Syllabus Document

B. Tech. in CSE (Artificial Intelligence & Machine Learning)

For 2022 Batch onwards

1st Semester

Course Code	MAL105	Course Title	Calculus for Data Science			
Category	Core	Credit Assigned	L	T	P	C
			3	1	0	4
Prerequisite (If any)	-	Type of Course	Basic Science			

Course Outcomes:

1) To analyze the nature (convergence or divergence) of a sequence or series.
2) To apply mean value theorems in the study of motion of an object.
3) To use integration in the calculation of area, volume, mass, and centre of gravity.
4) To apply multivariable calculus to study the nature of multivariable functions.
5) To understand the concept of Differential equation and its application

Course Contents:

Module 1:

Sequences and series: Sequences of real numbers, Series, ratio and root test.

Module 2:

Calculus of functions of single variable: Review of limits, continuity, and differentiability. Mean value theorems: Rolle's theorem, Lagrange's theorem, Cauchy's theorem, Taylor's theorem with remainders, indeterminate forms, curvature, curve tracing. Fundamental theorem of Integral calculus, mean value theorems of integral calculus, evaluation of definite integrals, applications in area, length, volumes and surface of solids of revolutions, Improper integrals: Beta and Gamma functions, differentiation under integral sign.

Module 3:

Calculus of Functions of Several Variables: Limit, continuity and differentiability of functions of several variables, partial derivatives and their geometrical interpretation, Tangent plane and normal line. Total differentiation, chain rules, Taylor's formula, maxima and minima, Lagrange's method of undetermined multipliers. Double and triple integrals, Jacobian, change of order of integration, change of variables, application to area, volumes, Mass, Centre of gravity.

Module 4:

Differential equation and its modelling with curve fitting: Modelling with Differential Equations , Direction Fields and Euler's Method , Linear and Bernoulli's differential equations, Nonlinear differential equations, Polar curves, angle between the radius vector and the tangent, angle between two curves. Pedal equations. Curvature and Radius of curvature - Cartesian, Parametric, Polar and Pedal forms. Problems Canter and circle of curvature, evolutes and involutes.

Text Books:

1. Kreyszig, E., Advanced Engineering Mathematics, John Wiley & Sons

Reference Books:

1. Piskunov, N., Differential and Integral calculus, Mir publishers Moscow (Vol. 1, Vol. 2)

<table>
<tr><td>Course Code</td><td>CSL110</td><td>Course Title</td><td colspan="4">Conversational AI</td></tr>
<tr><td rowspan="2">Category</td><td rowspan="2">Core</td><td rowspan="2">Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>Prerequisite (If any)</td><td>-</td><td>Type of Course</td><td colspan="4">Computer Science and Engineering</td></tr>
<tr><td colspan="7">Course Outcomes:
Students will understand the concepts of chatbot designing
Students will be able to build their own chatbots
Students will be able to deploy chatbot for its practical use</td></tr>
<tr><td colspan="7">Course Contents:

Module 1:
Conversational Design Process: Introduction to virtual assistant/chatbot, use cases, what is conversational design, conversational design process, designing conversational flows, writing the script, designing your conversations, Introduction to Dialogflow, Setting up Dialogflow

Module 2:
Building blocks of Interaction models: Agents, types of Intents, creating Intents, training phrases, Entities, configuring rich responses, small talk and salutations, Configuring and testing Intents on Google Assistant, Working on Connected Flows.

Module 3:
Linear and Non-linear dialogue: Actions & Parameters, understanding slot filling, context, extended Lead Generation, linear dialogue, nonlinear Dialogue, webhook, Fulfilment.

Module 4 :
Fulfilment: Fulfilment using webhook, basic setup of webhook code, Extracting parameter values and structuring responses, fulfilment using cloud function

Module 5:
Deployment: Introduction to Heroku, Deploying to Heroku, Deploying on Alexa, Re-training , Validation & Testing.</td></tr>
<tr><td colspan="7">Text Books:
Hands-on chatbot with Google Dialogflow, Loonycorn, O’Reilly, Packt publishing
Hands-on chatbots and conversational UI development, Srini Janarthanam, Packt publishing</td></tr>
</table>

<table>
<tr><td>Course Code:</td><td>CSL 101</td><td>Course Title:</td><td colspan="4">Computer Programming</td></tr>
<tr><td>Category:</td><td>Core</td><td>Credit Assigned</td><td colspan="4"><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>0</td><td>2</td><td>4</td></tr></table></td></tr>
<tr><td>Pre-Requisite (if Any)</td><td>Nil</td><td>Type of Course</td><td colspan="4">Computer Science Engineering</td></tr>
</table>

Course Outcomes:

CO1 - Outline basics of programming and develop logical thinking of students.
CO2 - To illustrate how to model real world problems into the software and develop practical programming skills of students.
CO3 - To use mathematical and statistical applications into programming.

Course Contents:

Module 1:
Introduction – Computer generation and evolution, flowcharts, algorithm, What is C?, constants, variables, scope of variable, data types, operators, arithmetic expression, Hierarchy of operators, control flows, conditional operator, loops, switch concept.
Program Structure – Basic programs to illustrate structure of C program and its flow in execution.

Module 2:
Function – Introduction to function and parameter passing, returning value, recursive functions, macros.

Module 3:
Arrays – One-dimension and multi-dimension arrays, array initialization, how arrays are stored in memory, array as parameter in functions, programs based on arrays.

Module 4:
Pointers – Initialization, accessing a variable through pointers, pointers as function arguments, pointer to array, arrays of pointers, pointers to pointers.

Module 5:
Structure and Union – Defining a structure, accessing structure members, Array of structure, unions.

Module 6:
File Handling- reading from and writing to a file.

Text books:
1) The C Programming Language by Brian W. Kernighan and Dennis M. Ritchie, PHI.
2) Programming in C by E. Balguruswamy, Tata Mcgraw Hill Publishing.

List of Lab Assignments / Experiments

1. Programs using function.
2. Programs using arrays.
3. Programs on structures.
4. File Handling

Course Code:	**CSL111**	**Course Title:**	**AI, Ethics, and Society**
Category:	**Core**	**Credit Assigned**	**L** | **T** | **P** | **C**

			2 | 0 | 0 | 2
Pre-Requisite (if Any)	**Nil**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:

After successfully completion of the course, Students shall be able to

1. To understand the power and impact that analytics and AI/ML have on individuals and society.
2. To understand the underlying components of big data
3. To understand and apply basic AI/ML techniques to data scenarios, with a focus on identifying fairness and bias issues.
4. To utilize tools and methods to quantify bias and examine ways to use algorithmic fairness to mitigate this bias.

Course Contents:

Module 1 - Data, Individuals, and Society

Power and impact of analytics and AI/ML on individuals and society, fairness and bias, ethics, legality, data collection and public use.

Module 2 – The foundation of Big Data

Various components of big data, statistical techniques to data scenarios, issues in learning from big data, ranging from data biases, overfitting, causation vs correlation.

