learning Outcome	By the end of this course, students will be able to: • Understand the core concepts and principles of Artificial Intelligence and intelligent agents. • Apply uninformed and informed search strategies to solve complex problems. • Formulate and solve constraint satisfaction problems and engage in adversarial search. • Represent knowledge using propositional and first-order logic and perform inference and planning. • Utilize various learning techniques and understand AI applications in different domains.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

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

Course Number	**CS2204**
Course Credit (L-T-P-C)	**0-2-2-3**
Course Title	**IT Workshop**
Learning Mode	Offline
Learning Objectives	• To understand the basics of shell scripting and its applications in automating tasks. • To learn the fundamentals of Android programming and app development. • To gain practical experience in writing scripts and developing Android applications. • To develop problem-solving skills through scripting and programming exercises. • To explore the integration of shell scripts within Android environments.
Course Description	This undergraduate course provides a foundational understanding of both shell scripting and Android programming. Students will start by learning the essential concepts of shell scripting, including syntax, commands, and script writing techniques to automate tasks in Unix/Linux environments. The course then transitions into Android programming, covering the basics of Java/Kotlin, Android Studio, and app development. By combining these two areas, the course aims to equip students with a versatile skill set that is highly valuable in the tech industry. Through a series of lectures, hands-on labs, and projects, students will gain the knowledge and experience needed to create efficient scripts and functional Android applications.
Course Outline	1. Introduction to Shell Scripting: Overview of Unix/Linux systems, Basic shell commands and utilities, Writing and executing simple shell scripts 2. Advanced Shell Scripting: Control structures (loops, conditionals) , Functions and arrays in shell scripting , Script debugging and error handling 3. Practical Shell Scripting: Automating tasks and processes, File manipulation and text processing, Networking and system administration scripts 4. Introduction to Android Programming: Overview of Android OS and development environment, Setting up Android Studio and creating a basic app, Introduction to Java/Kotlin for Android development 5. Building Android Applications: User interface design and XML layouts, Activity lifecycle and event handling, Using intents and data passing between activities 6. Advanced Android Features: Working with databases and content providers, Networking and web services in Android, Integrating shell scripts within Android apps 7. Project Development and Deployment: Developing a complete Android app project, Testing and debugging Android applications Lab to be conducted on a 2-hour slot weekly.
Learning Outcome	• Write and execute shell scripts to automate various tasks in Unix/Linux environments. • Understand and apply advanced shell scripting techniques for more complex automation. • Develop Android applications using Java/Kotlin and Android Studio. • Design and implement user interfaces for Android apps. • Integrate shell scripting functionalities within Android applications.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

1. "Learning the bash Shell: Unix Shell Programming" by Cameron Newham, 3rd Edition.
2. "Shell Scripting: How to Automate Command Line Tasks Using Bash Scripting and Shell Programming" by Jason Cannon, 1st Edition.
3. "Android Programming: The Big Nerd Ranch Guide" by Bill Phillips, Chris Stewart, and Kristin Marsicano, 4th Edition.
4. "Head First Android Development: A Brain-Friendly Guide" by Dawn Griffiths and David Griffiths, 2nd Edition.
5. "Kotlin for Android Developers: Learn Kotlin the Easy Way While Developing an Android App" by Antonio Leiva, 1st Edition.

Course Number	**CS2206**
Course Credit (L-T-P-C)	**3-0-3-4.5**
Course Title	**Computer Architecture**
Learning Mode	Offline
Learning Objectives	The course is designed to provide basic understanding of structure, and function of various building blocks of computer system. Students will be able to design various functional units and components of computers and to identify the elements of modern instructions sets and their impact on processor design including memory hierarchy
Course Description	Using a set of fundamental techniques and technologies, the computer systems theme broadly explains how computing platforms work at various levels of abstraction, including both software and hardware. The course introduces computer architecture with focus on bridging the gap between high-level programming languages and the hardware (e.g., micro-processors) on which associated programs execute.
Course Outline	Computer types, RISC, CISCs, Structure of basic computer components. CPU Design Assembly language programming processor (MIPS); CPU control unit design: hardwired and micro-programmed design approaches, design of single Cycle MIPS, Multi cycle MIPS processor; Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards; Superscalar Architecture, Memory organization: cache memory, cache size vs block size, mapping functions, replacement algorithms, write policy; Peripheral devices and their characteristics: Input-output subsystems, Super Scalar Architectures, Super Pipelined Architecture, VLIW Architecture, SPARC and ARM processors. **Practical component**: The objective to this course is give hands on experience on computer architecture. It will provide an overview of understanding of the basic building blocks such Arithmetic Logic Blocks, register file, and memory. It also focuses on exploring various computer architecture simulators which include Simulation of Data Path and Control of CPUs and assembly language programming.
Learning Outcome	The student will be able to: • Demonstrate an understanding of how data is represented within a computer system. • Appreciate understanding of the basic blocks, key terminology, and current industry trends in computer architecture. • Demonstrate classic components of a computational system (i.e. input, output, memory, data path, control) and understanding their functionality. • Understand the processor (CPU) subsystem. • Employ concepts of the memory subsystem and hierarchy
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term.

Text books:

- M. Moris Mano, “Computer Systems Architecture”, 4th Edition, Pearson/PHI, ISBN:10:0131755633
- Carl Hamacher, Zvonko Vranesic, Safwat Zaky, “Computer Organization”, 5th Edition, McGraw Hill.
- John L. Hennessy and David A. Patterson, “Computer Architecture a quantitative approach”, 4th Edition Elsevier, ISBN:10:0123704901
- William Stallings, “Computer Organization and Architecture”, 6th Edition, Pearson/PHI, ISBN:10:0-13-609704-9

Sl. No.	**Subject Code**	**SEMESTER V**	**L**	**T**	**P**	**C**
1.	CS3101	Operating System	3	0	3	4.5
2.	CS3102	Computer Network	3	0	3	4.5
3.	CS3103	Machine Learning	3	0	3	4.5
4.	CS3104	Compiler	3	0	3	4.5
5.	XX31PQ	IDE - II	3	0	0	3
		TOTAL	**15**	**0**	**12**	**21**

Course Number	**CS3101**
Course Credit (L-T-P-C)	**3-0-3-4.5**
Course Title	**Operating System**
Learning Mode	Offline
Learning Objectives	This course provides an in-depth understanding of the fundamental concepts, principles, and mechanisms of operating systems. Topics include process management, memory management, file systems, concurrency, and scheduling.
Course Description	This course comprehensively introduces the fundamental concepts and principles underlying operating systems. Key topics include definitions of operating systems, the concept of a process, inter-process communication mechanisms, and multi-threading concepts. The course also addresses critical issues such as deadlock, discussing the necessary conditions for its occurrence and strategies for avoidance and prevention. In the realm of memory management, students will learn about both contiguous and non-contiguous allocation, paging concepts, and page table architecture. Further, the virtual memory concept will be explored, focusing on demand paging, replacement algorithms, and the phenomenon of thrashing. The course also includes a detailed study of file systems and disk management. By the end of this course, students will have a robust understanding of the essential components and functions of operating systems, preparing them for advanced studies and practical applications in the field of computer science.
Course Outline	**Basics of Operating System:** Definition and objectives of operating systems **Types of operating systems:** Batch, Time-sharing, Real-time, Distributed Systems **Concept of process:** Process control block, State transition, Scheduling algorithms, context switching, Process synchronization and inter-process communication **Threads:** Popular thread libraries, thread synchronization, multi-therading concepts **Deadlock:** necessary conditions, avoidance and prevention **Memory management:** Contiguous and non-contiguous allocation, Physical and logical addresses, Paging, different Page Table architectures, **Virtual Memory:** demand paging, replacement algorithms, thrashing. **File systems:** file operations, organization, mounting, sharing, File system implementation **Disk management:** disk structure, disk scheduling, disk management **Practical component**: Lab to be conducted on a 3-hour slot weekly. It will be conducted with the theory course so the topics for problems given in the lab are already initiated in the theory class.
Learning Outcome	• Understand the basic principles and functionalities of operating systems. • Analyse and evaluate different operating system components and their interactions. • Apply operating system concepts to solve real-world problems. • Develop an appreciation for the role of operating systems in modern computing environments.
Learning Outcome	• Understand the basic principles and functionalities of operating systems. • Analyse and evaluate different operating system components and their interactions. • Apply operating system concepts to solve real-world problems. • Develop an appreciation for the role of operating systems in modern computing environments.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

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

Course Number	**CS3102**
Course Credit (L-T-P-C)	**3-0-3-4.5**
Course Title	**Computer Network**
Learning Mode	Offline
Learning Objectives	The primary objectives of this course are to provide students with a solid foundation in computer networking principles and to prepare them for real-world networking challenges. Students will learn about network architectures, protocols, and technologies, and develop the skills necessary to design, implement, and manage networks. By the end of the course, students will be proficient in understanding network layers, configuring network devices, and troubleshooting network issues.
Course Description	This course provides an in-depth study of computer networks, covering essential concepts and technologies that form the backbone of modern communication systems. Students will learn about network topologies, protocols, hardware, and software that enable data transmission across networks. The course will also delve into advanced topics such as network security, wireless networking, and network management. Through practical exercises and projects, students will apply theoretical knowledge to real-world networking scenarios.
Course Outline	Introduction to computer networks and layered architecture, network applications, web architecture. Application Layer: HTTP, email protocols, DNS, and peer-to-peer applications. Transport layer: TCP, UDP, SCTP, and congestion control. Network layer: IP addressing, routing, and protocols like IPv4 and IPv6. link layer: LAN, error detection, MAC protocols. Physical Layer: Basics of data communication, transmission media and topology Future trends in networking: SDN, NFV **Practical component:** Lab to be conducted on a 3-hour slot weekly. It will be conducted with the theory course so the topics for problems given in the lab are already initiated in the theory class.
Learning Outcome	• Demonstrate an understanding of the core concepts and principles of computer networks. • Design and configure various types of network topologies and protocols. • Implement and manage network services and applications. • Identify and mitigate network security threats. • Analyze network performance and troubleshoot issues effectively.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Text Books:

1. "Computer Networking: A Top-Down Approach" (7th Edition) by James F. Kurose and Keith W. Ross
2. "Data Communications and Networking" (5th Edition) by Behrouz A. Forouzan
3. "Computer Networks" (5th Edition) by Andrew S. Tanenbaum and David J. Wetherall
4. "Network+ Guide to Networks" (8th Edition) by Jill West, Tamara Dean, and Jean Andrews
5. "TCP/IP Illustrated, Volume 1: The Protocols" (2nd Edition) by Kevin R. Fall and W. Richard Stevens

Course Number	**CS3103**
Course Credit (L-T-P-C)	**3-0-3-4.5**
Course Title	**Machine Learning**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) to understand the fundamental concepts of machine learning; (b) to develop the basic problem-solving skills by implementing the basic machine learning algorithms; (c) to learn about various paradigms of machine learning and various approaches under different paradigms; and (d) to achieve proficiency in designing some real-life project using machine learning.
Course Description	This course provides a comprehensive introduction to the field of Machine Learning (ML), covering fundamental concepts, techniques, and applications. It is designed to give students a solid foundation in understanding how machines learn from data and make decisions. Through a combination of theoretical insights and practical applications, students will explore various aspects of machine learning, including supervised and unsupervised learning, generalization, regression, classification, clustering, data reduction, and ensemble learning.
Course Outline	1.Understanding of Machine Learning: Definition, Tasks (Classification, Regression, Prediction, and Clustering), Supervised and unsupervised machine learning. 2.Learning to Generalization: Bias-Variance Trade-off, Overfitting vs. Underfitting, Regularization 3.Regression (single & multivariate, linear and nonlinear, Logistic Regression 4.Classification: (kNN, Bayes classifier, decision tree, random forest, Support vector Machines) 5.Unsupervised Learning: K-Means & variants, Hierarchical techniques 6.Data Reduction and Ensemble Learning **Practical component**: Lab to be conducted on a 3-hour slot weekly. It will be conducted with the theory course so the topics for problems given in the lab are already initiated in the theory class.
Learning Outcome	• Understanding of fundamental concepts of ML • Understanding different types of ML tasks: Classification, Regression, and Clustering • Understanding of various algorithms under different paradigms of ML: supervised, unsupervised, semi-supervised. • Capable of conducting some real-life projects using machine learning algorithms
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- T. Mitchell. Machine Learning. McGraw-Hill, 1997.
- Duda, Richard O., and Peter E. Hart. Pattern classification. John Wiley & Sons, 2006.
- Pattern recognition and machine learning by Christopher Bishop, Springer Verlag, 2006.
- Machine Learning in Action by Peter Harrington
- Probability, Random Variables and Stochastic processes by Papoulis and Pillai, 4th Edition, Tata McGraw Hill Edition.
- Linear Algebra and Its Applications by Gilbert Strand. Thompson Books.
- Data Mining: Concepts and Techniques by Jiawei Han, Micheline Kamber, Morgan Kaufmann Publishers.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988

Course Number	**CS3104**
Course Credit (L-T-P-C)	**3-0-3-4.5**
Course Title	**Compiler**
Learning Mode	Offline
Learning Objectives	The objective of the course is to introduce basic theory underlying different components and phases of a compiler, including parsing, code generation, optimization, etc. The students will also learn how to use various tools that are used for building modern compilers.
Course Description	This course is designed to cover various phases of compilers. Topics include regular languages for lexical analysis, context free languages for syntactic analysis, SDD/SDT for semantic analysis, IR code generation, and code optimization.
Course Outline	1. **Introduction to Compilers:** The role of language translation in programming; Interpreters Vs. Compilers, Language translation phases, Machine-dependent and machine-independent aspects of translation 2. **Lexical Analysis:** Regular expressions, Finite automata, Conflict Resolution 3. **Syntax Analysis:** Context-free grammars, Ambiguous grammers, Top-down parsing (LL parsing), Bottom-up parsing (LR parsing), CYK parser, Conflict resolution 4. **Semantic Analysis:** Formal Language Semantics, Symbol tables, Type checking, Attribute grammars 5. **Intermediate Code Generation:** Various Intermediate Representations (IR) of code (e.g., Abstract syntax trees (AST), Single Assignment (SA), Three-address code, etc.), Translation schemes **6.** **Optimization Techniques:** Various optimization scopes (Constant folding, constant propagation, invariant motion, etc.), Data-flow analysis (Liveness, available expression, very busy expression, reaching definition) 7. **Code Generation:** Instruction selection, Code generation for different architectures; Code generation tools **Lab Component:** Hands-on experience with various parsers, such as ANTLR, Lex, Flex, Yacc, Bison, etc.
Learning Outcome	The student will gain proficiency in both theory Parsing, Code generation, optimization) and practical (Lexer, Parser) aspects. They will be able to construct a compiler that converts from a non-trivial high level language to machine code.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term.