Module 3 – Fairness in AI/ML

Basic AI/ML techniques for data handling, identification of fairness and bias, issues in the design of decision-making systems, Fairness and bias in the social and legal context of facial recognition, natural language processing, and predictive algorithms,

Module 4 – Bias Mitigation and Future Opportunities

Quantify of bias, Various methods of algorithmic fairness to mitigate this bias, uses of analytics and AI/ML to transform a current biased data-set into a more objective solution.

Text Books:

1. Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy by Cathy O'Neil (2016)

Reference Books:

1. AI ethics by mark Coeckelbergh, MIT Press, 2020.
2. S.Matthew Liao, Ethics of Artificial Intelligence, Oxford University Press, 2020

Course Code	**ECL103**	**Course Title**	**Applied Electronics**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Prerequisite (If any)	-	**Type of Course**	**Electronics Engineering**			
Course Contents:						

Module 1: ELECTRONIC DEVICES

Theory of P-N Junction Diode, Junction Transistors Theory of Operation, Static Characteristics, Break Down Voltages, Current Voltage Power Limitations, Field Effect Transistor & MOSFET, Principle of Operation & Characteristics.

Module 2: APPLICATIONS of ELECTRONIC DEVICES
Rectifiers, Zener Diode as Regulators, Biasing of BJT Different Biasing Arrangements, Stability Factor, Small Signal Analysis & High Frequency Analysis of BJT, Power Amplifiers, Push Pull Configuration, Complimentary Symmetry, Feedback Amplifiers, RC, LC & Crystal Oscillators.

Module 3: COMBINATIONAL and SEQUENTIAL LOGIC
Logic minimization using K-map method, multiplexers, demultiplexers, decoders, encoders, Arithmetic circuits, Adders, Combinational multiplier and code converters. Basic latches, master-slave latch, Flip flops, Registers, Counters.

Module 4: MEMORIES
Introduction to PLA, PAL and ROM, Programmable Logic Devices and FPGAs.

Module 5: INTRODUCTION TO MICROPROCESSORS
Architecture, bus structure, timing diagrams, T-states, machine cycle, instruction cycle. Memory and IO devices interfacing.

Reference Books:

1) Electronic devices and circuit theory / Robert L. Boylestad, Louis Nashelsky
2) Milman and Halkias, "Integrated Electronics", Second Edition, 2011, McGraw Hill.
3) Digital Design by M. Morris Mano and Michael D. Ciletti
4) Microprocessor Architecture, Programming, and Applications with the 8085 by Ramesh Gaonkar

Course Code:	**HUL 102**	**Course Title:**	**Environmental Studies**
Category:	**Core**	**Credit Assigned**	**L** | **T** | **P** | **C** **2** | **0** | **0** | **2**
Pre-Requisite (if Any)	**Nil**	**Type of Course**	**Basic Science**

Course Outcomes:

1. Identify natural resources, ecosystem, and biodiversity, their structure and functions.
2. Describe the importance of environmental components, and their role in human life.
3. Illustrate the possible causes of various forms of environmental pollution, their consequences, and methods of prevention.
4. Define the concept of sustainable development and mechanism to attain it.
5. Recognize the integration of social issues and environmental problems.

Course Contents:

Module 1: Natural resources: Forest resources, Water resources, Mineral resources, Food resources, Energy resources, Land resources.

Module 2: Ecosystem: Concept of an ecosystem, Structure and functions of an ecosystem, Producers, consumers and decomposers, Ecological succession, Food chain, food webs and pyramids.

Module 3: Biodiversity and its conservation: Introduction, definitions: genetics, species and diversity, Value of biodiversity, Biodiversity at global, national and local level, India as a mega-diversity nation, Hot-spot of biodiversity, Threat to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts, Conservation of biodiversity: in-situ and ex-situ conservation.

Module 4: Environmental pollution: Definition, Causes, effects and control measures of: Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution, Nuclear hazards. Solid waste management: Causes, effects and control measures of urban and industrial wastes.

Module 5: Social issues and environment: Sustainable development, Water conservation, Rain water harvesting, Watershed management, Climate change, Global warming, Ozone layer depletion, Nuclear accident, Environmental rules and regulations, Human population and environment. Role of information technology in environment and human health.

Case studies related to ecosystem, environmental pollution, sustainable development will be discussed in a class.

Text:

1. Rajagopalan, Raghavachari. *Environmental studies: from crisis to cure*. No. Ed. 3. Oxford University Press, 2015.

Reference:

1. Joseph, Benny. *Environmental studies*. Tata McGraw-Hill Education, 2005.
2. Chopra, Kanchan and G. K. Kadaekodi, Operationalizing Sustainable Development: Economic- Ecological Modeling for Developing Countries: Sage, Chapter 1, 1999.
3. Kolstad, Charles, D., *Environmental Economics*, Press, 2000.
4. Reed, David (Ed.), *Structural Adjustments, the Environment and Sustainable Development*, Earthscan, Chapters 1, 12, 13 and 14, 1996.
5. Bharucha, Erach. *Textbook of Environmental Studies for Undergraduate Courses*. Universities Press, 2005.

Course Code:	**SAP 101**	**Course Title:**	**Health, Sports & Safety**
Category:	**Core**	**Credit Assigned**	**L** **T** **P** **C** **0** **0** **2** **0**
Pre-Requisite (if Any)	**Nil**	**Type of Course**	**Basic Science**

Course Outcomes:

1. To provide physical fitness and good health.

2. Create awareness among the students about their health status by conducting various tests and measurements and suggest them suitable remedial physical fitness program so that they can improve physical and physiological health status.
3. To improve productivity, foster social harmony, inculcate sense of discipline and dedication in general life, develop the spirit of team work, through various sports activities.

Course Contents:

Development of components of fitness through conditioning exercises: Strength: (Strength Endurance, Maximum Strength, explosive strength), Endurance: (aerobic endurance, anaerobic endurance, speed endurance and strength endurance), Speed, Co-coordinative ability, Flexibility

Physical Efficiency Test Level 1(Testing and Evaluation of Physical Fitness): Cooper Test 12 minute run or walk test, Sit and reach test, 100 meter run, one minute sit up test, Push up/Bent knee push up test

Teaching and development of sports skills: Cognitive, Perceptual, Motor, Perceptual motor. First Aid training

Intramural phase 1: Identification of sports talent through exposing students to inter- section tournament. Football, Volleyball, throw ball, table tennis & Chess.

Yoga, Meditation and Personal Safety.

2nd Semester

Course Code	**MAL106**	**Course Title**	**Probability and Statistics**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**1**	**0**	4
Prerequisite (If any)	-	**Type of Course**	**Basic Science**			

Course Outcomes:

At the end of the course, students will be able to :

1. Solve problems of basic probability, two types of random variables and their probability functions.
2. Observe and analyze the behaviour of various discrete and continuous probability distributions.

3. Formulate an appropriate null and alternative hypothesis. Perform test of Hypothesis for decision making and validation.
4. Perform Regression and correlation analysis.
5. Apply sampling distributions to testing of hypotheses.

Course Contents:

Module 1:

Sample space and events – Probability – The axioms of probability, Calculating probability: sets, counting, tree diagram, Conditional probability – Baye's theorem, Bayesian reasoning, Bayesian inference, Priors, Conjugate priors, Random variables – Discrete and continuous, probability density function; probability distribution function for discrete and continuous random variable. Combinatorics: Permutation and combination.

Module 2:
Definition of mathematical expectation, functions of random variables, mean, moments, variance and standard deviations, moment generating function.

Module 3:
Discrete Random Variables (RVs): Bernoulli, Binomial, Geometric, Poisson. Sampling distributions –Sampling distribution of means (known and Unknown). Continuous Random Variable, Uniform, Exponential, Random variable for Normal Distribution, Joint probability distribution, Linearity (and product) of expectation, Conditional expectation, Sum of a random number of RVs, Probability inequalities, Markov's Inequality, Chebyshev's inequality, Weak Law of Large Numbers, Central Limit Theorem.

Module 4:
Tests of hypothesis, point estimations – interval estimations, Bayesian estimation. Large samples, Null hypothesis – Alternate hypothesis type I, & type II errors – critical region confidential interval for mean testing of single variance. Difference between the mean. Confidential interval for the proportions. Tests of hypothesis for the proportions single and difference between the proportions.

Module 5:
Regression and Correlation Analysis: Introduction, Basics of Regression, Simple Linear Regression, Multiple Linear Regression, estimation and analysis of simple regression models, correlation coefficients, analysis of correlation coefficients, Hypothesis tests associated with regression and correlation coefficients, curvilinear regression models, Multiple regression models, multiple and partial correlation coefficients. EWMA Time Series modeling, AR Time Series modelling.

Text Books:

1. D. K. Murugesan & P. Guru Swamy, "Probability & Statistics", Anuradha Publications.

2. G. S. S. Bhisma Rao, "Probability & Statistics for Engineers", Scitech Publications.

3. Spiegel, Murray, "Probability and Statistics", Schaum's series,.

Reference Books:

1. K.V. Iyengar & B. Krishna Gandhi , "Probability & Statistics", S.Chand.

2. William Mendenhall & Others, " Probability & Statistics", Cengage Publications.

3. P. Billingsley, *"*Probability and Measure*"*, John Wiley & Sons (SEA) Pvt. Ltd.

4. W. Feller, "An introduction to probability theory and its applications", John Wiley and Sons.5.

5. Levin, Rubin, Rastogi, "Statistics For Management", 8th edition, Pearson.

Course Code	**MAL107**	**Course Title**	**Introduction to Linear Algebra**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**1**	**0**	4
Prerequisite (If any)	-	**Type of Course**	**Basic Science**			

Course Outcomes:

1. Describe properties of linear systems using vectors and Solve systems of linear equations and interpret their results
2. Demonstrate an understanding of linear transformations and Perform and interpret matrix operations
3. Compute and interpret determinants of matrices and Demonstrate an understanding of vector spaces and sub-spaces
4. Demonstrate an understanding of eigenvalues and eigenvectors

Course Contents:

Module-1

Introduction to Vectors, Vectors and Linear Combinations, Lengths and Dot Products, Matrices, Solving Linear Equations,Vectors and Linear Equations,The Idea of Elimination, Elimination Using Matrices, Rules for Matrix, Operations, InverseMatrices, Elimination = Factorization: $A = LU$, Transposes and Permutations Vector Spaces and Subspaces, Spaces of Vectors,The Nullspace of A: Solving $Ax = 0$ and $Rx =$ The Complete Solution to $Ax = b$.

Module-2

Independence, Basis and Dimension, Dimensions of the Four Subspaces, Orthogonality, Orthogonality of the Four Subspaces, Projections, Least Squares Approximations, Orthonormal Bases and Gram-Schmidt Determinants, The Properties of Determinants, Permutations and Cofactors, Cramer's Rule, Inverses, and Volumes, Eigenvalues and Eigenvectors, Introduction to Eigenvalues

Module-3

Diagonalizing a Matrix, Systems of Differential Equations, Symmetric Matrices, Positive Definite Matrices, The Singular Value Decomposition (SVD), Bases and Matrices in the SVD, Principal Component Analysis (PCA by the SVD) The Geometry of the SVD Linear Transformations,The Idea of a Linear Transformation

Module-4

The Matrix of a Linear Transformation, The Search for a Good Basis, Complex Vectors and Matrices, Complex Numbers, Hermitian and Unitary Matrices, The Fast Fourier Transform, Applications, Graphs and Networks, Matrices in Engineering, Markov Matrices, Population, and Economics, Linear Programming, Fourier Series: Linear Algebra for Functions.

Module-5

Numerical Linear Algebra, Gaussian Elimination in Practice, Norms and Condition Numbers, Iterative Methods and Preconditioners, Mean, Variance, and Probability, Covariance Matrices and Joint Probabilities,Multivariate Gaussian and Weighted Least Squares, Matrix Factorization

Text Books:

1. Kenneth Hoffman and Ray Kunze: Linear Algebra, Prentice Hall of India limited, New Delhi, 1971.
2. Gilbert Strang : Linear Algebra And Its Applications (Paperback) , Nelson Engineering (2007).
3. Introduction to Linear Algebra: Gilbert Strang

Reference Books:

1. Gilbert Strang: Introduction to Linear Algebra, Wellesley- Cambridge Press, Fourth Edition, 2011.
2. Jin Ho Kwak and Sungpyo Hong, Linear Algebra, Springer, Second edition, 2004.
3. V. Krishnamoorthy et. al., An introduction to linear algebra, Affiliated East West Press, New Delhi.
4. Elementary of Linear Algebra Howard Anton

Course Code:	**CSL 102**	**Course Title:**	**Data Structures**
Category:	**Core**	**Credit Assigned**	**L** | **T** | **P** | **C** **3** | **0** | **2** | **4**
Prerequisite (if Any)	**CSL 101 (Computer Programing)**	**Type of Course**	**Computer Science**

Course Outcomes:

Students will be able to:

1. Design and differentiate the recursive and iterative versions of the program.
2. Implement and analyse algorithms using dynamic memory allocation.
3. Apply, implement and evaluate various concepts of linear data structures for solving real life problems.
4. Apply, implement and evaluate non-linear data structures in solving various problems.

Course Contents:

Module 1: Types and operations, Iterative constructs and loop invariants, Quantifiers and loops, Structured programming and modular design, Illustrative examples, Scope rules, parameter passing mechanisms, recursion, program stack and function invocations including recursion, Overview of arrays and array based algorithms - searching and sorting, Mergesort, Quicksort, Binary search, Introduction to Program complexity (Big Oh notation)

Module 2: Implementation of Structures (Records) and array of structures (records). Database implementation using array of records. Dynamic memory allocation and deallocation. Dynamically allocated single and multi-dimensional arrays.

Module 3: Lists as dynamic structures, operations on lists, implementation of linked list using arrays pointers and its operations. Introduction to Overview of linked list implementation using self-referential-structures/pointers. Stack, Queues and its operations. Concept of an Abstract Data Type (ADT),Implementation of stacks and queues using both array-based and pointer-based structures. Uses of stacks in simulating recursive procedures/ functions. Applications of stacks and queues.

Module 4: Lists - Singly-linked lists, doubly linked lists and circular linked lists. List traversal, insertion, deletion at different positions in the linked lists, concatenation, list-reversal etc. Mergesort for linked lists.