Suggested Readings:

1. J. E. Hopcroft, R. Motwani and J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Pearson Education India (3rd edition).
2. K. L. P. Mishra, N. Chandrasekaran, Theory of Computer Science: Automata, Languages and Computation, PHI Learning Pvt. Ltd. (3rd edition).
3. D. I. A. Cohen, Introduction to Computer Theory, John Wiley & Sons, 1997.
4. J. C. Martin, Introduction to Languages and the Theory of Computation, Tata McGraw-Hill (3rd Ed.).
5. H. R. Lewis and C. H. Papadimitriou, Elements of the Theory of Computation, Prentice Hall, 1997.
6. Garey, D.S., Johnson, G., Computers and Intractability: A Guide to the Theory of NP- Completeness, Freeman, New York, 1979
7. M. Sipser, Introduction to the Theory of Computation, Thomson, 2004

Sl. No.	Subject Code	SEMESTER VI	L	T	P	C
1.	CS3201	Cyber Security	3	0	2	4
2.	CS3202	Deep Learning	3	0	3	4.5
3.	CS3203	Internet of Things	3	0	3	4.5
4.	CS32XX	DE-I (CS ELECTIVES LIST)	3	0	0	3
5.	CS3299	Capstone Project	0	0	6	3
		TOTAL	**12**	**0**	**14**	**19**

Course Number	**CS3201**
Course Credit (L-T-P-C)	**3-0-2-4**
Course Title	**Cyber Security**
Learning Mode	offline
Learning Objectives	To understand the basic concepts of cyber-attacks, legal issues and countermeasures.
Course Description	The course covers cyber-attacks, legal issues and countermeasures various aspects of cybersecurity, including basic principles, legal considerations, risk assessment, and security management. The course covers essential topics such as cybercrime, phishing attacks, cryptography basics, authentication mechanisms, and authorization protocols. Additionally, it delves into specific areas of vulnerability assessment and mitigation, focusing on secure programming practices and identifying threats to networks.
Course Outline	Introduction to cybersecurity: Basic concepts, cybercrime, legal issues, risk analysis and security management, phishing attack. Crypto basics, Authentication and authorization, Kerberos, PKI Vulnerabilities and Countermeasure: Vulnerabilities in code, Secure programming. Threats to network, network defense, social network security issues and countermeasures, email security Cyber system security: Hardware security, mobile security. Practical component: Lab to be conducted on a 2-hour slot weekly. It will be conducted with the theory course so the topics for problems given in the lab are already initiated in the theory class.
Learning Outcome	After completion of this course a student will have: • Understanding the legal aspects, risk and vulnerabilities in cyberspace. • Understanding the concepts of different attacks and their countermeasures in cyberspace.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

Nina Godbole and Sunit Belapure, Cyber Security, Wiley India

Course Number	**CS3202**
Course Credit (L-T-P-C)	**3-0-3-4.5**
Course Title	**Deep Learning**
Pre-requisite	offline
Learning Mode	This course aims to provide an introductory overview of deep learning and its application varied domains. The course will provide basic understanding of neural networks, mathematical description of it and finally applications of it in multiple domains. A few open source tools will be demonstrated during the course to provide hands-on experience.
Learning Objectives	This course will provide an overview of neural networks and hands-on experience for the same.
Course Description	Introduction: Introduction to bigdata problem, overview of linear algebra Feature engineering: Basics of machine learning (linear regression, classification) Neural network: Deep feed forward network, cost function, activation functions, overfitting, underfitting, Universal approximation theorem Gradient based learning: Gradient Descent, Stochastic Gradient Descent, Backpropagation Regularization: L2, L1, L\infinity, drop-out, early stopping, data augmentation, etc. Optimization: Multivariable taylor series, momentum, adaptive learning rate, ADAM, Nesterov Accelerated Gradient (NAG), AdaGrad, etc. Convolutional Neural Network (CNN): Theory and its application in computer vision Recurrent Neural Network (RNN): Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) and their applications in natural language processing Advanced topics: Autoencoder, Transformer, Deep reinforcement learning **Practical component:** Lab to be conducted on a 3-hour slot weekly. It will be conducted with the theory course so the topics for problems given in the lab are already initiated in the theory class.
Course Content	• Basic understanding of deep learning and neural networks • Problem modeling skill • Usage of different open source tools / libraries Analysis of large volume of data
Learning Outcome	Knowledge on various forms of stresses in pressure vessels and their relation. Mechanical designing of different parts/components used in heat exchangers or in separation units such as nuts/bolts, flanges, heads, shell, etc. Consideration and elementary sizing calculation on tall, horizontal/vertical vessels, and their constructional supports.
Assessment Method	Assignments, Quiz, Mid-semester examination and End-semester examination.

Suggested Reading:

- Ian Goodfellow, Yoshua Bengio and Aaron Courville, “Deep Learning”, Book in preparation for MIT Press, 2016.
- Reference books:
- Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie, “The elements of statistical learning”, Springer Series in Statistics, 2009.
- Charu C Aggarwal, “Neural Networks and Deep Learning”, Springer.
- Aston Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola, "Dive into Deep Learning"
- Iddo Drori, "The Science of Deep Learning", Cambridge University Press
- Simon O. Haykin, "Neural Networks and Learning Machines", Pearson Education India
- Richard S. Sutton, Andrew G. Barto, "Reinforcement Learning: An Introduction", MIT Press
- C. M. Bishop, H. Bishop, "Deep Learning: Foundations and Concepts", Springer, 2022
- Simon J. D. Prince, "Understanding Deep Learning", MIT Press 2023

Course Number	**CS3203**
Course Credit (L-T-P-C)	**3-0-3-4.5**
Course Title	**Internet of Things**
Learning Mode	Offline
Learning Objectives	• The layout of the course follows various popular IoT courses being followed by various universities globally • This course will also provide a brief overview of the very basics of networking, the precursor technologies of IoT and the emergence of IoT so that the students are not abruptly faced with complex ideas and terminologies from the onset of this course. • This course includes various building blocks, which are essential for developing an IoT platform • After describing different computing technologies in IoT, a few use cases are described. This part of the course will help a student to understand the use of different components, described in the previous lectures, in real life. • Conduct tutorial classes, in which the practical knowledge of IoT implementation will be provided to the students. Additionally, two projects will be provided to the students to learn the hands-on in IoT
Course Description	This undergraduate course provides a foundational understanding of Internet of Things (IoT). Students will start by learning the essential concepts of IoT, including sensors, actuators, connectivity and communications protocols. This course provides a comprehensive overview of the principles, technologies, and applications of IoT. Through hands-on projects and real-world case studies, students will learn how to create innovative IoT applications in various domains such as smart homes, healthcare, industrial automation, and smart cities.
Course Outline	Introduction to IoT: Predecessors of IoT, Emergence of IoT Sensors, Actuators, and Processors: Sensors, Actuators and their types, IoT Processing Topologies and types IoT Communications Technologies: IEEE 802.15.4, ZigBee, Wireless HART, RFID, NFC, Z-Wave, LoRa, Sigfox, NB-IoT, Wi-Fi IoT Communication Protocols: Introduction to Constrained Environment, Infrastructure Protocols, Discovery Protocols, Data Protocols Cloud and Fog Computing for IoT: Virtualization, Cloud models, Service-Level Agreement in Cloud, Basics of Fog computing, Fog Nodes IoT Applications: Agricultural IoT, Vehicular IoT, Healthcare IoT, Paradigm, Challenges, and the Future of IoT **Lab outline:** - Working with IoT Processor boards: Node MCU and Arduino, Raspberry-Pi - Sensors and actuators integration - Working with IoT Communication module: Integration and data communication - Implementing IoT communication protocols: Infrastructure, discovery, and data protocols - IoT-based alert generation systems: Temperature, Humidity, and Air quality monitoring systems, - Implementation of cloud and fog computing for IoT - Developments of Smart home, smart irrigation, and smart parking systems using cloud and fog computing Lab to be conducted on a 3-hour slot weekly.
Learning Outcome	• A student will be capable of applying the various computing technologies and designing the system for implementing an IoT platform.

	• A student will be capable of applying ideas and knowledge in developing IoT systems. • A student will know primary focus areas in IoT, including interoperability, Cloud Computing, Fog, and Edge computing. • A student will be capable of developing IoT projects for different applications • The tutorial classes of the course will provide an opportunity to the student for learning IoT hands-on, and working in the projects
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term.

Suggested Reading

- Peter Waher, Learning Internet of Things, Packt Publishing Ltd., UK, 2015
- Michael Miller, Internet of Things The: How Smart TVs Sm: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities Are Changing the World, Pearson Education, Inc, USA, 2015
- Sudip Misra, Anandarup Mukherjee, and Arijit Roy. Introduction to IoT. Cambridge: Cambridge University Press, (2021) doi:10.1017/9781108913560
- Cuno Pfister, Getting Started with the Internet of Things, O' Reilly Media, Inc, Gravenstein Highway North, Sebastopol, CA, 2011
- Adrian McEwen and Hakim Cassimally, Designing the Internet of Things, John Wiley and Sons, Ltd, West Sussex, UK, 2014

Department Elective - I						
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1.	CS3205	Object-Oriented Programming	3	0	0	3
2.	CS3206	Agile Computing	3	0	0	3
3.	CS3207	Software Engineering	3	0	0	3
4.	CS3208	Bayesian Data Analysis	3	0	0	3
5.	CS3209	Data Mining	3	0	0	3
6.	CS3210	Information Retrieval	3	0	0	3

Course Number	**CS3205**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Object-Oriented Programming**
Learning Mode	Offline
Learning Objectives	The primary objectives of this course are to introduce students to the principles and practices of object-oriented programming (OOP) and to equip them with the skills necessary to design and implement software using OOP techniques. Students will learn about core OOP concepts such as classes, objects, inheritance, polymorphism, encapsulation, and abstraction. They will also develop proficiency in using an object-oriented programming language such as Java or Python.
Course Description	This course provides a comprehensive introduction to the fundamental concepts and methodologies of OOP. The course covers essential topics such as class and object design, inheritance, polymorphism, and encapsulation, and explores advanced concepts including exception handling, file I/O, and graphical user interfaces (GUIs). Through a series of practical exercises and projects, students will gain hands-on experience in writing clean, efficient, and maintainable code. The course emphasizes best practices and design patterns that are critical for developing robust software applications.
Course Outline	1. Introduction to Object-Oriented Programming, Overview of programming paradigms, Key concepts of OOP: classes, objects, and methods, Benefits of OOP 2. Classes and Objects, Defining and creating classes, Constructors and destructors, Object lifecycle and memory management 3. Encapsulation and Data Hiding, Access modifiers (public, private, protected), Getters and setters, Maintaining data integrity 4. Inheritance and Polymorphism, Base and derived classes, Method overriding and overloading, Dynamic binding and polymorphic behavior 5. Abstraction and Interfaces, Abstract classes and methods, Interface implementation, Multiple inheritance in OOP 6. Object-Oriented Design Principles, SOLID principles, Design patterns (e.g., Singleton, Factory, Observer), UML diagrams for OOP design 7. Exception Handling and File I/O, Error detection and handling, using exceptions to manage errors, File input/output operations 8. Advanced OOP Concepts, Generic programming and templates, Reflection and metadata, Multithreading in OOP.
Learning Outcome	Upon successful completion of this course, students will be able to: • Understand and apply the core principles of object-oriented programming. • Design and implement software solutions using object-oriented techniques. • Develop and debug programs in an object-oriented programming language. • Utilize advanced OOP features such as inheritance, polymorphism, and interfaces effectively. • Write clean, maintainable, and efficient code following best practices and design patterns. • Create basic graphical user interfaces and handle events in GUI applications. • Apply OOP concepts in various real-world scenarios and software development projects.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

1. "Object-Oriented Analysis and Design with Applications" by Grady Booch
2. Data Structures and Algorithm Analysis in C++ Hardcover, by Mark A. Weiss, Jun 2013, Publisher: PHI; 4 editions, ISBN-10: 013284737X ISBN-13: 978-0132847377.
3. Algorithms in C++: Fundamentals, Data Structures, Sorting, Searching, Parts 1-4, 3rd Edition (Paperback), Pearson India, ISBN-10 8131713059, 2009, ISBN-13 9788131713051.
4. "Thinking in C++" by Bruce Eckel
5. "C++ Primer" by Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo
6. "Head First Object-Oriented Analysis and Design" by Brett McLaughlin, Gary Pollice, and David West

Course Number	**CS3206**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Agile Computing**
Learning Mode	Offline
Learning Objectives	• To gain a thorough understanding of agile computing principles, methodologies, and their application in software development. • To learn to effectively apply agile practices such as Scrum, Kanban, and Extreme Programming (XP) to enhance project visibility, collaboration, and adaptability. • To develop skills in managing and leading agile teams, utilizing agile project management tools for planning and tracking projects. • To acquire knowledge of metrics and performance measurement techniques to analyze and optimize agile processes. • To apply agile principles to cultivate a culture of continuous improvement and innovation within organizations.
Course Description	This course provides a comprehensive exploration of agile computing, focusing on its principles, methodologies, and practical applications in software development. Students will delve into popular agile frameworks like Scrum, Kanban, and Extreme Programming (XP), learning how these methodologies enhance project management, collaboration, and responsiveness to change. Topics include agile estimation, planning, testing, quality assurance, and scaling agile practices. The course also covers agile leadership, metrics for performance measurement, and fostering an agile culture of continuous improvement and innovation.
Course Outline	Introduction to Agile Computing and scope Overview of popular agile methodologies like Scrum, Kanban, and Extreme Programming (XP), Scrum roles, artifacts, and events, Lean and Kanban Principles, Extreme Programming (XP): est-driven development (TDD) and pair programming, Agile Estimation and Planning, Agile Testing and Quality Assurance, Scaling Agile, Agile Leadership and Culture, Agile Metrics and Performance Measurement, Applications of agile computing
Learning Outcome	By the end of this course, students will be able to: • Understand the principles and philosophy of agile computing. • Apply various agile methodologies and practices to software development projects. • Effectively manage and lead agile teams. • Use agile project management tools to plan, track, and deliver projects. • Analyze and optimize agile processes using metrics and performance measurement techniques. • Apply agile principles to foster a culture of continuous improvement and innovation within organizations
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

1. "Agile Estimating and Planning" by Mike Cohn, 2006
2. "Agile Testing: A Practical Guide for Testers and Agile Teams" by Lisa Crispin, Janet Gregory, 2009
3. "Scrum: The Art of Doing Twice the Work in Half the Time" by Jeff Sutherland, 2014
4. "Kanban: Successful Evolutionary Change for Your Technology Business" by David J. Anderson, 2010
5. "Extreme Programming Explained: Embrace Change" by Kent Beck, 2004
6. "Lean Software Development: An Agile Toolkit" by Mary Poppendieck, Tom Poppendieck, 2003

Course Number	**CS3207**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Software Engineering**
Learning Mode	Offline
Learning Objectives	This course aims to help the students • with a comprehensive understanding of the fundamental principles and concepts of software engineering; • with the software development life cycle (SDLC) and various software process models; • in using modern software engineering tools and techniques for efficient software development; and • understanding of quality assurance practices and the importance of software project documentation.
Course Description	This comprehensive course provides an in-depth understanding of the principles and practices of software engineering. Students will explore the software development lifecycle, including requirements analysis, design, implementation, testing, and maintenance. Emphasis is placed on methodologies such as Agile, Waterfall, and DevOps. Key topics include software project management, version control, software architecture, design patterns, and quality assurance. Through hands-on projects and case studies, students will gain practical experience in developing reliable, scalable, and maintainable software systems. This course prepares students for real-world challenges in software engineering, equipping them with the skills necessary for successful careers in the tech industry.
Course Outline	Software life cycle- important steps and effort distribution. Aspects of estimation and scheduling. Software evaluation techniques-modular design- coupling and cohesion, Software and complexity measures. Issues in software reliability. System Analysis- Requirement analysis. Specification languages. Feasibility analysis. File and data structure design, Systems analysis tools. **S**oftware design methodologies- Data flow and Data Structure oriented design strategies. Software development, coding, verification, and integration. Issues in project management- team structure, scheduling, software quality assurance.
Learning Outcome	• Demonstrate a clear understanding of the fundamental concepts and methodologies in software engineering. • Apply software engineering principles and techniques to design, develop, test, and maintain software systems. • Use modern software engineering tools and environments effectively in software development tasks. • Plan and manage software projects, including tasks such as requirements analysis, project scheduling, risk management, and quality assurance. • Produce and maintain comprehensive documentation for all phases of the software development process. • Work effectively as part of a software development team, demonstrating strong collaboration and communication skills.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides (The Gang of Four)
- "Software Architecture in Practice" by Len Bass, Paul Clements, and Rick Kazman
- "Software Requirements" by Karl E. Wiegers and Joy Beatty
- "Software Engineering: A Practitioner's Approach" by Roger S. Pressman and Bruce R. Maxim
- Fundamentals of Software Engineering, Fifth Edition, Rajiv Mall

Course Number	**CS3208**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Bayesian Data Analysis**
Learning Mode	Offline
Learning Objectives	The learning objectives of the course include comprehending the fundamental concepts of Bayesian statistics, such as likelihood and priors, and applying these to develop various models, including single-parameter, multi-parameter, and hierarchical models. Additionally, techniques for validating these models will be covered. Students will also learn the programming skills necessary to computationally implement these models for different real-world problems.
Course Description	The primary goal of this course is to introduce Bayesian approaches for data analysis and apply these techniques to various real-world problems. Although the focus will be on issues pertinent to computer science, the skills acquired are broadly applicable across several disciplines related to machine learning. The lectures will cover the fundamental theory behind Bayesian statistical inference. Additionally, the course will introduce programming languages like R and Stan, which are well-suited for implementing these Bayesian concepts.
Course Outline	Basics of Probability and Inference, Single Parameter Models, Multiparameter models, Programming Bayesian models using R, Bayesian Computation Techniques, Markov-chain Monte Carlo simulations, Programming Stan with R, Efficient Markov chain simulation techniques, Hierarchical models, Model checking, Model Evaluation, Case studies
Learning Outcome	On successful completion of this course students will be able to: • Assess the fundamental philosophical differences between Bayesian probability and traditional frequentist approaches. • Construct flexible Bayesian models using likelihood and prior functions. • Implement Markov Chain Monte Carlo (MCMC) algorithms in R and Stan for inference in small to medium-sized problems. • Develop Bayesian machine learning algorithms capable of inference in high-dimensional problems.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

- Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin, Bayesian Data Analysis, Third Edition, CRC Press
- John K. Kruschke, Doing Bayesian Data Analysis, A Tutorial with R, JAGS, and Stan, Second Edition, Academic Press

Course Number	**CS3209**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Data Mining**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) Understand the fundamental concepts and techniques of data mining. (b) Gain proficiency in data preprocessing, including data cleaning, transformation, and reduction. (c) Apply various data mining algorithms for classification, clustering, association, and anomaly detection. (d) To achieve proficiency in designing some real-life projects using data mining techniques.
Course Description	This comprehensive course on data mining aims to equip students with the knowledge and skills required to extract meaningful insights from large datasets. By focusing on core concepts and providing practical experiences, students will learn to apply various data mining techniques and tools effectively. Through a combination of lectures and real-world projects, students will explore topics such as classification, clustering, association rule mining, and anomaly detection. Upon completion, students will be adept at transforming raw data into actionable knowledge, enabling them to solve complex problems and make data-driven decisions in academic and professional settings.
Course Outline	Fundamentals of data warehousing, architectures, schemas, OLAP technology, and data cube processing. Data preprocessing, integration, transformation, reduction, and basics of data mining techniques. Association rule mining, algorithms (Apriori, FP-Growth), and latest trends in association rule mining. Data classification and clustering techniques, algorithms, prediction methods, and outlier analysis. Introduction to web, spatial and temporal text mining, security, privacy, and ethical issues.
Learning Outcome	• Mastery of fundamental concepts and techniques in data mining. • Proficiency in various data mining algorithms. • Comprehensive understanding of essential data mining tasks such as association rule mining, clustering, and classification. • Ability to apply data mining techniques to real-world projects.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

<u>Suggested Reading:</u>

- Arun K. Pujari “Data Mining Technique” University Press
- Han, Kamber, “Data Mining Concepts & Techniques”,
- M. Kaufman., P.Ponnian, “Data Warehousing Fundamentals”, JohnWiley.
- M.H.Dunham, “Data Mining Introductory & Advanced Topics”, Pearson Education.
- Ralph Kimball, “The Data Warehouse Lifecycle Tool Kit”, John Wiley.
- E.G. Mallach, “The Decision Support & Data Warehouse Systems”, TMH

Course Number	**CS3210**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Information Retrieval**
Learning Mode	Offline
Learning Objectives	The potential learning objectives of the course includes understanding the fundamental concepts and theories of information retrieval, including indexing, querying, and relevance ranking. Furthermore, the students will gain proficiency in utilizing various retrieval models, such as boolean, vector space, and probabilistic models. They would learn about the challenges and techniques involved in processing natural language for information retrieval purposes. The students would be familiarized with the architecture and components of modern search engines and recommendation systems.
Course Description	This course focuses on Information Retrieval (IR), which involves extracting pertinent data from extensive document sets. IR finds utility in various realms such as proprietary retrieval systems, the World Wide Web, Digital Libraries, and commercial recommendation platforms. The course aims to acquaint students with the theoretical foundations of IR with several real world applications and examples.
Course Outline	Introduction: concepts and terminology of information retrieval systems, Information Retrieval Vs Information Extraction Indexing: inverted files, encoding, Zipf's Law, compression, boolean queries Fundamental IR models: Boolean, Vector Space, probabilistic, TFIDF, Okapi, language modeling, latent semantic indexing, query processing and refinement techniques Performance Evaluation: precision, recall, F-measure; Classification: Rocchio, Naive Bayes, k-nearest neighbors, support vector machine Clustering: partitioning methods, k-means clustering, hierarchical Introduction to advanced topics: search, relevance feedback, ranking, query expansion.
Learning Outcome	Course training via lectures & tutorial sessions to • Understand the fundamental concepts and theories of information retrieval, including indexing, querying, and relevance ranking. • Gain proficiency in utilizing various retrieval models, such as boolean, vector space, and probabilistic models. • Learn about the challenges and techniques involved in processing natural language for information retrieval purposes. • Acquire knowledge of evaluation metrics and methodologies used to assess the performance of information retrieval systems. • Familiarize with the architecture and components of modern search engines and recommendation systems. • Analyze case studies and real-world applications of information retrieval in diverse domains, including web search, digital libraries, and e-commerce.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schtze, Introduction to Information Retrieval, Cambridge University Press. 2008.
- Ricardo Baeza-Yates and Berthier Ribeiro-Neto, Modern Information Retrieval, Addison Wesley, 1st edition, 1999.
- Soumen Chakrabarti, Mining the Web, Morgan-Kaufmann Publishers, 2002.
- Bing Liu, Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Springer, Corr. 2nd printing edition, 2009.

Sl. No.	Subject Code	SEMESTER VII	L	T	P	C
1.	CS41XX	DE-II (AI ELECTIVES LIST)	3	0	0	3
2.	CS41XX	DE-III (AI ELECTIVES LIST)	3	0	0	3
3.	XX41PQ	IDE - III	3	0	0	3
4.	HS41XX	HSS Elective - II	3	0	0	3
5.	CS4198	Summer Internship*/ Summer Project	0	0	12	3
6.	CS4199	Project – I	0	0	12	6
		TOTAL	**12**	**0**	**24**	**21**

Department Elective - II						
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1.	CS4101	Pattern Recognition	3	0	0	3
2.	CS4102	Principles of Programming Languages	3	0	0	3
3.	CS4103	Social Networks	3	0	0	3
4.	CS4104	Multimedia System	3	0	0	3
5.	**CS4105**	Program Analysis and Verification	3	0	0	3

Course Number	**CS4101**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Pattern Recognition**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) Understand the fundamental principles and techniques of pattern recognition, including classification and clustering methods. (b) To develop basic problem-solving skills by implementing the basic pattern recognition algorithms. (c) To gain proficiency in feature extraction, selection, and dimensionality reduction to enhance pattern recognition performance. (d) Apply pattern recognition algorithms to practical applications in image processing, speech recognition, and data mining.
Course Description	This course on pattern recognition aims to equip students with the theoretical foundations and practical skills necessary to identify and analyze patterns in data. By focusing on essential principles, students will develop the ability to implement and evaluate various pattern recognition algorithms. Students will enhance their understanding of machine learning, statistical methods, and data preprocessing techniques through interactive lectures, exercises, and projects. Upon completion, students will be proficient in designing and applying pattern recognition systems for applications such as image processing, speech recognition, and data mining, thereby enhancing their analytical and problem-solving capabilities in diverse domains.
Course Outline	Introduction to pattern recognition, key concepts, learning types, approaches, decision boundaries, and distance metrics. Pattern extraction and preprocessing, pattern classification and algorithms Different paradigms and representations for pattern clustering techniques and validation. Feature extraction and selection methods, problem statements, and relevant algorithms (branch and bound, sequential selection). Recent advances in pattern recognition, including structural pattern recognition, neuro-fuzzy techniques, and real-life applications.
Learning Outcome	• Mastery of fundamental concepts in pattern recognition. • In-depth understanding of various algorithms across different pattern recognition paradigms. • Comprehensive knowledge of theoretical aspects of feature selection, feature extraction, and projection techniques. • Ability to apply pattern recognition algorithms to real-world projects
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Pattern recognition and machine learning by Christopher Bishop, Springer Verlag, 2006.
- Trevor Hastie, Robert Tibshirani , Jerome Friedman. The elements of Statistical Learning. Springer Verlag (2009).
- Fundamentals of Pattern Recognition and Machine Learning by Ulisses Braga-Neto. Springer Cham (2020)
- Probability, Random Variables and Stochastic processes by Papoulis and Pillai, 4th Edition, Tata McGraw Hill Edition.
- Linear Algebra and Its Applications by Gilbert Strand. Thompson Books.
- Data Mining: Concepts and Techniques by Jiawei Han, Micheline Kamber, Morgan Kaufmann Publishers.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988

Course Number	**CS4102**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Principles of Programming Languages**
Learning Mode	Offline
Learning Objectives	To make students understand the existence of different programming language paradigms (i.e., logic, functional, procedural, object-oriented), their specific features, and to choose an appropriate language for a given application. To make students capable to learn new languages easily and to make clear and efficient use of any given language.
Course Description	The objective of this course is to study the design and implementation of programming languages from a foundational perspective.
Course Outline	**Introduction:** History of Programming Languages; Evolution of the Major Programming Languages; Art of Programming Language Design; Properties and Success of Programming Languages. **Programming Language-Paradigms:** Imperative (e.g. C, Pascal, Fortran); Functional (e.g. LISP, HASKELL, OCaml); Object Oriented (e.g. JAVA, C++, Scala); Logic-based (e.g. Prolog); Multiparadigm programming languages (e.g. Python, C++11). **Programming Language Concepts:** Values and Data Types; Block Structure; Scope, Binding and Lifetime of Variables; Static vs. Dynamic Typing; Static vs. Dynamic Scoping; Memory Management; Procedural Abstraction; Data Abstraction; Concurrency; etc. **Case Study:** Defining Syntax and Semantics of IMP (a simple WHILE-language) and COOL (Classroom Object Oriented Language).
Learning Outcome	• Understand a variety of concepts underpinning modern programming languages. • Understand the concepts and terms used to describe languages that support the imperative, functional, object-oriented, and logic programming paradigms. • Critically evaluate what paradigm and language are best suited for a new problem. • Solve problems using the functional paradigm. • Solve problems using the object-oriented paradigm. • Solve problems using the logic programming paradigm. • Understand how to design and implement your own (domain-specific) language.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Michael L. Scott, **“Programming Language Pragmatics”**, Morgan Kaufmann, 3rd Edition.
- Harold Abelson, Gerald Jay Sussman, Julie Sussman, **“Structure and Interpretation of Computer Programs**”, MIT Press, 2nd Edition.
- Ravi Sethi, K.V. Vishwanatha, “Programming Languages: Concepts and Constructs”, 2/e, Pearson Education, 2007.
- T.W. Pratt and M.V. Zelkowitz, “Programming Languages – Design and Implementation”, Prentice-Hall.
- Robert W. Sebesta, “Concepts of Programming Languages”, Addison-Wesley.
- D. A. Watt, “Programming Language Design Concepts”, John Wiley & Sons.
- Kenneth C. Louden and Kennath A. Lambert, “Programming Languages: Principles and Practice”, Cengage Learning.
- Recent Research Papers relevant to the course.