Module 5: Trees, binary trees, binary trees- basic algorithms and various traversals. Binary Search Trees (BSTs) and insertion, deletion in BSTs.

Module 6: Generalization of trees to graphs – their representation & traversals. Directed Acyclic Graphs and topological sort, Dijkstra‟s shortest path algorithm,

Text:

1. Data Structures & Program Design in C: Robert Kruse, G. L. Tondo and B. Leung PHI-EEE.
2. Fundamentals of Data Structures in C : E. Horowitz, S. Sahni, and S. Anderson-Freed, University Press

Reference:

1. Aho, Hopcroft and Ullmann, ―Data Structures and Algorithms,‖ Addison Wesley, 1983.

Course Code:	**CSL 103**	**Course Title:**	**Application Programming**
Category:	**Core**	**Credit Assigned**	**L** | **T** | **P** | **C** **3** | **0** | **2** | **4**
Pre-Requisite (if Any)	**CSL 101 (Computer Programming)**	**Type of Course**	**Computer Science**

Course Outcomes:

1. Aware about different tools for Web Programming.
2. Background of working on web.
3. Construct efficient web pages with CSS and Javascript.
4. Demonstrate competency in the use of common HTML code.

Course Contents:

Internet fundamentals, LAN, WAN, Introduction to common Internet terms, www. Basics of networking, DNS, URL, firewall, proxy, Web protocols – http and https.

Designing web pages: HTML, forms, DHTML, XML, CSS. Extensible Hypertext Mark up Language (XHTML): XHTML syntax, headings, linking, images, special characters and horizontal rules, lists, tables, forms, internal linking, meta elements.

Introduction to Web Server – Setting up and configuration of Apache Tomcat server, Accessing pages from another machine.

Server Side Programming: Introduction to web programming with PHP. Client side programming with Javascript.

Introduction to Python - Statements and Control Flow, Expressions, Methods, Typing, Libraries and Developmental Environment, Web Programming using Python.

Text:

1. Deitel H.M. and P. J. Deitel, Internet & World Wide Web - How to Program, Prentice-Hall.
2. Goodman D, Morrison M., JavaScript Bible; Wiley India
3. Lutz, Mark, Learning Python (4th ed.). O'Reilly Media

Reference:

1. Garfinkle S., Spafford G; Web Security, Privacy and Commerce; O'Reilly, 2002.
2. Atkinson L., Core PHP Programming, Prentice Hall.
3. N.P.Gopalan, Akilandeswari, Web Technology, Prentice-Hall.

Course Code:	**HUL 101**	**Course Title:**	**Communication Skills**
Category:	**Core**	**Credit Assigned**	**L** | **T** | **P** | **C** **2** | **0** | **2** | **3**
Pre-Requisite (if Any)	**Nil**	**Type of Course**	**Humanities**

Course Outcomes:

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

1. Utilize functional English grammar for accurate and enhanced language skills.
2. Construct and use effective interpersonal and workplace communication
3. Acquire better reading comprehension, pronunciation and reading skills
4. Introspect and illustrate the personality traits and soft skills
5. Develop the skills for better pre and post placement communication through effective presentations, personal interviews and group discussions

Course Contents:

Module 1
Communication:-
Definition of Communication, Process of Communication, Stages of Communication, Content of the message, Types of communication, Transmission, Medium/Modes of Communication, Verbal and Non-verbal Communication (Kinesics, Proxemics, Chronemics, Haptics, Paralinguistic Feature), Levels of Communication, Flow of Communication, Communication Networks, Grapevine, Barriers to Communication, Choice of Medium,

Module 2
Listening Skills:-
Art of Listening, Listening vs Hearing, (Poor Listening vs Effective Listening), Advantages of Good Listening, Barriers to Effective Listening, Techniques of Effective Listening
Reading Skills:-
Reading Comprehensions, Process of Reading, Techniques of Reading, Techniques for Good Comprehension, Reading Skills(Skimming, Scanning, Intensive Reading, SQ3R), Orientation in Literary and Scholarly Article

Module 3
Speaking Skills:-
Types of Speech, Public Speaking, Components of Effective speech, Stage Presence & Personality Development, Clarity and Fluency, Body Language, , Barriers to Effective Speaking
Presentation Skills:-
Characteristics of a Successful Presentation, Power Point Presentation, Using Audio Visual Aids

Module 4
Group Discussion:-
Do's and Don'ts of GD, Essential Skills for GD, Evaluation Pattern
Personal Interview:-
Objectives of Interview, Types of Interview, Job Interviews, Employer's Expectations, Do's & Don'ts of Social Media Profile, Success Factors, Failure Factors

Module 5
Grammar:-
Transformation of Sentences, Punctuation, Spellings and Mechanics of Writing

Text Books:

1. Orient Longman, A Textbook of English for Engineers and Technologists
2. M. Ashraf Rizvi, Effective Technical Communication. Tata Mc Grwa-Hill Publishing Company Limited, 2009

Reference Books:

1. Quirk R. and Greenbaum S., A University Grammar of English.
2. Krishnaswamy N., English Grammar (Longman Publication) (Macmillan India Ltd)
3. Sanjay Kumar and PushpaLata, Communication Skills. Oxford Publication
4. Meenakshi Raman and Sangita Sharma. Technical Communication. Second Edition. Oxford Publication,2011

List of Lab Assignments/Experiments

1. Speaking Skills (Verbal/Non verbal Skit, Role Play, Extempore, Story Telling, Word Wheel, Debate)
2. Presentation Skills (Film/Book Review, PPT Presentation)
3. Group Discussion (Practice GD, Mock GD)
4. Personal Interview/ SWOT Analysis (SWOT Analysis, Mock PI)
5. Comprehending a Technical Report/News Paper Article.
6. Presenting a Book Chapter using PowerPoint slides

Course Code:	**CSP 201**	**Course Title:**	**IT Workshop-I**
Category:	**Core**	**Credit Assigned**	**L** | **T** | **P** | **C** **0** | **0** | **4** | **2**

Pre-Requisite (If Any)	**None**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:

1. Effectively use the Unix programming environment - shell, file system, scripts, filters, program development tools.
2. Develop good programming style using Python with usage of packages: math, Cmath and functions.
3. Be familiar with writing of real time application programs using the concepts like class, object, inheritance, constructor, tkinter.
4. Use of effective procedures and tools for data analytics using graphical outcomes: Pandas, numpy, openpyxl and matplotlib.
5. Learn to automate tasks for making predictions using machine learning: scikit learn, countplot.

Course Contents:

Module 1:
Introduction to different tools for identification and possibility of errors in C program – gdb, concepts of "core dump", backtracing using "bt", using "info" to dump all registers, creating watch-list / watch variables. DDD (Data Display Debugger) – introduction and usage, debugging with ddd (step, step into, step over). Using DevCpp and/or VisualStudio b. Setting compiler options and linker options. Unix tools - Awk, sed, Emacs. Make files and automated builds.

Module 2:
Text editors. Users, files, permissions, and processes on Linux. Introduction to shell: Set and Unset a variable, Displaying – using echo, Using Expr & Test, Getting input – using read, Header files of shell script – using Shabang, Sample Shell script program. Assigning a command to a variable, Storing output to a variable, Assigning global value – using Export. Command Line Arguments, Conditional & Looping Statement, Functions.