Course Number	**CS4103**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Social Networks**
Learning Mode	Offline
Learning Objectives	The major objectives of the course would be to make the students understand the basic concepts of social network, understand the fundamental concepts in analyzing the large-scale data that are derived from social networks, implement mining algorithms for social networks, and perform mining on large social networks and illustrate the results.
Course Description	This course delves into the analysis of data within social networks, emphasizing efficient strategies for managing large-scale networks. It presents fundamental theoretical findings in social network mining alongside practical exercises addressing critical topics within the field.
Course Outline	Introduction to social networks. Illustration of various social network mining tasks with real-world examples. Data characteristics unique to these settings and potential biases due to them. Social Networks as Graphs. Random graph models/ graph generators (Erdos-Renyi, power law, preferential attachment, small world, stochastic block models, Kronecker graphs), degree distributions. Models of evolving networks. Node based metrics, ranking algorithms (Pagerank). Graph visualisation. Social network exploration/ processing: Graph kernels, graph classification, clustering of social-network graphs, centrality measures, community detection and mining, degeneracy (outlier detection and centrality), partitioning of graphs. Information Diffusion in Social Networks: Information diffusion in graphs - Cascading behavior, spreading, epidemics, heterogeneous social network mining, influence maximization, outbreak detection; Opinion analysis on social networks - Contagion, opinion formation, coordination and cooperation. Dynamic social networks, Link prediction, Social learning on networks.
Learning Outcome	By completing the course, the students will be able to: • Understand the basic concepts of social networks • Understand the fundamental concepts in analyzing the large-scale data that are derived from social networks • Implement mining algorithms for social networks • Perform mining on large social networks and illustrate the results.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

- David Easley and Jon Kleinberg, Networks, crowds, and markets, Cambridge University Press, 2010.
- Jure Leskovec, Anand Rajaraman and Jeffrey David Ullman, Mining of massive datasets, Cambridge University Press, 2014.

Course Number	**CS4104**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Multimedia Systems**
Learning Mode	Offline
Learning Objectives	The main objective of this course is to provide students with a comprehensive understanding of multimedia systems. Students will learn about the various components and technologies involved in multimedia systems, including audio, video, and image processing. They will explore the principles of multimedia compression, storage, and retrieval, as well as the techniques used for multimedia communication and networking. By the end of the course, students will have a solid theoretical foundation in multimedia systems and will be able to apply this knowledge to solve real-world problems in the field.
Course Description	The course begins with an introduction to multimedia systems, covering the basics of multimedia data representation and the different types of multimedia data. This is followed by a detailed study of multimedia compression techniques, including lossless and lossy compression methods for text, images, audio, and video. The course then explores multimedia storage and retrieval, discussing the different storage media and retrieval techniques used for multimedia data. Next, students will learn about multimedia communication and networking, including the protocols and architectures used for multimedia transmission over networks. The course concludes with a discussion of advanced topics in multimedia systems, such as quality of service, synchronization, and security.
Course Outline	Introduction to Multimedia Systems Understanding multimedia data types: Text, images, audio, and video. Multimedia Data Representation: Pixel-based representation for images, waveform representation for audio, and frame-based representation for video. Compression techniques for multimedia data: Lossy and lossless compression algorithms. Multimedia Storage and Retrieval Multimedia Networking and Streaming Multimedia Synchronization and Interactivity: Timecodes, timestamps, and synchronization protocols, Hypermedia, and interactive multimedia applications. Multimedia applications and trends: virtual and augmented reality
Learning Outcome	By the end of this course, students will be able to: • Understand the fundamental concepts and components of multimedia systems. • Analyse and evaluate different multimedia data types and their representation techniques. • Design and implement multimedia storage, retrieval, and streaming solutions. • Evaluate multimedia networking protocols and techniques for efficient multimedia transmission. • Implement multimedia synchronization and interactivity features in multimedia applications. • Explore real-world applications of multimedia systems and identify future trends in multimedia technology
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

- Multimedia Systems: Algorithms, Standards, and Industry Practices" by Parag Havaldar, Gerard Medioni
- "Multimedia Computing: Algorithms, Systems, and Applications" by Ralf Steinmetz, Klara Nahrstedt
- "Introduction to Multimedia Systems" by Sugata Mitra, Tamalika Chaira

Course Number	**CS4105**
Course Credit	**3-0-0-3**
Course Title	**Program Analysis and Verification**
Learning Mode	Offline
Learning Objectives	This course will focus on static and dynamic program analysis techniques that can be used to perform various software engineering tasks. The students will learn the concepts behind the techniques and will apply their learning to develop analyses using the state-of-art tools.
Course Description	This course covers both foundations and practical aspects of the automated analysis of programs. It covers how to represent source codes in appropriate forms, enabling one to apply tools and techniques to extract relevant information about the code and to verify them.
Course Outline	Introduction: Program Analysis and Verification; Abstraction Vs. Approximation; Precision-Efficiency Dilemma. Control and Data Flow: Control-flow analysis; Data-flow analysis; Inter-procedural Analysis. Dependency Analysis: Program dependence graphs; Program slicing; Pointer and Alias analysis. Verification and Validation: Verification and validation strategies; Different testing methodologies; Introduction to formal verification approaches.
Learning Outcome	By taking this course, students will gain an understanding of the concepts and theories that underlie these analysis techniques. Students will also learn to design, implement, and leverage program analysis techniques to solve new verification problems.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term.

Suggested Readings:

- Flemming Nielson, Hanne R. Nielson, Chris Hankin. Principles of Program Analysis, Springer, 1999.
- Edsger Wybe Dijkstra. A Discipline of Programming. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1997.
- David Gries. The Science of Programming. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1987.
- S. S. Muchnick and N. D. Jones, editors. Program Flow Analysis: Theory and Applications. Prentice-Hall: Englewood Cliffs, NJ, 1981.
- Cem Kaner, Hung Q. Nguyen, and Jack L. Falk. Testing Computer Software. John Wiley & Sons, Inc., New York, NY, USA, 1993.
- Recent Research Papers relevant to the course.

Department Elective - III						
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1.	CS4106	Graph Machine Learning	3	0	0	3
2.	CS4107	Bioinformatics	3	0	0	3
3.	CS4108	Time Series Analysis	3	0	0	3
4.	CS4109	Computational Data Analysis	3	0	0	3
5.	CS4110	Blockchain Technology	3	0	0	3
6.	CS4112	Graph Theory	3	0	0	3

Course Number	**CS4106**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Graph Machine Learning**
Learning Mode	Offline
Learning Objectives	Several real world systems can be represented as a network of entities that are connected to each other through some relations. Often the number of entities is immensely large, thus forming a very large network. Typical examples of such large networks include network of entities in knowledge graphs, co-occurrence graph of the keywords in natural languages, interaction graph of users in social networks, protein-protein interaction graphs and the network of routers in Internet to name a few. Study of these networks is often needed for relational learning tasks, as well as for developing frameworks for representing the intrinsic structure of the data. This course will mainly deal with both the traditional as well as current state of the art machine learning techniques to be applied on Graphs for different downstream tasks.
Course Description	The course will provide knowledge on the representation and statistical descriptions of large networks, along with traditional machine learning and deep learning techniques applied on graphs. Several use cases of Graph Machine Learning across different domains including Natural Language Processing, Social Network Analysis and Computational Biology would be studied.
Course Outline	Introduction and background knowledge of graphs; Network analysis metrics like paths, components, degree distribution, clustering, degree correlations, centrality etc., social network analysis methods; Spectral Analysis of Graphs and its applicability to graph partitioning and community detection; Overview of machine learning applications on graphs; Shallow embedding and deep Learning techniques for generating node and graph representations – Graph Neural Networks, Graph Attention Networks Random Networks; Graph Evolution, Generative models for graphs
Learning Outcome	Course training via lectures & tutorial sessions to • Represent and analyze the structure of graphs • Discover recurring and significant patterns of interconnections in your data with network motifs and community structure. • Gain Knowledge on traditional machine learning techniques applied on graphs • Leverage graph-structured data to make better predictions using graph neural networks • Understand the problems in dealing with large graphs for machine learning tasks and learn how to improvise.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

- M.E.J. Newman, Networks - An introduction , Oxford Univ Press, 2010.
- Yao Ma and Jilian Tang, Deep Learning on Graphs, Cambridge University Press, 2021
- Goyal, Palash and Emilio Ferrara. “Graph embedding techniques, applications, and performance: A survey.” *Knowl.-Based Syst.* 151 (2018): 78-94.

Course Number	CS4107
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Bioinformatics
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) Gain a thorough understanding of fundamental concepts in bioinformatics. (b) Develop problem-solving skills by implementing basic algorithms tailored for bioinformatics applications. (c) Explore various paradigms and approaches in bioinformatics as applied to biological data, such as sequence alignment, clustering, and classification. (d) Achieve proficiency in designing and implementing real-life bioinformatics projects that integrate deep learning techniques for data analysis and interpretation.
Course Description	This interdisciplinary course on bioinformatics aims to equip students with the knowledge and skills necessary to analyze and interpret biological data using computational tools and techniques. By focusing on fundamental concepts and providing hands-on experiences, students will learn to manage and analyze large-scale biological datasets. Through a combination of lectures, practical lab sessions, and collaborative projects, students will explore topics such as sequence alignment, gene expression analysis, protein structure prediction, and biological databases. Upon completion, students will be proficient in utilizing bioinformatics software and algorithms to address complex biological questions, preparing them for careers in research, biotechnology, and related fields.
Course Outline	Overview of biological databases: Protein Data Bank, SCOP, genome databases, and Cambridge Structural Database. Introduction to protein structures and biophysical methods for structure determination. Protein structure analysis, visualization techniques, and molecular modelling. Mining techniques using protein sequences and structures, including short sequence alignments and multiple sequence alignments. Phylogenetic analysis, genome context-based methods, and RNA/transcriptome analysis techniques. Mass spectrometry applications in proteome and metabolome analysis. Protein docking, dynamics simulation, and algorithms for handling big biological data challenges. Applications of Bioinformatics.
Learning Outcome	• Mastery of fundamental principles and techniques in bioinformatics, including sequence analysis, structural biology, and genomic data interpretation. • Proficiency in applying pattern recognition algorithms to solve biological data problems, such as sequence alignment, clustering, and classification. • Ability to critically analyze and interpret bioinformatics data using computational tools and techniques. • Understanding of the interdisciplinary nature of bioinformatics and its applications in biological research and medicine.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Mount, D.W., Bioinformatics: Sequence and Genome Analysis, Cold. Spring Harbor Laboratory Press, 2001.
- Protein Bioinformatics: From Sequence to Function by M. Michael Gromiha Academic Press, 2010
- Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins 4th Edition, by Andreas D. Baxevanis (Editor), Gary D. Bader (Editor), David S. Wishart (Editor), WILEY
- C. Branden and J. Tooze (eds) Introduction to Protein Structure, Garland, 1991

Course Number	**CS4108**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Time Series Analysis**
Learning Mode	Offline
Learning Objectives	• The course is designed to provide basic understanding time series analysis. • Develop Skills in statistical time series Analysis. • To learn variety of modeling techniques that can be used for time series analysis. • Gain proficiency in forecasting and anomaly detection methods • Apply the basic machine learning for time series analysis
Course Description	Using a set of fundamental techniques and broadly explains how time series analysis work at various levels of abstraction. The course introduces time series analysis with focus on applications
Course Outline	Basics of inferential and descriptive statistics: Population vs Sample; Measures of Central tendency, Measures of Variability, probability density functions, properties, mathematical expectation, hypothesis testing, ANOVA. Mathematical models for analysing time series data: Time Series Modelling, autoregressive integrated moving average (ARIMA), Exponential smoothing in time series analysis, process and the Box-Jenkins methodology. Outlier Analysis for Time Series, Multivariate Time Series Models and State-space Models, Forecasting Methods and Application Examples. Transfer Function Model Building. Imputation techniques, Point forecast and confidence intervals. Machine Learning Approaches for Time Series, Probabilistic Neural Networks, Different methods of estimation and inferences of modern dynamic stochastic general equilibrium models: simulated method of moments.
Learning Outcome	The student will be able to: • Appreciate understanding of the time series analysis, key terminology, and current industry trends in time series modeling • Evaluate time series model performance. • Create real-time applications, including anomaly detection and predictive maintenance
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Palit, Ajoy K., and Dobrivoje Popovic. *Computational intelligence in time series forecasting: theory and engineering applications*. Springer Science & Business Media, 2006.
- Box, George EP, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- Brockwell, Peter J., Richard A. Davis, and Matthew V. Calder. *Introduction to time series and forecasting*. Vol. 2. New York: springer, 2002.
- Pollock, David Stephen Geoffrey, Richard C. Green, and Truong Nguyen, eds. *Handbook of time series analysis, signal processing, and dynamics*. Elsevier, 1999.
- Shumway, Robert H., and David S. Stoffer. *Time series analysis and its applications: with R examples*. Springer, 2017.

Course Number	**CS4109**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Computational Data Analysis**
Learning Mode	Offline
Learning Objective	In this subject, the students will be trained with the knowledge of various computational techniques required for multi-dimensional data analysis such that they are able to apply these techniques in practice through programming, modeling etc.
Course Description	Modern day data is vast and diverse owing to their different acquisition systems and medium. This course aims to give an in-depth view to different data generation/acquisition mechanisms over diverse domains and the challenges incurred. It will discuss the role of computational data analysis techniques to understand and mathematically model data formation process. It will also teach them about the various data processing techniques required to manipulate and operate data to suit various objectives.
Course Outline	Understanding multi-dimensional data formation from physical acquisition devices with example cases in Remote Sensing, Geoscience, Medical sciences. Drawbacks and challenges in data acquisition, Necessity for computational modelling and analysis of data. Mathematical models for data formation and analysis, Probability models, Linear inverse optimization models, L1-L2 Regularizers, Minimizers, Cascade Modelling, Multiscale Modelling, Machine Learning models. Data Interpretation: Handling missing/corrupted data, Handling outliers, Imputation techniques, Interpolation techniques, Curve based approximation, non-convex optimization, sparse regularizers, Non-convex minimizers, Machine learning based. Data compression: Necessity, Applications, Lossless compression techniques, Lossy compression techniques, JPEG compression, Machine learning based. Statistical Models, Data preprocessing techniques in Machine learning, Signal processing techniques for multi-dimensional data, Application in various domains.
Learning Outcome	After completion of course, students will be able to • Understand data formation/generation process and the role of computational techniques in analyzing those data. • Apply the Mathematical principles behind computational techniques for data analysis. • Understand the utilities of statistical models and ML models in data analysis.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Signal Processing: A Mathematical Approach, Charles L. Byrne, Second Edition, Chapman & Hall, 2014.
- Digital Functions and Data Reconstruction: Digital-Discrete Methods, Li M Chen, Springer, 2013.
- Machine Learning with Neural Networks: An Introduction for Scientists and Engineers, Bernhard Mehlig, Cambridge University Press, 2021
- Signal Processing and Machine Learning with Applications, Michael M. Richter, Sheuli Paul, Veton Këpuska, Marius Silaghi, Springer Cham, 2022
- Data Compression: The Complete Reference, David Solomon, 4th Edition, Springer, 2007

Course Number	**CS4110**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Blockchain Technology**
Learning Mode	Offline
Learning Objectives	This course will introduce the fundamentals of the blockchain technology. It will highlight the use of blockchain technology in different applications and the learners will be able to develop decentralized applications.
Course Description	This course provides an introductory background of this revolutionary technology, followed by an interesting case study on bitcoin to demonstrate how the technology works. Following this, we would introduce Ethereum and Hyperledger. In addition, the course includes a number of hands-on sessions where we introduce basic blockchain tools and techniques, such as geth, ganache, remix, metamask, truffle, hyperledger, and real case studies.

Course Outline	Introduction and History;
	Blockchain Foundations;
	Generic elements of a blockchain; Features of blockchain; Types of blockchain;
	Applications of blockchain technology;
	Cryptocurrency and bitcoin basics;
	Introduction to Ethereum/Hyperledger and Programming;
	Privacy, Safety and Security Issues in blockchain;
	Some ongoing research topics.

Learning Outcome	• Gain proficiency in blockchain technology. • Understanding of how bitcoin/ethereum/hyperledger work. • Hands-on experience with various blockchain platforms, tools and techniques.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Bitcoin and Cryptocurrency Technologies – A Comprehensive Introduction, Princeton University Press, 2016.
- Roger Wattenhofer, The Science of the Blockchain, Inverted Forest Publishing, First Edition, 2016.
- Recent Research Papers relevant to the course.

Course Number	**CS4112**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Graph Theory**
Learning Mode	Offline
Learning Objectives	This course aims to help the students **(a)** to provide students with a comprehensive understanding of advanced topics in graph theory; **(b)** to analyze and solve complex problems involving graphs, understand and apply various graph algorithms, and explore the theoretical underpinnings of graph theory; and **(c)** develop critical thinking and analytical skills, enabling them to approach and solve real-world problems using graph theory concepts.
Course Description	This course delves into the advanced aspects of graph theory, building on foundational knowledge to explore more complex and abstract concepts. Students will study advanced topics such as graph coloring, planarity, network flows, and extremal graph theory. The course will also cover specialized graph classes, advanced algorithms, and their applications in various fields such as computer science, biology, and social sciences. Through a blend of theoretical exploration and practical application, students will gain a deep understanding of how graph theory can be used to model and solve intricate problems.
Course Outline	Introduction and Review of Basic Concepts- Review of fundamental graph theory concepts (graphs, subgraphs, isomorphism, paths, cycles), Types of graphs (simple, multigraphs, weighted graphs, directed and undirected graphs), Graph representation (adjacency matrix, adjacency list, incidence matrix) Graph Connectivity- Connectivity and components, Menger's Theorem, Network flow and cuts, Applications in network design and reliability, Vertex coloring, Edge coloring, Chromatic number and chromatic polynomial, Applications in scheduling and register allocation Euler's formula for planar graphs, Kuratowski's and Wagner's theorems, Graph embedding, Applications in geographic mapping and circuit design, Graph minors and the Minor Theorem, Well-quasi-ordering and its implications, Applications in algorithm design and complexity theory Shortest path algorithms (Dijkstra's, Bellman-Ford, Floyd-Warshall), Maximum flow algorithms (Ford-Fulkerson, Edmonds-Karp, Push-Relabel), Matching algorithms (Bipartite matching, Hungarian algorithm, Stable matching), Basic models of random graphs (Erdős–Rényi model), Properties of random graphs (connectivity, diameter, phase transition), Applications in network theory and epidemiology Eigenvalues and eigenvectors of graphs, Laplacian matrix and its properties, Cheeger's inequality and applications, Applications in clustering and data analysis, Small-world networks, Scale-free networks, Community detection algorithms, Applications in social networks, biological networks, and information networks Graph theory in computational biology (protein-protein interaction networks, metabolic networks), Graph theory in computer networks (routing, fault tolerance), Graph theory in machine learning (graph neural networks, data mining)
Learning Outcome	• Understand and articulate advanced concepts in graph theory. • Analyze and solve complex problems involving graphs. • Apply various graph algorithms to practical problems. • Understand and implement graph coloring techniques and their applications. • Analyze the properties of planar graphs and utilize graph drawing algorithms. • Apply network flow algorithms to solve problems in various fields. • Understand and apply principles of extremal graph theory, including Turán's theorem and Ramsey theory. • Identify and work with specialized graph classes such as bipartite, perfect, and chordal graphs. • Develop and implement advanced graph algorithms, including shortest path, matching, and covering algorithms. • Apply graph theory concepts to real-world scenarios in computer science, biology, and social sciences.

	- Design and conduct practical projects that demonstrate the application of advanced graph theory concepts.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term.

Suggested Reading

- Diestel, R. (2017). *Graph Theory* (5th ed.). Springer.
- Bondy, J. A., & Murty, U. S. R. (2008). *Graph Theory* (1st ed.). Springer.
- West, D. B. (2001). *Introduction to Graph Theory* (2nd ed.). Prentice Hall.
- Gross, J. L., & Yellen, J. (2005). *Graph Theory and Its Applications* (2nd ed.). CRC Press.
- Chartrand, G., & Zhang, P. (2012). *A First Course in Graph Theory* (1st ed.). Dover Publications.

Sl. No.	**Subject Code**	**SEMESTER VIII**	**L**	**T**	**P**	**C**
1.	CS42XX	DE-IV (CS ELECTIVES LIST)	3	0	0	3
2.	CS42XX	DE-V (CS ELECTIVES LIST)	3	0	0	3
3.	CS42XX	DE-VI (CS ELECTIVES LIST)	3	0	0	3
4.	CS4299	Project – II	0	0	16	8
TOTAL			**9**	**0**	**16**	**17**
GRAND TOTAL (including Semester I & II)			**167**			

Department Elective - IV						
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1.	CS4201	Multivariate Analysis	3	0	0	3
2.	CS4202	Generative AI	3	0	0	3
3.	CS4203	Statistical Machine Learning	3	0	0	3
4.	CS4204	Text Mining	3	0	0	3
5.	CS4214	Combinatorial Optimization	3	0	0	3

Course number	**CS4201**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Multivariate Analysis**
Learning Mode	offline
Learning Objectives	Multivariate analysis is about handling vector valued data. In ordinary regression modeling we are used to a vector valued predictor. But a vector valued response variable brings new issues. Sometimes we can handle a k dimensional response by treating it as k unrelated 1 dimensional problems. But often that approach will fail to find the key structure. Sometimes we are forced to study the data as an inherently k dimensional thing. It can also pay to reduce the dimension k, sometimes to 3 or 2 where plotting is available, sometimes to k=1 where ordinary methods can then be applied. Also, some of the methods are useful for exploratory work and not just for modeling responses.
Course Description	This course will provide an overview of different statistical methods applied in data science.
Course Outline	1. Multivariate Normal Distribution Theory: Joint, marginal, and conditional distribution; distributions of linear functions and quadratic forms of multivariate normal random variables 2. Correlation Analysis, Linear Regression, and Predication: Simple correlation, partial correlation, multiple correlation, linear regression equation, best prediction function and best linear predication function 3. Sampling Distributions: Sampling distributions for the mean vector and for the various correlation coefficients, partitioning of sum of squares, Hotelling's T2 distribution, the Wishart distribution 4. Introduction to Multivariate Probability Inequalities via Dependence and Heterogeneity 5. Estimation of Parameter Vectors via applications of the results on the topics in (3) and (4) above, especially for elliptical and rectangular confidence regions 6. Hypotheses Testing for Parameter Vectors 7. Multivariate Discriminant Analysis and Classification Theory, with Specific Applications to Medicine and Pattern Recognition
Learning Outcome	• Basic understanding of multivariate analysis • Problem modeling skill considering uncertainty
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Multivariate Statistical Methods: a Primer" by B.F.J. Manly.
- "Modern Applied Statistics with S" by Venables and Ripley.

Course Number	**CS4202**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Generative AI**
Learning Mode	Offline
Learning Objectives	• To provide a comprehensive understanding of advanced AI concepts with a focus on generative AI. • To design and implement various generative models such as GANs, VAEs, and Diffusion Models. • To explore the architecture and applications of Generative Pre-trained Transformers (GPT). • To design application-specific architectures for prompt engineering and multimodal generative AI. • To analyze and address ethical considerations in the development and deployment of generative AI models. • To conduct independent research and projects involving advanced generative AI techniques.
Course Description	This course provides an in-depth exploration of advanced artificial intelligence (AI) concepts, with a specific focus on generative AI (GenAI). Students will delve into advanced generative models, including Generative Adversarial Networks (GANs), Variational AutoEncoders (VAEs), Diffusion Models, and Generative Pre-trained Transformers (GPT). The course also covers the application of these models across various domains, the design of application-specific architectures for prompt engineering, and multimodal generative AI. Additionally, ethical considerations surrounding the use of generative AI will be discussed. By the end of the course, students will have the knowledge and skills to design, implement, and evaluate advanced generative AI models and understand their ethical implications.
Course Outline	Introduction to Generative AI (GenAI): Overview of GenAI, historical context and scope. Generative Adversarial Networks (GAN) and Deep Convolutional GAN (DCGAN): Understanding the architecture of GANs, Training dynamics and loss functions in GANs, Implementation and applications of DCGANs, Challenges and solutions in training GANs. Advanced Variational AutoEncoders (VAE): Fundamentals of VAEs and their architectures, Latent space representation and sampling techniques, Advanced VAE variants and their improvements, Applications of VAEs in image and data generation. Basics of Diffusion Models and Attention Mechanisms in Generative Models: Introduction to diffusion models and their principles, Understanding the role of attention mechanisms in generative models, Implementation of attention-based generative models, Case studies and applications of diffusion models. Generative Pre-trained Transformers (GPT) Basics: Overview of transformer architecture, Understanding the training and functioning of GPT models, Applications of GPT models in text generation and NLP, Fine-tuning and optimizing GPT for specific tasks. Application-Specific Architecture for Prompt Engineering and Multimodality: Designing and optimizing prompt engineering techniques, Exploring multimodal generative models, Integrating text, image, and audio in generative models, Case studies of application-specific generative architectures. Ethical Considerations in Generative AI: Understanding the ethical implications of Generative AI, Addressing bias, fairness, and accountability in generative models, Privacy concerns and data security in Generative AI.
Learning Outcome	By the end of this course, students will be able to: • Understand the foundational concepts and the latest advancements in artificial intelligence and generative AI. • Design and implement Generative Adversarial Networks (GANs) and their advanced variants, such as DCGAN. • Develop and apply advanced Variational AutoEncoders (VAEs) for generative tasks. • Grasp the basics of Diffusion Models and the role of attention mechanisms in enhancing generative models. • Understand the architecture and functioning of Generative Pre-trained Transformers (GPT) and their applications.

	• Create application-specific architectures for prompt engineering and explore the integration of multimodal generative AI techniques. • Analyze and address ethical considerations and challenges in the development and deployment of generative AI models. • Conduct independent research and projects involving advanced generative AI techniques, demonstrating a comprehensive understanding of both theoretical and practical aspects.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Foster, D. (2022). Generative deep learning: Teaching Machines to Paint, Write, Compose, and Play. O'Reilly Media, Inc.
- Valle, R. (2019). Hands-On Generative Adversarial Networks with Keras: Your guide to implementing next-generation generative adversarial networks. Packt Publishing Ltd.
- Research Papers and Articles from Journals such as JMLR, IEEE Transactions on Neural Networks and Learning Systems, etc., and Conference Proceedings from NeurIPS, ICML, and CVPR,etc.

Course Number	**CS4203**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Statistical Machine Learning**
Learning Mode	Offline
Learning Objectives	The learning objectives of the course includes understanding the basic concepts of machine learning, and classic algorithms such as Support Vector Machines and Neural Networks, Deep Learning. The students would be able to explain the basic principles and theory of machine learning, that would guide to invent their own algorithms.
Course Description	This is an introductory course on statistical machine learning which presents an overview of many fundamental concepts, popular techniques, and algorithms in statistical machine learning. It covers basic topics such as dimensionality reduction, linear classification and regression as well as more recent topics such as ensemble learning/boosting, support vector machines, kernel methods and manifold learning. This course will provide the students the basic ideas and intuition behind modern statistical machine learning methods. After studying this course, students will understand how, why, and when machine learning works on practical problems.
Course Outline	Statistical Theory: Maximum likelihood, Bayes, minimax, parametric versus nonparametric methods, Bayesian versus Non-Bayesian approaches, classification, regression, density estimation. Convexity and Optimization: Convexity, conjugate functions, unconstrained and constrained optimization, KKT conditions. Parametric Methods: Linear regression, model selection, generalized linear models, mixture models, classification, graphical models, structured prediction, hidden Markov models Sparsity: High dimensional data and the role of sparsity, techniques for handling sparsity. Nonparametric Methods: Nonparametric regression and density estimation, nonparametric classification, clustering and dimension reduction, manifold methods, spectral methods, the bootstrap and subsampling, nonparametric Bayes. Other Learning Methods: Semi-supervised learning, reinforcement learning, minimum description length, online learning, the PAC model, active learning
Learning Outcome	On successful completion of this course students will be able to: • Explain the basic concepts of machine learning, and classic algorithms such as Support Vector Machines and Neural Networks, Deep Learning. • Explain the basic principles and theory of machine learning, which may guide students to invent their own algorithms in future. • Ability to program the algorithms in the course. • Ability to do mathematical derivation of the machine learning algorithms.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

- Chris Bishop, Pattern Recognition and Machine Learning, Springer, Information Science and Statistics Series, 2006.
- Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Texts in Statistics, SpringerVerlag, New York, 2001.
- Larry Wasserman, All of Statistics: A Concise Course in Statistical Inference, Springer Texts in Statistics, Springer-Verlag, New York, 2004.