Module 3:
Advanced Commands: SED, Replacing values in a file, STTY, TOP, Sending an email using MAIL, HERE. Scheduler: Scheduling a job – using 'Crontab', 'at' and 'nohup. Shell Programming: Essential systems administration with shell scripting and elementary Python, Version control. Advanced Shell Scripting: Monitoring a file, Handling Shell Script Interrupts, Extracting data from HTML/XML file, Trapping Signals Database Connectivity, Connecting MYSQL to Shell, Running SQL queries from Shell Script.

Module 4:
Bash and Bash Scripting: Common shell programs, Advantages of BASH, Executing commands, Building blocks, developing good scripting, variables, conditionals, loops, finding logged in users. Writing and Debugging Scripts.

Module 5:
Bash Environment: Shell Initialization files, Quoting characters, Shell expansion, Aliases and More options in Bash. Regular Expressions: Meta characters, Extended regular expressions Using GREP, Pattern matching. Python Integration, Testing and Debugging with Software Development Practice.

Text Books:
1. Christopher Negus "Linux Bible", Wiley
2. Steve Parker "Shell Scripting: Expert Recipes for Linux, Bash & more" Wrox

3. Richard Petersen “Linux: The Complete Reference”, TMH
4. Robert Collins “Shell Programming and Bash Scripting: Ultimate Beginners Guide Book”, CreateSpace

2nd Year Syllabus

3rd Semester

<table>
<tr><td>Course Code:</td><td>CSL202</td><td>Course Title:</td><td>Introduction to Object Oriented Programming</td></tr>
<tr><td>Category:</td><td>Core</td><td>Credit Assigned</td><td><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>0</td><td>2</td><td>4</td></tr></table></td></tr>
<tr><td>Pre-Requisite (if Any)</td><td>None</td><td>Type of Course</td><td>Computer Science & Engg.</td></tr>
<tr><td colspan="4">Course Outcomes:
Students will be able to:
1. Design solution for small and large systems or organizations using object oriented concepts.
2. Evaluate procedural and object oriented paradigms.
3. Apply the concepts of object oriented programming for developing software.</td></tr>
<tr><td colspan="4">Course Contents:
Module 1:
Object Oriented Programming, Features of object oriented programming languages like data encapsulation, inheritance, polymorphism and late binding.

Module 2:
Concept of a class, Access control of members of a class, instantiating a class, static and non-static members, overloading a method. Deriving a class from another class, access control of members under derivation, different ways of class derivation, overriding of a method, run time polymorphism.

Module 3:
Concept of an abstract class. Concept of an interface. Implementation of an interface. Exception and exception handling mechanisms. Study of exception handling mechanisms in object-oriented languages

Module 4:
Introduction to streams, use of stream classes. Serialization and de-serialization of objects. Templates, Implementation of data structures like linked lists, stacks, queues, trees, graphs, and hash table etc. using object oriented programming languages.

Module 5:
Introduction to concept of refactoring, modelling techniques like UML, Design patterns.</td></tr>
</table>

Text:

1. Bjane Strostrup, "The C++ programming language", Addison-Wesley
2. Herbert Shildt, "C++: The Complete Reference", 4th Edition
3. Arnold Ken, Gosling J, "The Java Programming Language", Addison Wesley
4. Matt Weisfeld, "The Object-Oriented Thought Process", Pearson
5. Cox Brad, "Object –Oriented Programming: An Evolutionary Approach", Addison –Wesley

Course Code	**CSL 210**	**Course Title**	**Data Structures with Applications**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			2	**0**	**2**	**3**
Prerequisite (If any)	**Data structures(CSL-102)**	**Type of Course**	**Computer Science and Engineering**			

Course Outcomes:

1. Ability to design and analyze the applications based on dynamic memory allocation such as linked lists.
2. Ability to apply and relate the concepts of height balanced trees for comparative analysis, and their applications to real world.
3. Ability to incorporate the knowledge of tries and skip lists for different applications.
4. Ability to apply the knowledge of graph data structures for various applications and algorithm design paradigms.

Course Contents:

Module 1:

Applications of lists in polynomial representation, multi-precision arithmetic, Hash-tables, Radix Sort etc. Multi linked structures and an example application like sparse matrices. Implementation of priority queues.

Module 2:

Overview of Binary Search Tree (BST), Height-balanced (AVL) trees, insertion/deletion and rotations. Heaps and heapsort.

Multi-way trees and external sorting - B-trees – insertion and deletion, Introduction to B+ trees with insertion and deletion algorithms. Red-black trees, Splay trees.

Module 3:

Tries, Multi-way tries, Suffix trees, Segment trees. Applications of the above mentioned trees. Introduction to Skip lists, Data structures for disjoint set representation

Module 4 :

Overview and definition of Graph as data structure, Traversals (BFT, DFT, Topological Sort), Data structures for Dijkstra's Shortest Path Algorithm, All-pairs shortest paths, Minimum spanning trees – Algorithms (Kruskal, Prim) and data structures. Huffman coding. Introduction to network flow problem.

Text Books:

1. Data Structures & Program Design in C: Robert Kruse, G. L. Tondo and B. Leung PHI-EEE.
2. Fundamentals of Data Structures in C : E. Horowitz, S. Sahni, and S. Anderson-Freed, University Press

Reference:

1. Aho, Hopcroft and Ullmann, ―Data Structures and Algorithms,‖ Addison Wesley, 1983.

Course Code:	**CSL 204**	**Course Title:**	**Discrete Maths and Graph Theory**
Category:	**Core**	**Credit Assigned**	**L** | **T** | **P** | **C** **3** | **1** | **0** | **4**
Pre-Requisite (if Any)	**None**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:

1. Student should be able to use different proof techniques.
2. Students would be able to argue about limits by using Pigeon Hole principle.
3. Solve problems based on set theory, Permutations and Combinations, as well as Discrete Probability.
4. Students will be able to solve mathematical problems on partial orders, and group theory.
5. Students would be able to model and analyze computational problems in graph theoretical framework.

Course Contents:

Module 1:
Set Theory: Definition of sets, countable and uncountable sets, Venn Diagrams, proofs of some general identities on sets Relation: Definition, types of relation, composition of relations, Pictorial representation of relation, Equivalence relation, Partial ordering relation, Job-Scheduling problem.

Module 2:
Algebraic Structures: Definition, Properties, types: Semi Groups, Monoid, Groups, Abelian group, properties of groups, Subgroup, cyclic groups, Cosets, factor group, Permutation groups, Normal subgroup, Homomorphism and isomorphism of Groups, example and standard results, Rings and Fields: definition and standard results.

Module 3:
Propositional Logic: Proposition, First order logic, Basic logical operation, truth tables, tautologies, Contradictions, Algebra of Proposition, logical implications, logical equivalence, predicates, Normal Forms, Universal and existential quantifiers. Introduction to finite state machine Finite state machines as models of physical system equivalence machines, Finite state machines as language recognizers.