Course Number	**CS4204**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Text Mining**
Learning Mode	Offline
Learning Objectives	• To understand the fundamental principles and key concepts in text mining. • To gain the ability to collect and preprocess text data, including cleaning and integration. • To master text preprocessing techniques such as tokenization, stemming, stopword removal, and normalization. • To learn the construction and utilization of knowledge graphs for relationship extraction. • To implement frequent pattern mining and association rules using algorithms like apriori. • To extract features using methods like Bag-of-Words, TF-IDF, and word embeddings. • To apply clustering and classification techniques to text data. • To utilize text mining techniques in practical applications, such as sentiment analysis.
Course Description	This course provides a comprehensive understanding of the fundamental principles and techniques used in text mining. Students will learn the entire process from data collection and preprocessing to advanced techniques for mining patterns and analyzing text. The course covers practical applications, such as sentiment analysis, equipping students with the skills needed to extract meaningful insights from large datasets and text corpora. By the end of the course, students will be adept at employing text mining techniques to solve real-world problems.
Course Outline	Text mining introduction: Overview, motivation, challenges and opportunities, Data Collection and Pre-processing: Techniques for collecting data from various sources Data cleaning and integration: Handling noise, missing values, and inconsistent formats in text data Text preprocessing: tokenization, stemming, stopword removal, and normalization Knowledge graph construction: Basics of graph construction and relationship extraction Basic concepts of frequent patterns, association rules, mining frequent patterns: apriori algorithm. Feature extraction, Bag-of-Words, TF-IDF, word embeddings Clustering and classifying text data Some applications: sentiment analysis, etc.
Learning Outcome	By the end of this course, students will be able to: • Grasp key concepts, motivation, and challenges in text mining. • Collect and preprocess data, including cleaning and integration. • Perform text preprocessing tasks like tokenization, stemming, stopword removal, and normalization. • Construct and utilize knowledge graphs for relationship extraction. • Implement frequent pattern mining and association rules using the apriori algorithm. • Extract features using Bag-of-Words, TF-IDF, and word embeddings. • Apply clustering and classification to text data. • Use data mining and text analytics techniques in applications such as sentiment analysis.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Srivastava, A. N., & Sahami, M. (Eds.). (2009). Text mining: Classification, clustering, and applications. CRC press.
- Jiawei, H., & Micheline, K. (2006). Data mining: concepts and techniques. Morgan kaufmann.
- Witten, I. H., Frank, E., Hall, M. A., Pal, C. J., & Data, M. (2005, June). Practical machine learning tools and techniques. In Data mining (Vol. 2, No. 4, pp. 403-413). Amsterdam, The Netherlands: Elsevier.

Course number	**CS4214**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Combinatorial Optimization**
Learning Mode	offline
Learning Objectives	Introduces the use of combinatorial and algorithmic techniques for optimization problems. Emphasis on algorithms with provable correctness and efficiency. Illustrates the use of these techniques for several real-world applications.
Course Description	This course will provide an overview of different methods of solving computationally intractable problems.
Course Outline	Linear and Integer Programs: Overview of modeling problems in linear and integer programming model Geometry of Polyhedra: Feasible region of LPs and polyhedra, Convexity, Extreme points, Faces and facets, tools for analysis Solving linear programs: Possible outcomes (infeasibility, unboundedness, and optimality) and their certificates, bases and canonical forms, the Simplex method Duality: Weak duality, strong duality, complementary slackness, Farkas' Lemma Combinatorial Optimization: Primal-dual method for exact and approximation algorithms, Shortest paths, Minimum cost perfect matchings, Max-Flow Min-Cut Theorem, Totally Unimodular Matrices Additional topics: Interior-point methods, Randomized/Online algorithms for LPs, Integer Programs, Convex Optimization, Matroids, T-joins, Applications to Game Theory.
Learning Outcome	• Basic understanding of combinatorial optimization • Problem modeling skill • Solving computationally intractable problems
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term.

Suggested Reading:

- Papadimitriou, C. H., and K. Steiglitz. *Combinatorial Optimization*.
- Lovasz, Laszlo. *Matching Theory*.
- Ahuja, R., T. Magnanti, and J. Orlin. *Network Flows*.
- Schrijver, A. *Theory of Linear and Integer Programming*.
- Chvatal, V. *Linear Programming*.
- Bertsimas, D., and J. Tsitsiklis. *Linear Optimization.*
- Cook, W. J., W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*.

Department Elective - V						
Sl. No.	Course Code	Course Name	L	T	P	C
1.	CS4205	Cloud Computing	3	0	0	3
2.	CS4206	Quantum Computing	3	0	0	3
3.	CS4207	Drone Data Processing	3	0	0	3
4.	CS4208	Edge Computing	3	0	0	3
5.	CS4209	Wireless Networks	3	0	0	3
6.	CS4215	Distributed Computing	3	0	0	3
7.	CS4216	Parallel Computing	3	0	0	3

Course Number	**CS4205**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Cloud Computing**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) define and explain the fundamental concepts and principles of cloud computing; (b) identify and describe various cloud computing service models (IaaS, PaaS, SaaS) and deployment models (public, private, hybrid, community); (c) understand the underlying technologies and infrastructure used in cloud computing, including virtualization, containers, and software-defined networking; (d) evaluate the benefits and challenges of adopting cloud computing for businesses and organizations; (e) design and implement cloud-based solutions for common use cases, such as web hosting, data storage, and application development; and (f) analyze security, privacy, and compliance considerations in cloud computing environments.
Course Description	This course provides a comprehensive overview of cloud computing, covering its fundamental concepts, architecture, and deployment models. Students will explore the various service models, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), and understand the benefits and challenges associated with each. The course delves into cloud storage, computing resources, and virtualization technologies, offering hands-on experience with leading cloud platforms such as AWS, Azure, and Google Cloud. Security, compliance, and cost management in cloud environments are also addressed, equipping students with the skills to design, deploy, and manage cloud-based solutions effectively.
Course Outline	Introduction to Cloud Computing- Overview of cloud computing and its key principles, Fundamentals of distributed systems: Models and architectures. Cloud Storage and Virtualization- Understanding cloud storage technologies: Key-value stores, NoSQL databases, Virtualization techniques for resource abstraction and management Distributed Algorithms in Cloud Computing- Fault tolerance and consensus algorithms: PAXOS, leader election, Time ordering and distributed mutual exclusion. Industry Systems and Cloud Platforms- Overview of industry-standard cloud platforms: Apache Spark, Apache Zookeeper, HBase, Introduction to containerization technologies: Docker, Kubernetes Advanced Topics in Cloud Computing- Big data processing in the cloud: MapReduce, Apache Cassandra, Emerging trends in cloud computing: Edge computing, serverless architectures
Learning Outcome	• Define and explain the key concepts and components of cloud computing, including virtualization, elasticity, and on-demand provisioning. • Evaluate different cloud computing service models and deployment models, and select appropriate options for specific use cases and requirements. • Demonstrate proficiency in deploying and managing cloud-based resources using popular cloud platforms (e.g., AWS, Azure, Google Cloud). • Analyze the economic factors and cost considerations associated with cloud computing, including pricing models and Total Cost of Ownership (TCO) calculations. • Design and implement scalable and resilient cloud architectures using best practices and design patterns. • Assess security risks and implement appropriate security controls to protect cloud-based assets and data.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Distributed and Cloud Computing From Parallel Processing to the Internet of Things; Kai Hwang, Jack Dongarra, Geoffrey Fox Publisher: Morgan Kaufmann, Elsevier, 2013.
- Cloud Computing: Principles and Paradigms; Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski Publisher: Wiley, 2011.
- Distributed Algorithms Nancy Lynch Publisher: Morgan Kaufmann, Elsevier, 1996.
- Cloud Computing Bible Barrie Sosinsky Publisher: Wiley, 2011.
- Cloud Computing: Principles, Systems and Applications, Nikos Antonopoulos, Lee Gillam Publisher: Springer, 2012.

Course Number	**CS4206**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Quantum Computing**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) comprehend the foundational principles of quantum mechanics that underpin quantum computing.(b) proficiency in designing and analyzing quantum circuits.(c) explore and understand advanced quantum algorithms used in quantum computing.(d) quantum computing principles to solve computational problems and simulate quantum systems.
Course Description	Explore the foundational principles and transformative potential of quantum mechanics and quantum computing in this comprehensive course. Students will delve into quantum mechanics, covering concepts like superposition, entanglement, and quantum measurement, and their application to quantum computing. Through lectures, practical sessions, and case studies, participants will master quantum circuit design, analyze advanced quantum algorithms such as Grover's and Shor's algorithms, and apply these principles to solve real-world computational problems. By the end of the course, students will possess the theoretical understanding and practical skills needed to contribute to the rapidly advancing field of quantum computing across diverse industries.
Course Outline	States, Wavefunction, Orthogonality and Orthonormality of Wave function, Superposition Quantum Circuits: Single-qubit gates, Multiple qubit gates, Design of quantum circuits, Dirac Notations, Measurements, Bloch Sphere Entanglement, Bell State, Teleportation, Q-Sphere, Data Structures for Quantum Computing, Quantum Annealing Quantum Algorithms: Grover’s Search Algorithm, Shor’s Factoring Algorithm, Quantum Amplitude Estimation, Quantum Phase Estimation, Quantum Fourier Transform
Learning Outcome	• Understand Fundamental Quantum Mechanics Principles. • Develop Skills in Quantum Circuit Computing and Analysis. • Explore Advanced Quantum Computing Concepts. • Gain proficiency in Master Quantum Algorithms.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Textbooks:

- Nielsen, M.A. and Chuang, I.L., 2010. Quantum computation and quantum information.
- Pittenger, A.O., 2012. An introduction to quantum computing algorithms (Vol. 19).
- Relevant research articles.

Reference books:

- Bernhardt, C., 2019. *Quantum computing for everyone*.

Course Number	**CS4207**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Drone Data Processing**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) gain foundational knowledge of unmanned aerial systems (UAS), including their history, components, and classifications; (b) comprehend the various elements that make up a drone system, such as the air vehicle, communication data links, command and control elements, payloads, and launch/recovery systems; (c) acquire the ability to design and plan drone missions, including studying area maps, designing flight routes, and calibrating sensors; (d) learn the principles and practices of photogrammetry and geographic information systems (GIS) for processing and analyzing drone-collected data; (e) understand the importance of data quality, accuracy standards, error estimation, and strategies for achieving high-precision geospatial data.
Course Description	This course offers an in-depth exploration of Unmanned Aerial Systems (UAS) and drone operations, providing a comprehensive understanding of their history, types, and technological advancements. Students will learn about the various categories and missions of drones, the design and communication systems essential for drone functionality, and the roles and responsibilities in UAS operations. The course covers the fundamentals of geospatial data, photogrammetry, and GIS, emphasizing map accuracy and mission planning.
Course Outline	Introduction to Unmanned Aerial Systems (UAS). Types, Categories, and Missions of Drones. Drone Design and Communication Systems. Concepts of Operations (CONOP) and Risk Assessment Geospatial Data and Photogrammetry. Drone Mission Planning and Control. Route Planning and Operational Fundamentals. Regulatory Requirements and Guidelines. Applications and Challenges in Drone Operations
Learning Outcome	• Identify and categorize various types of unmanned aerial systems and their specific missions. • Create comprehensive mission plans, including route design, sensor selection, and calibration, ensuring optimal data collection. • Utilize photogrammetric methods and GIS tools to process and analyze drone-collected data, producing accurate geospatial products. • Assess data accuracy and quality, understand and apply mapping standards, and manage errors in measurements effectively. • Apply drone technology in diverse fields such as agriculture, construction, environmental monitoring, and disaster response.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Barnhart, R., Michael, M., Marshall, D., and Shappee, E. ed. 2016. Introduction to Unmanned Aircraft Systems, 2nd edition. Boca Raton. CRC Press.
- Fahlstrom, P. and Gleason, T. 2012. Introduction to UAV Systems. 4th edition. United Kingdom. John Wiley & Sons Ltd.
- Wolf, P., DeWitt, B., and Wilkinson, B. 2014. Elements of Photogrammetry with Applications in GIS, 4th edition. McGraw-Hil
- Introduction to UAV Systems, Paul G. Fahlstrom and Thomas J. Gleason
- Drone Technology in Architecture, Engineering, and Construction, Daniel Tal and Jon Altschuld
- UAV or Drones for Remote Sensing Applications, edited by Felipe Gonzalez Toro and Antonios Tsourdos

Course Number	**CS4208**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Edge Computing**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) to define edge computing and its role in modern computing paradigms; (b) to understand the principles and benefits of moving computation closer to the data source; (c) to identify various edge computing architectures, including fog computing and mobile edge computing; (d) to analyze and compare edge computing frameworks and platforms; (e) design and implement edge computing solutions to address latency, bandwidth, and privacy concerns; (f) to evaluate the impact of edge computing on traditional cloud computing models and network infrastructures; (g) to discuss emerging trends and challenges in edge computing, such as security, interoperability, and resource management; and (h) to apply edge computing principles and techniques to real-world scenarios and use cases.
Course Description	This course provides a comprehensive overview of edge computing, starting with the limitations of cloud computing in supporting low latency and round trip time (RTT), and the subsequent innovation waves leading to edge computing. Students will delve into edge computing architectures and their applications, including 5G slicing and self-driving cars. Key concepts of distributed systems such as time ordering, clock synchronization, and distributed snapshots will be explored within the context of edge computing. The course also introduces edge data centers, lightweight edge clouds, and services provided by various service providers. Practical knowledge of Docker containers and Kubernetes in edge computing, along with the design of edge storage systems like key-value stores, will be covered. Additionally, students will learn about MQTT and Kafka for creating end-to-end edge pipelines and edge analytics topologies for M2M and WSN networks. The course concludes with use cases of machine learning for edge sensor data, including predictive maintenance, image classification, and deep learning on-device inference to support latency-sensitive applications.
Course Outline	Introduction to Cloud and its limitations to support low latency and Round Trip Time (RTT). From Cloud to Edge computing: Waves of innovation. Introduction to Edge Computing Architectures. Edge Computing to support User Applications (5G-Slicing, self-driving cars and more) Concepts of distributed systems in edge computing such as time ordering and clock synchronization, distributed snapshot, etc. Introduction to Edge Data Center, Lightweight Edge Clouds and its services provided by different service providers. Introduction to docker container and Kubernetes in edge computing. Design of edge storage systems like key-value stores. Introduction to MQTT and Kafka for end-to-end edge pipeline. Edge analytics topologies for M2M and WSN network (MQTT) Use cases of machine learning for edge sensor data in predictive maintenance, image classifier and self-driving cars. Deep Learning On-Device inference at the edge to support latency-based application
Learning Outcome	• Define and explain the concept of edge computing and its significance in distributed computing architectures. • Analyze the advantages and limitations of edge computing compared to traditional centralized and cloud-based approaches. • Identify and describe different edge computing architectures, such as hierarchical, decentralized, and hybrid models. • Evaluate edge computing platforms and tools for their suitability in various application domains. • Design and implement edge computing solutions that leverage distributed computing principles to improve performance, reliability, and efficiency. • Analyze the impact of edge computing on network traffic, data privacy, and regulatory compliance. • Critically assess the security implications of deploying edge computing systems and propose mitigation strategies.