Module 4:
Graph Theory: Introduction and basic terminology of graphs, Planer graphs, Multigraphs and weighted graphs, Isomorphic graphs, Paths, Cycles and connectivity, Shortest path in weighted graph, Introduction to Eulerian paths and circuits, Hamiltonian paths and circuits, Graph coloring, chromatic number, Isomorphism and Homomorphism of graphs. Graphs, hypergraphs, transitive closure, trees, spanning trees. Eulerian tours, Hamiltonian cycles, Planar Graphs, Connectivity, Colorability, Line Graphs

Module 5:
Posets, Hasse Diagram and Lattices: Introduction, ordered set, Hasse diagram of partially, ordered set, isomorphic ordered set, well ordered set, properties of Lattices, bounded and complemented lattices.
Combinatorics: Introduction, Permutation and combination, Binomial Theorem, Multimonial Coefficients Recurrence Relation and Generating Function, Recurrence Relation and Recursive algorithms , Linear recurrence relations with constant coefficients, Homogeneous solutions, Particular solutions, Total solutions .

Text Books:
1. C.L.Liu, “Elements of Discrete Mathematics” Tata Mc Graw-Hill Edition.
2. Trembley, J.P & Manohar; “Discrete Mathematical Structure with Application CS”, McGraw Hill.
3. Kenneth H. Rosen, “Discrete Mathematics and its applications”, McGraw Hill.
4. Lipschutz; Discrete mathematics (Schaum); TMH
5. Deo, Narsingh, “Graph Theory With application to Engineering and Computer Science.”, PHI.

Course Code	**CSL216**	**Course Title**	**Foundations of Computing**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Prerequisite	-	**Type of Course**	**Computer Science & Engineering**			

(If any)			

Course Outcomes:

After the successful completion of the course, the students will be able to:
1. Design an appropriate machine for the recognition of a given language.
2. Prove the grammar, language, and automata by using formal mathematical methods.
3. Analyse the lexical, syntactic, and semantic structures of a language.
4. Design a scanner, parser, and semantic analyzer for language translation.

Course Contents:

Module 1:
Concept of language - grammars and production rules- Chomsky hierarchy, Regular grammars, deterministic finite automata, non-determinism, conversion to deterministic automata ∈-closures, minimization of automata, regular expressions, regular sets, Pump lemma for regular sets closure properties of regular sets, decision properties for regular sets.

Module 2:
Context free languages, parse trees and ambiguity, reduction of CFGS, Chomsky and Griebach normal forms, push down Automata (PDA), non-determinism, CFLs and PDAs Pumping lemma for context free languages, Closure and decision properties of CFLs. Introduction of Turning machine.

Module 3:
Compilers and translators, phase structure of a typical compiler, Number of passes, ideas about lexical analysis, syntax analysis, code optimization and code generation, design of lexical analyzer. Lexical Analysis: Role of lexical analyzer, recognition of tokens, A language of specifying lexical analyzers, Design of a lexical analyzer generator and tool for study of lex.

Module 4:
Design of top-down parser, bottom-up parsing technique, LR parsing algorithm, Design of SLR, LALR, LR parsers.

Module 5:
Study of syntax-directed definitions and syntax-directed translation schemes as a notational frame work to specify the translations.

Text books:
1. Martin John, "Introduction to languages and the theory of compu tation", TM
2. Hotwani Hopcroft, Ullman, "Introduction to Automata Theory, Languages and computation", Pearson Education
3. Principles of Compiler Design: Aho A. V., Ullman J. D., Addison Wesley.

Reference books:
1. Michael Sipser, "Introduction to the theory of Computation", 3rd edition, Cengage Learning
2. Principles and practice of compiler writing: Aho, Sethi , Ullman , Addison Wesley
3. Compiler Design in C: Alan Holub , PHI
4. Crafting a compiler: Fischer and LeBlanc , Addison Wesley

Course Code:	**CSL 203**	**Course Title:**	**Computer System Organisation**
Category:	**Core**	**Credit Assigned**	

			L T P C 3 0 0 3
Pre-Requisite (if Any)	**None**	**Type of Course**	**Computer Science**

Course Outcomes:

Students will be able to:

1. Inspect the basic functional organization of a computing system.
2. Evaluate the performance of the machines based on the memories, I/O interfaces and pipelined architecture, etc.
3. Identify and interpret the implementation of memory chips, caches, modules, etc.
4. Analyse and compare CPU implementations, I/O methods, bus structures, modes of operations, ISA, etc.
5. Compare and contrast advanced architectures such as GPU in advanced digital systems.

Course Contents:

1. Addressing methods, their application in implementation of HLL constructs and data
2. Structures, instruction formats, expanding opcode method, subroutine linkage in PDP-11 and 68000, zero address machine such as HP3000.
3. Processing unit, bus architecture, execution of a complete instruction, sequencing of control signals, micro programmed control, microinstruction format, microinstruction sequencing, and bit slice concept.
4. Arithmetic, number representations and their operations, design of fast address, signed multiplication, Booth's Algorithm, bit-pair recording, division , floating point numbers and Operations, guard bits and rounding.
5. Main memory organization, various technologies used in memory design, higher order Memory design, multimodal memories and interleaving, cache memory, concept of cache memory, mapping functions, replacement algorithms. Input-output organization, I/O mapped I/O and memory mapped I/O, Direct Memory Access, interrupts and interrupt handling mechanisms, device identification, vectored interrupts, interrupt nesting, I/O Interfaces, synchronous vs. asynchronous data transfer, I/O channels.
6. Computer peripherals, I/O devices such as video terminals, video displays, graphic input devices, printers, magnetic disk, magnetic tape , CDROM systems.
7. RISC philosophy, pipelining, basic concepts in pipelining, delayed branch, branch prediction, data
8. dependency, influence of pipelining on instruction set design, multiple execution units, performance
9. considerations, basic concepts in parallel processing & classification of parallel architectures

Textbook:

1. Computer Organization , Hamacher, Carl V. et al, McGraw Hill
2. Structured Computer Organization , Tanenbaum A.S, Prentice Hall of India Ltd
3. Computer Organization & Design, The Hardware/ Software Interface, Patterson D. A
J. L. Second Edition. Harcourt, Hennessy Asia

Course Code	**CSP203**	**Course Title**	**AI/ML Workshop-I**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			0	**0**	**4**	**2**

Prerequisite **(If any)**	**-**	**Type of Course**	**Computer Science**

Course Outcomes:

On successful completion of the course, students shall be able to:

1. Understand the basics of the Python Programming Language.
2. Understand Lists, Dictionaries in Python.
3. Create functions and use recursions in python.
4. Learn about the NumPy, pandas, NLTK, Scipy libraries in Python

Course Contents:

Module 1:
Data Understanding – Characteristics; Sources; Types- structured unstructured, balanced, unbalanced; Supervised and Unsupervised analysis on data.

Module 2:
Introduction to Anaconda Platform, Google Colab – Installation and hands-on practice.

Module 3:
Basics of Python Programming- Introductory concepts - working environment, comparison with other programming languages, Basic syntax - expressions, types, statements, variables, etc., Conditional blocks: if, else, elif, For Loops in Python, Inbuilt containers – dictionary, tuples, lists, sets, maps, Python Functions: Defining a function, using a function, Recursion.