	• Collaborate in teams to develop and present case studies or projects demonstrating the practical application of edge computing concepts and techniques.
Assessment ethod	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Fog and Edge Computing: Principles and Paradigms, Rajkumar Buyya (Editor), Satish Narayana Srirama (Editor), Wiley, 2019
- Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley, 2011
- Cloud and Distributed Computing: Algorithms and Systems, Rajiv Misra, Yashwant Patel, Wiley 2020.
- Journal papers as references.

Course Number	**CS4209**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Wireless Networks**
Learning Mode	Offline
Learning Objectives	In this subject, the students will be trained with the knowledge of 802.11 wireless networks, including protocol knowledge and the associated security vulnerabilities.
Course Description	In the consumer, industrial, and military sectors, 802.11-based wireless access networks have been widely used due to their convenience. This application, however, is reliant on the unstated assumptions of availability and anonymity. The management and media access protocols of 802.11 may be particularly vulnerable to malicious denial-of-service (DoS) and various security attacks. This course analyzes these 802.11-specific attacks, including their applicability, effectiveness, and proposed low-cost implementation improvements to mitigate the underlying vulnerabilities.
Course Outline	Overview of 802.11 networks, 802.11 MAC Layer, Wireless LAN physical components. Wireless LAN topologies and technologies - 802.11 a/b/g/n/ac features. Configure and install wireless adapters, access points. 802.11 architecture (access points, SSID, channels, beacons, scanning, association), Hidden terminal problem, RTS/CTS, 802.11 CSMA-CA protocol. Wireless communication technology: FHSS, DSSS, CDMA etc. Physical Layer, MAC Layer, MAC Management, Power Management. Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. 802.11 Frame Structure & WLAN services-association, disassociation, re-association, distribution, integration, authentication, de-authentication and data delivery services. Security Features of 802.11: WEP, WPA1, and WPA2, PSK Authentication, TKIP Encryption and AES-CCMP Encryption.
Learning Outcome	On successful completion of the course, students should be able to: • Understand the fundamentals of 802.11 wireless networks • Describe the WLAN services-association, disassociation, re-association, distribution, integration, authentication, de authentication and data delivery services • Comprehend the vulnerabilities associated with 802.11 protocol.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Text Books and References:

1. "Wireless Communications: Principles and Practice" by Theodore S. Rappaport (2nd Edition)
2. "802.11 Wireless Networks: The Definitive Guide" by Matthew S. Gast (2nd Edition)
3. "Wireless Communications & Networks" by William Stallings (2nd Edition)
4. "Wireless Communications: Principles and Practice" by Andreas F. Molisch (2nd Edition)
5. "Fundamentals of Wireless Communication" by David Tse and Pramod Viswanath (1st Edition)
6. "Next Generation Wireless LANs: 802.11n and 802.11ac" by Eldad Perahia and Robert Stacey (2nd Edition)
7. "Wireless Networking: Understanding Internetworking Challenges" by Anurag Kumar, D. Manjunath, and Joy Kuri 1st Edition)
8. "Wireless Communications: Principles and Practice" by Kaveh Pahlavan and Prashant Krishnamurthy (1st Edition)

Course Number	**CS4215**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Distributed Computing**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) understand the fundamental concepts and challenges of distributed computing; (b) learn various communication models and protocols used in distributed computing; (c) explore distributed system architectures and design principles; (d) gain knowledge of distributed coordination algorithms and techniques; (e) study fault tolerance mechanisms and reliability in distributed computing; (f) investigate scalability issues and techniques for scaling distribute computing.; (g) understand security concerns and mechanisms in distributed environments; (h) analyze case studies of real-world distributed computing; and (i) explore emerging trends and technologies in distributed computing.
Course Description	This course offers a thorough exploration of distributed computing, focusing on the fundamental concepts, architectures, and algorithms that enable multiple computers to work together seamlessly. Students will study the principles of distributed systems, including communication, synchronization, fault tolerance, and consistency. Key topics include distributed databases, cloud computing, distributed file systems, and parallel processing. The course also covers contemporary technologies and frameworks such as Hadoop, Spark, and Kubernetes. Through practical assignments and projects, students will gain hands-on experience in designing, implementing, and managing distributed systems, preparing them to tackle complex problems in a distributed computing environment.
Course Outline	Introduction to Distributed Computing: Overview of distributed systems, Characteristics and challenges, Examples of distributed systems. Distributed System Models and Architectures: Client-server architecture, Peer-to-peer (P2P) architecture, Middleware-based architectures, Distributed object-based systems Communication in Distributed Systems: Message passing, Remote procedure calls (RPC), Message-oriented middleware (MOM), Publish-subscribe systems. Distributed Algorithms: Mutual exclusion, Leader election, Distributed consensus, Distributed transactions Distributed Data Management: Replication and consistency, Distributed file systems (e.g., HDFS), Distributed databases (e.g., Cassandra, MongoDB). Fault Tolerance and Reliability: Fault models, Failure detection and recovery, Replication and redundancy, Byzantine fault tolerance Scalability and Performance: Scalability patterns, Load balancing, Caching strategies, Performance measurement and optimization. Case Studies and Advanced Topics: Google File System (GFS), MapReduce, Apache Kafka, Docker and container orchestration (e.g., Kubernetes)
Learning Outcome	• Students will demonstrate a deep understanding of the fundamental concepts, principles, and challenges of distributed computing. • Students will be able to analyze, design, and implement distributed computing solutions to address specific problems or requirements. • Students will critically evaluate different distributed system architectures, algorithms, and technologies, considering their strengths, weaknesses, and trade-offs. • Students will work collaboratively in teams to design and implement distributed system projects, demonstrating effective teamwork and project management skills. • Students will apply distributed systems principles and techniques to solve real-world problems, demonstrating the relevance and applicability of course concepts. • Students will be able to adapt to new developments and emerging trends in distributed computing, demonstrating a continuous learning mindset. • Students will reflect on their learning experiences throughout the course, identifying areas of growth, challenges encountered, and lessons learned for future application.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term.

Suggested Reading

- Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services, Brendon Burns
- Distributed Systems: Principles and Paradigms, Andrew S. Tanenbaum and Maarten Van Steen
- Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systemsm, Martin Kleppmann
- Distributed Algorithms, Nancy Lynch
- Distributed Systems: Concepts and Design, George Coulouris, Jean Dollimore, and Tim Kindberg
- Distributed Systems for Fun and Profit, Mikito Takada

Course number	**CS4216**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Parallel computing**
Learning Mode	offline
Learning Objectives	Fundamental theoretical issues in designing parallel algorithms and architectures. Shared memory models of parallel computation.
Course Description	This course will provide an overview of different methods and tricks to solve problems in a multiprocessor environment.
Course Outline	**Parallel Programming Models:** Shared-memory model (PRAM, MIMD, SIMD), network model (line, ring, mesh, hypercube), performance measurement of parallel algorithms. **Algorithm Design Techniques for PRAM Models:** Balancing, divide and conquer, parallel prefix computation, pointer jumping, symmetry breaking, pipelining, accelerated cascading. **Algorithms for PRAM Models:** List ranking, sorting and searching, tree algorithms, graph algorithms. **Parallel Complexity:** Lower bounds for PRAM models, the complexity class NC, P-completeness.
Learning Outcome	• An understanding of computer architectures at a high level, in order to understand what can and cannot be done in parallel, and the relative costs of operations like arithmetic, moving data, etc., • To recognize programming "patterns" to use the best available algorithms and software to implement them, • To understand sources of parallelism and locality in simulation in designing fast algorithms.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term.

Suggested Reading:

- Michael J Quinn, *Parallel Computing: Theory and Practice*, second edition, McGraw Hill, 1994/2002.
- Michael J Quinn, *Parallel Programming in C with MPI and OpenMP*, first edition, McGraw Hill, 2004/2003.
- Ananth Grama, Anshul Gupta, George Karypis and Vipin Kumar, *Introduction to Parallel Computing*, second edition, Addison-Wesley/Pearson, 1994/2003.
- Russ Miller and Laurence Boxer, *Algorithms: Sequential and Parallel — A Unified Approach*, third edition, Cengage, 2013.
- Fayez Gebali, *Algorithms and Parallel Computing*, Wiley, 2011.
- Seyed H Roosta, *Parallel Processing and Parallel Algorithms: Theory and Computation*, Springer, 2000.

Department Elective - VI						
Sl. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1.	CS4210	Computer Security	3	0	0	3
2.	CS4211	Cryptography	3	0	0	3
3.	CS4212	Big Data Analytics	3	0	0	3
4.	CS4213	Computer Forensics	3	0	0	3

Course Number	**CS4210**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Computer Security**
Learning Mode	Offline
Learning Objectives	To have a clear understanding of security and privacy issues in various aspects of computing, including: Programs, Operating systems & Networks
Course Description	The course covers. security and privacy issues in various aspects of computing, including: Programs, Operating systems, Networks, Web Applications
Course Outline	Introduction to Computer Security and Privacy: security and privacy; types of threats and attacks; methods of defense Program Security: nonmalicious program errors; vulnerabilities in code, Secure programs; malicious code; Malware detection Operating System Security: Methods of protection; access control; user authentication Network Security: Network threats; firewalls, intrusion detection systems Application Security and Privacy: Basics of cryptography; security and privacy for Internet applications, IPSEC, TLS
Learning Outcome	After completion of this course a student will have • Understanding of security issues in computing at program, , • Understand the operations of different malware • The ability to analysis Malwares • Ability to analyse the security of Operating system and Networks
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

1. Security in Computing, Charles P. Pfleeger and Shari Lawrence Pfleeger, 4th edition or later Prentice-Hall, 2007
2. *Computer Security*: Principles and Practice, Dr. William Stallings and Lawrie Brown, Pearson

Course Number	**CS4211**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Cryptography**
Learning Mode	physical
Learning Objectives	To have a clear understanding of design and analysis of different cryptographic primitives
Course Description	The course covers design and analysis of different cryptographic primitives including Symmetric and asymmetric key cryptography
Course Outline	Mathematical Background: Modular Arithmetic, Finite Fields, The Group Law, Elliptic Curves over Finite Fields , Projective Coordinates. Symmetric Encryption: Shift Cipher, Substitution Cipher, Permutation Cipher, Stream Cipher Basics, Linear Feedback Shift Registers, RC4; Block Ciphers: DES, AES, and Different modes of Block ciphers. Key Management, Secret Key Distribution. Hash Functions and Message Authentication Codes: SHA, MD5, HMAC. Public Key Encryption: RSA, ElGamal Encryption, Rabin Encryption, Elliptic curve based encryption. Digital Signatures: RSA based, DSA, ECDSA. Public key based infra structure. Key Exchange: Diffie–Hellman Key Exchange, Authenticated Key Agreement
Learning Outcome	After completion of this course a student will have • Understanding of modular arithmetic and Finite fields, • Understanding and analysis of symmetric key cryptography DES, AES • Understanding and analysis of Hash function, MAC function, • Understanding and analysis of asymmetric key cryptography • Understanding and analysis of key agreement protocols
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- W. Mao, Modern Cryptography: Theory and Practice. Pearson Education
- Hand book of applied cryptography by A. Menezes, CRC press
- Doug Stinson, Cryptography: Theory and Practice, Chapman and Hall/CRC,

Course Number	**CS4212**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Big Data Analytics**
Learning Mode	Offline
Learning Objectives	The objective of this course is to provide students **(a)** with a comprehensive understanding of Big Data analytics, covering the challenges, applications, and technologies involved in managing and analyzing large-scale data; **(b)** about the Big Data stack, various Big Data platforms (such as Apache Spark, HDFS, and YARN), and the MapReduce programming model; **(c)** knowledge of Big Data storage platforms, streaming platforms, and machine learning algorithms in Spark, including an introduction to deep learning for Big Data; **(d)** information about Big Data applications in graph processing.
Course Description	This comprehensive course provides an in-depth overview of big data and its significant impact across various industries. Students will explore the foundational characteristics of big data, including Volume, Velocity, Variety, Veracity, and Value, and understand the distinctions between big data and traditional data.
Course Outline	Introduction to Big Data: Overview of big data and its characteristics (Volume, Velocity, Variety, Veracity, Value), Big data vs. traditional data, Introduction to big data technologies and tools, Applications of big data in various industries. Big Data Architecture, Components of big data architecture, Distributed computing and storage, Introduction to Hadoop ecosystem (HDFS, YARN, MapReduce), Overview of other big data platforms (Spark, Flink, Storm) Data Ingestion and Storage, Data ingestion techniques and tools (Flume, Kafka, Sqoop), NoSQL databases (HBase, Cassandra, MongoDB), Data warehousing solutions (Hive, HBase), Real-time data processing. Data Processing with Hadoop, Hadoop Distributed File System (HDFS), MapReduce programming model, Writing and executing MapReduce jobs, Data processing workflows with Apache Pig. Data Processing with Apache Spark, Introduction to Apache Spark, Spark Core and RDDs (Resilient Distributed Datasets), Spark SQL and DataFrames, Spark Streaming for real-time data processing Data Analysis and Visualization, Exploratory Data Analysis (EDA) techniques, Data visualization tools (Tableau, Power BI, D3.js), Creating dashboards and reports, Visualizing big data with Python (Matplotlib, Seaborn). Applying machine learning algorithms to big data (classification, regression, clustering), MLlib: Spark's machine learning library, Time-series analysis and forecasting, Text mining and sentiment analysis, Graph analytics with big data, Recommender systems Overview of cloud platforms for big data (AWS, Azure, Google Cloud), Cloud-based big data services and tools, Deploying big data applications in the cloud, Scalability and performance optimization. Security and Privacy in Big Data, Data privacy and security challenges in big data, Data anonymization and encryption techniques, Regulatory and compliance considerations (GDPR, CCPA), Best practices for securing big data, Real-world big data applications in healthcare, finance, marketing, and IoT.
Learning Outcome	• Comprehend the introduction, challenges, and applications of Big Data. • Understand the components and distribution packages of the Big Data stack. • Work with Apache Spark, HDFS, YARN, and implement the MapReduce programming model. • Manage Big Data Storage • Apply Machine Learning in Big Data • Explore Big Data Applications in Graph Processing • Understand and utilize Pregel, Giraph, and Spark GraphX for graph processing.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Bart Baesens, Analytics in a Big Data World: The Essential Guide to Data Science and its Applications, Wiley, 2014
- Dirk Deroos et al., Hadoop for Dummies, Dreamtech Press, 2014.
- Chuck Lam, Hadoop in Action, December, 2010
- Mining of Massive Datasets. Leskovec, Rajaraman, Ullman, Cambridge University Press
- Data Mining: Practical Machine learning tools and techniques, by I.H. Witten and E. Frank

- Erik Brynjolfsson et al., The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies, W. W. Norton & Company, 2014

Course Number	**CS4213**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Computer Forensics**
Learning Mode	Offline/online
Learning Objectives	This course aims to: • impart principles and techniques for digital forensics investigation • make aware of various digital forensics tools • guide one how to perform forensics procedures to ensure court admissibility of evidence, as well as the legal and ethical implications
Course Description	Digital forensics involves the investigation of computer-related crimes with the goal of obtaining evidence to be presented in a court of law. In this course, students will learn the principles and techniques for digital forensics investigation and the spectrum of available computer forensics tools. One will learn about core forensics procedures to ensure court admissibility of evidence, as well as the legal and ethical implications. One will learn how to perform a forensic investigation on both Unix/Linux and Windows systems with different file systems. One will also be guided through forensic procedures and review and analyze forensics reports. Although the course does not have any lab components but students may have to work out some assignments/case project works related to data analysis and data recovery, data acquisition, recovering graphics file, validation of a forensic image file, etc.
Course Outline	**Digital Forensics Fundamentals: Overview,** Preparation for Digital Forensics, Conducting Investigation, Understanding Forensics Lab requirements, Cyber Laws Data Acquisition: Understanding the storage formats, Determining acquisition method, Use of acquisition tools, Validating data acquisition Processing crime and incident scenes: Identifying digital evidence, preparing for a search, Seizing and storing Digital Evidence Working with Windows and Linux File Systems: Understanding File Systems, Exploring Microsoft File Structure, Examining NTFS Disks, Windows Registry, Virtual Machine, File structure in Ext4, Some Forensics Tools: Software Tools, Hardware Tool, Validating and Testing Forensics Software, Password protection, Password Recovery Tools Recovering Graphics Files: Recognizing Graphics File, Understanding Data Compression, Identifying Unknown File Formats, Understanding Copyright Issues with Graphics Digital Forensics Analysis and Validation: Determining what data to collect and analyze, Validating Forensics Data, Addressing Data Hiding Techniques, Forensics handwriting and signature analysis Overview Email and Social Media Investigations, Mobile Device Forensics, Cloud Forensics, Memory Forensics
Learning Outcome	Upon successful completion of this course, the students will: • be able to perform forensics analysis using digital evidence • gain exposure on analyzing the performance of various forensics tools • obtain more in depth knowledge on various file system related artifacts
Assessment Methods	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Amelia Phillips, Bill Nelson, Christopher Steuart - “Guide to Computer Forensics and Investigations”, 6th Editon, Cengage
- Darren Hayes: Practical Guide to Digital Forensics Investigations, Pearson

- Michael K. Robinson: Digital Forensics: Hands-on Activities in Digital Forensics, Createspace Independent Pub; Workbook edition
- Gerard Johnsen, Digital Forensics and Incident Response: Incident response tools and techniques for effective cyber threat response, 3rd Edition, 2022
- William Oettinger, Learn Computer Forensics: Your one-stop guide to searching, analyzing, acquiring, and securing digital evidence, 2nd Edition, 2022
- Thomas J. Holt, Adam M. Bossler, Kathryn C. Seigfried-Spellar, Cybercrime and Digital Forensics: An Introduction, 3rd Edition, 2022

IDE CSE (Available to students other than Dept. of CSE)

IDE	Semester	Course Code	Course Name	L	T	P	C
IDE-I	Semester-4	CS2207	Introduction to Data Science	3	0	0	3
IDE-II	Semester-5	CS3106	Computer Graphics	3	0	0	3
IDE-III	Semester-7	CS4113	Data Analysis and Visualization	3	0	0	3

Course Number	**CS2207 (IDE-1)**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Introduction to Data Science**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) to understand the fundamental concepts and principles of data science. (b) to provide an understanding of the data science process, including data collection, cleaning, analysis, and interpretation (c) to develop understanding in statistical and machine learning techniques for data analysis (d) to conduct exploratory data analysis (EDA) and to create predictive models (e) in developing of problem-solving skills using data science methodologies (f) to develop skills in visualizing data and creating compelling data stories (g) to highlight the importance of ethical decision-making in data science projects endeavors.
Course Description	This academic course on Introduction to Data Science aims to introduce methods for data collection and cleaning and finally inferring insightful information from the data and presenting that to audience in meaningful way. Major thrust is given on data processing and model preparation for some insightful information. Upon completion, students will excel in data handling, raising meaningful question for insights and come with model/ statistical test for acquiring the insight. Finally, a number of data representation methods are used to present the result in meaningful way.
Course Outline	**Unit I** Introduction to the data science and Python. **Unit II** Exploratory Data Analysis and the Data Science Process - Basic tools (Pandas, ScikitLearn, NumPy, Matplotlib, etc.). **Unit III** Python Programming for Statistics: Probability, Random Variable, Probability Distribution, central limit theorem **Unit IV** Inferential Statistics: population and sample, Point estimation, Interval estimation, hypothesis testing **Unit V** Supervised Learning- Linear Regression, k-Nearest Neighbors (kNN), Naïve Bayes, Decision Trees **Unit VI** Unsupervised Learning- k-means, DBSCAN, GMM, Principal Component Analysis
Learning Outcome	• A clear understanding of the core concepts and methodologies in data science. • Knowledge regarding programming languages (e.g., Python) and data manipulation libraries (e.g., pandas, NumPy) to clean, process, and analyze data. • Knowledge regarding exploratory data analysis (EDA) and capability to create predictive models using appropriate data science tools and techniques. • Drawing data-driven insights and recommendations from data. • Create visualizations and reports that convey findings in a compelling and understandable manner.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Probability and Statistics for Engineers and Scientist by Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying E. Ye, Pearson, 9th Edition
- An Introduction to Statistical Learning with Applications in R by Gareth James Daniela Witten, Travor Hastie, Robert Tibshirani, Springer
- Machine Learning by Tom Mitchel, McGraw Hill Education
- Cathy O'Neil and Rachel Schutt. Doing Data Science, Straight Talk From The Frontline O'Reilly. 2014
- Anil K. Jain, Richard C. Dubes, Algorithms for clustering data, Prentice Hall Advanced Reference Series: Computer Science, (2008)
- Rajeev Motwani and Prabhakar Raghavan, Python for Rusers a data science approach, Wiley, Year: 2018
- John D. Kelleher, Brendan Tierney, Data Science, The MIT Press, 2018

Course Number	**CS3106 (IDE-2)**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Computer Graphics**
Learning Mode	Offline
Learning Objective	The objective of the course is to provide a conceptual and theoretical understanding of the organization and functioning of a computer graphics rendering pipeline.
Course Description	Computer Graphics comprises of a pipeline of technologies that play an important role in developing computer vision and image processing technologies with wide applications in the field of Artificial Intelligence (AI).
Course Outline	Graphics imaging pipeline, Rasterization, Display devices, CRT displays, Random scan display, Raster scan display, Raster Scan Basics. 2D transformations, 3D transformations, Vanishing points, Viewing Transformation. Coding sessions in class using C++, Python. Digital Differential Algorithms, Bresenham's algorithms, polygon filling, Windowing and Clipping, problems of aliasing. Coding sessions in class using C++, Python. Graph based models, B-REP model, Constructive Solid Geometry (CSG), Octree based representation, Quadtree based representation. Parametric representation of curves, parametric cubic curves, Bezier curves, continuity of curves, modeling of surfaces. Hidden Surface Removal, Back face removal, Z-Buffer Algorithm, Scan-line algorithm for VSD, algorithm, BSP trees. Coding sessions in class using C++, Python.
Learning Outcome	• This course will teach the fundamentals of imaging graphics through which you will be able to develop various imaging applications. • This course also accompanies coding, using Python or C++ or Java and OpenGL, of every algorithm/technology that will be taught giving a first hand experience of imaging app development and how it works.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- P. Shirley, M. Ashikhmin and S. Marschner, Fundamentals of Computer Graphics, 3rd Edition, CRC Press, 2009.
- E. Angel and D. Shreiner, Interactive Computer Graphics, A top-down approach with OpenGL, 6th Edition, Addison Wesley, 2012.
- J. D. Foley, A. van Dam, S. Feiner, and J. F. Hughes, Computer Graphics: Principles and Practice, 2nd Ed, Addison-Wesley, 1996.
- D. F. Rogers and J. A. Adams, Mathematical Elements for Computer Graphics, 2nd Edition, McGraw-Hill International Edition, 1990.

Course Number	**CS4113 (IDE-3)**
Course Credit (L-T-P-C)	**3-0-0-3**
Course Title	**Data Analysis and Visualization**
Learning Mode	Offline
Learning Objectives	• To understand the fundamental concepts and principles of data analysis. • To acquire skills in data collection, cleaning, and preparation for analysis. • To learn statistical techniques and methods for analyzing data. • To gain proficiency in using software tools for data analysis, such as Python, R, and Excel. • To develop the ability to create meaningful and effective data visualizations. • To interpret and communicate data findings clearly and accurately. • To apply data analysis and visualization techniques to real-world problems.
Course Description	This course provides a comprehensive introduction to data analysis and visualization techniques. Students will learn how to gather, clean, and analyze data using various tools and methodologies. The course covers statistical analysis, data manipulation, and visualization best practices. Through hands-on projects and real-world examples, students will develop the skills necessary to transform data into actionable insights and effectively communicate their findings using visualizations.
Course Outline	Introduction to Data Analysis and Visualization: Overview of Data Analysis and Visualization, Importance of Data in Decision Making, Data Preprocessing Tasks, Some Mathematical Preliminaries Introduction to various tools: Python, R, Tableau, etc. Exploratory Data Analysis programming: Descriptive Statistics, Data Cleaning and Handling Missing Values, Data Visualization with ggplot2, Correlation and Covariance, Data Distribution and Outliers, Introduction to Statistical Modeling programming: Linear Regression: Concepts and Implementation, Multiple Linear Regression Analysis, Supervised Data Analysis programming: Introduction of Supervised Analysis Techniques, Various Classifier Models- Logistic Regression, Naïve Bayes Classifier, LDA, KNN, SVM, Decision Trees. etc. Evaluation Parameters, Practice and Analysis using R Unsupervised Data Analysis programming: Introduction of Unsupervised Analysis, Various Clustering Strategies- K-Means, DBSCAN, Hierarchical. Evaluation Strategies, Practice and Analysis using R Real-world applications and case studies, industry-specific use cases, mini project
Learning Outcome	By the end of this course, students will be able to: • Apply various data analysis and visualization techniques using various tools. • Perform data preprocessing, including cleaning, handling missing values, and transforming data. • Conduct exploratory data analysis and create informative visualizations. • Implement and interpret statistical models and supervised learning techniques. • Execute unsupervised learning techniques and evaluate their effectiveness. • Apply learned techniques to real-world scenarios through case studies and projects, demonstrating their practical utility.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Data Analytics & Visualization, Jack A. Hyman et al, April 2024
- An Introduction to Statistical Learning with Application in R by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, 2nd Edition, Springer
- Applied Predictive Modeling by Max Kuhn and Kjell Johnson, 2nd Edition, Springer, ISBN: 978-1461468486
- Visual Analytics with Tableau by Alexander Loth, ISBN: 978-1119560203
- Introduction to Data Mining, by Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar, 2nd Edition, Pearson.
- Machine Learning with R by Brett Lantz, Packt Publishing
- Practical Data Science with R by Nina Zumel, John Mount, Manning Publication, ISBN- 978-1617291562

- The Art of R Programming by Norman Matloff, No Starch Press, ISBN: 9781593273842
- R in a Nuttshell- A Desktop Quick Reference by Joseph Adler, Shroff/O'Reilly, ISBN: 978-9350239209
- Hands-On Machine Learning with R by Brad Boehmke and Brandon Greenwell, CRC Press, 978-1138495685
- Mastering Tableau 2023 by Marleen Meier, Packt Publishing; 4th ed. Edition, ISB: 978-1803233765