Module 4
Numpy - Numerical Python library to perform a variety of mathematical operations on arrays and matrices, Pandas Python Library, NLTK Python Library, Scipy – Python Library to perform scientific and technical computing on large sets of data.

Text Books:

1. Programming Python, 4th Edition -- Mark Lutz
2. Essential Python Reference, xth Edition -- David M. Beazley
3. Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib Book by Robert Johansson, Apress, 2019

Reference Books:

1. Python for Everybody, Charles Severance, 2016.
2. Effective Python, Brett Slatkin, Pearson, 2015.
3. Learn Python The Hard Way, Zed A. Shaw, Addison-Wesley, Third Edition

Course Code	HUL103	Course Title	Introduction to Entrepreneurship			
Category	Core	Credit Assigned	L	T	P	C
			3	0	0	3
Prerequisite (If any)	-	Type of Course	Basic Science			

Course Outcomes:

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

- Define entrepreneurship and its association with engineering profession, and create basic understanding of conceiving, creating, and managing an entrepreneurial venture.
- Identify various characteristics of Entrepreneurship, entrepreneurial culture and India's status with respect to entrepreneurship development.
- Recognize the essential complimentary nature of ethics/ values and creativity for entrepreneurship development
- Describe the MSMEs, SEZ and entrepreneurship development schemes and financial resources
- Demonstrate broad framework of opportunities for smart entrepreneurial efforts and start-up development

Course Contents:

Module 1:

Meaning and Importance, Evolution of term 'Entrepreneurship', Factors influencing entrepreneurship'. Characteristics and types of an entrepreneur, New generations of entrepreneurship viz. social entrepreneurship, Barriers to entrepreneurship.

Module 2:

Motivation theory, Achievement Theory, Culture and Society, Values / Ethics- national entrepreneurial culture, make in India concept and practices, creativity and entrepreneurship, Decision making and Problem Solving (steps in decision making), entrepreneurship and employment.

Module 3:

Special Economic Zone (Meaning, features & examples), Export-oriented units, Small Scale Industries, Make in India initiated by the government of India and Support for Industries. Scheme and packages, Financial and legal assistance for entrepreneurial development.

Module 4:

Opportunity for smart entrepreneurial efforts, branding, the management of property rights,

social value, technological innovation, online commerce, emerging markets and entrepreneurial solutions

Module 5:

Introduction to Idea Selection, Selection of the Product / Service, Phases of a Project, Project Report, and Contents of a Project Report. Case studies to provide real knowledge.

Basic Readings

4. Gordon, E., Natarajan, K., & Arora, A. (2009). Entrepreneurship development. Himalaya publishing house.
5. Megginson, W.L., Byrd, M.J. and Megginson, L.C., 2000. Small business management: an entrepreneur's guidebook.
6. Watson, J., Gatewood, E.J. and Lewis, K., 2014. A framework for assessing entrepreneurial outcomes: an international perspective. *International Journal of Gender and Entrepreneurship*.
7. Katz, J.A. and Green, R.P., 2021. Entrepreneurship Small Business.
8. Blomberg, J., Burrell, M., and Guest, G. An Ethnographic Approach to Design, Human-Computer Interaction Handbook, L. Erlbaum Associates Inc. Hillsdale, NJ, USA, 2003
9. Lerner, J. and Schoar, A. eds., 2010. *International differences in entrepreneurship*. University of Chicago Press.

References

5. Oughton, E. J., Comini, N., Foster, V., & Hall, J. W. (2022). Policy choices can help keep 4G and 5G universal broadband affordable. *Technological Forecasting and Social Change*, *176*, 121409.
10. Sterling, B. The Epic Struggle of the Internet of Things, Moscow: Strelka Press, 2014.
11. Castells, Manuel (2001): Internet Galaxy. Oxford University Press
6. J. Timmons Roberts and Amy Bellone Hite, Eds. The Globalization and Development Reader: Perspectives on Development and Global Change, Blackwell: London, 2007
7. Udyamita (in Hindi) by Dr. MMP. Akhouri and S.P Mishra, pub. By National Institute for Entrepreneurship and Small Business Development (NIESBUD), NSIC-PATC Campus, Okhla
8. Amartya Sen, Development as Freedom, Anchor Books: New York, 1999
9. IT Governance: How Top Performers Manage IT Decision Rights for Superior Results Kindle Edition by Peter Weill (Author), Jeanne W. Ross
10. Science Tec. Entrepreneur (A Bi Monthly Publication), Centre for Entrepreneurship Development, M.P (CEDMAP)

4th Semester

Course Code:	**CSL 422**	**Course Title:**	**Machine Learning**			
Category :	**Core**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Prerequisite (if Any) :	**None**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

Student will be able to:

1. Understand the working of various supervised and unsupervised machine learning models
2. Apply classification and regression models to solve real world problems.
3. Apply unsupervised learning to solve real world problems.
4. Evaluate the performance of various machine learning models.

Course Contents:

Module 1:

Introduction to ML, Linear Regression, Logistic Regression, Inductive Classification

Module 2:

Decision Trees: Representing concepts as decision trees. Recursive induction of decision trees. Picking the best splitting attribute: entropy and information gain. Searching for simple trees and computational complexity. Occam's razor. Overfitting, noisy data, and pruning.

Module 3

Artificial Neural Networks: Neurons and biological motivation. Linear threshold units.

Perceptrons: representational limitation and gradient descent training. Multilayer networks and Backpropagation.

Module 4:

SVM, Multiclass & Ordinal Classification, Kernel Methods, Bayesian Learning: Naive Bayes learning algorithm. Parameter smoothing. Bayes nets and Markov nets for representing dependencies. Hidden Markov Model, and Bayesian networks.

Module 5:

Clustering and unsupervised learning: Clustering. Hierarchical Aglomerative Clustering. k-means partitional clustering. Expectation maximization (EM) for soft clustering. Semi-supervised learning with EM using labelled and unlabelled data.

Module 6:

Evaluating hypothesis: Training and test splits, k-fold cross validation, confusion matrix, Estimating hypothesis accuracy, sample and true error.

Text Books:

1. Machine Learning, Tom Mitchell, McGraw Hill, 1997.

2. Ethem Alpaydin, Introduction to Machine Learning, PHI, 2016.

Reference Books:

1. T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning, 2e, 2008.

2. Christopher Bishop. Pattern Recognition and Machine Learning. 2e, 2006.

3. Richard O. Duda, Peter E. Hart, David G. Stork. Pattern classification, Wiley, New York, 2001.

Course Code:	**CSL 205**	**Course Title:**	**Design and Analysis of Algorithms**
Category:	**Core**	**Credit Assigned**	**L** | **T** | **P** | **C** **3** | **0** | **2** | **4**
Pre-Requisite (if Any)	**Advanced Data Structures , Mathematics**	**Type of Course**	**Computer Science Engineering**

Course Outcomes:

1. Student will be able to derive the recurrence relations for algorithms and analyze the performance of algorithms using asymptotic notations.
2. Student will be able to perform the amortized analysis and evaluate the cost of various operations on the data structure.
3. Student will be able to analyze and apply various algorithm design paradigms for real world applications. Also, evaluate the performance of algorithm based on various parameters.
4. Student will be able to apply and relate various algorithms to solve the problems based on Graphs.

Course Contents:

Module 1:

Mathematical foundations, summation of arithmetic and geometric series, n, n^2, bounding summations using integration, recurrence relations, and solutions of recurrence relations using technique of characteristic equation and generating functions.

Module 2:

Asymptotic notations of analysis of algorithms, analyzing control structures, worst case and average case analysis, amortized analysis, sorting algorithms such as selection sort, insertion sort, bubble sort, heap sort, lower bound proof, elementary and advanced data structures with operations on them and their time complexity.

Module 3:

Divide and conquer basic strategy, binary search, quick sort, merge sort, Fast Fourier Transform etc. Greedy method - basic strategy, application to job sequencing with deadlines problem, minimum cost spanning trees, single source shortest path etc.

Module 4:

Dynamic Programming basic strategy, multistage graphs, all pairs shortest path, single source shortest paths, optimal binary search trees, traveling salesman problem.

Module 5:

Basic Traversal and Search Techniques, breadth first search and depth first search, connected components. Backtracking basic strategy, 8-Queen's problem, graph colouring, Hamiltonian cycles etc. NP-hard and NP-complete problems, basic concepts, nondeterministic algorithms, NP-hard and NP-complete, Cook's Theorem, decision and optimization problems, polynomial reduction.

Text:

1. Introduction to Algorithms : Cormen T.H. et.al : Prentice Hall of India
2. Computer Algorithms : Horowitz, Sahani, Rajsekharan , Galgotia Publications Pvt.Ltd
3. Fundamentals of Algorithms : Brassard, Bratley , Prentice Hall

Course Code:	**CSL206**	**Course Title:**	**Software Engineering**
Category:	**Core**	**Credit Assigned**	**L** | **T** | **P** | **C** **3** | **0** | **0** | **3**
Pre-Requisite (if Any)	**Computer Programming(CSL 101)**	**Type of Course**	**Computer Science Engineering**

Course Outcomes:

1. Develop ideas and techniques for designing, developing, and modifying large software systems.
2. Discuss the Function-oriented and object-oriented modular design techniques, designing for re-use and maintainability.
3. Analyse specification and documentation, Verification and validation, Cost and quality metrics and estimation.
4. Illustrate to work in industry as a team member on a substantial project and used the management skill to handle the crucial project.

Course Contents:

Module 1:

Software Engineering Process & Management: Generic view, Capability Maturity Model, Process models-waterfall, evolutionary, incremental etc., unified process, agile view, project management, metrices estimation, project scheduling, risk management.

Module 2:

Software engineering Principles and Practice: Communication, planning and modelling practices, system engineering and modeling, business process engineering requirement analysis, system analysis- flow oriented and class oriented modeling using data modelling concepts.

Module 3:

Software Design Engineering: Design Concepts: Abstraction Architecture, pattern modularity, information hiding, design classes, refactoring etc., Design of web application, architectural design, component level design, user interface design.

Module 4:

Software Testing and Quality Management: Testing strategies, testing for object oriented software testing for web applications, validation testing etc. Black box testing, white box testing, Basis path testing. Testing for specialized environments, architectures and application. Quality concepts, quality assurance, software reviews, statistical quality assurance.

Module 5:

Software configuration management and advance topics : Elements of configuration management system, process configuration for web engineering, component-based development, clean room software engineering, formal methods, software reengineering, Software Maintenance

Text:
1. Software Engineering by Ian Sommerville ; Pearson Ed
2. Software Engineering: A Practitioner's Approach by Roger Pressman ; Tata-McGraw Hill

Course Code:	**CSL 207**	**Course Title:**	**Operating Systems**
Category:	**Core**	**Credit Assigned**	**L** **T** **P** **C** **3** **0** **2** **4**
Pre-Requisite (if Any)	**Data Structures (CSL 102)**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:
Students should be able to

1. Identify the structure and design issues of operating systems.
2. Summarise the concepts of process management and relate the underlying programming constructs.
3. Analyse and evaluate the memory management techniques, I/O management and file systems.
4. Implement general operating system concepts using modern operating systems like Unix and others.

Course Contents:

Module 1:
Introduction, basic h/w support necessary for modern operating systems - Services provided by OS, system programs and system calls - brief discussions of evolution of OS - real time and distributed systems: a brief overview of issues.
Processes and 3 levels of scheduling - process control block and context switch - goals of scheduling and different scheduling algorithms - threads: user-level and kernel level.

Module 2:
Process cooperation and synchronization - mutual exclusion and implementation - semaphores, conditional critical regions and monitors - classical inter - process communication problems - message passing.
Deadlocks and strategies for handling them - protection and security issues - access lists, Capabilities.

Module 3:
Memory management techniques - contiguous and non-contiguous - paging and segmentation - translation look-aside buffers (TLB) and overheads - virtual memory and demand paging- page faults and instruction restart - problems of large address spaces - page replacement algorithms and working sets - miscellaneous issues.

Module 4:
File systems, user interface - disk space management and space allocation strategies - examples from UNIX, DOS, Windows etc - directory structures - disk caching - file system

consistency and logs - disk arm scheduling strategies.

Module 5:
Linker and Loader - Concept of static and dynamic relocation, external symbols, design of linker, design of object file for different loading schemes.
Common Object file format - Structure of object file and executable file, section or segment headers, symbol table, concept of storage class, string various, data types line insert, character, arrays structures.
Device Drivers - Device programming, system drivers, non-system drivers, virtual drivers, Incorporation of driver routines, Basic device driver operation, character and block drivers.

Text:
1. Silberchatz & Galvin, "Operating System Concepts", Addison Wesley
2. Tanenbaum A, "Modern Operating Systems", PHI 2 nd Ed
3. William Stallings, "Operating Systems", Pearson Publications

Course Code:	**CSL 301**	**Course Title:**	**Database Management System**			
Category :	**CORE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Prerequisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Student will be able to design and develop database using ER model with various SQL constraints and apply normalization for consistency in database.
2. Student will be able to write queries using relational algebra, tuple and domain relational calculus, and SQL to retrieve information from database based on data centric applications.
3. Student will be able to analyze and apply the concept of storage management and query processing to fine tune the performance of database at the time of information retrieval.
4. Student will be able to analyze and apply the conception of transaction processing, concurrency control and recovery mechanism in database.

Course Contents:

Module 1:

Database system concepts and Architecture - concept of relational database, Relational data model, Relational algebra, SQL-the relational database standard, ER and EER model.

Module 2:

Database design theory - Functional dependencies and normalization, relational database design algorithms, practical database design and demoralization, Relational constants, programmatic ways for implementing constraints, triggers, Chase algorithm.

Module 3:

Physical database design - Concept of physical and logical hierarchy, storage structures like cluster, index organized table, partitions, various table storage parameters and block storage parameters, concept of index, B-trees, hash index, function index, bitmap index.

Module 4:

Process and memory management in database - Various types of tasks in database, database buffer management, log buffer management code reuse, concept of two tier and N-tier architecture, data dictionary and catalog information database recovery technique. Arier Algorithm for recovery.

Module 5:

Query optimization and performance tuning - Various techniques for query optimization, strong and weak equivalence, cost base optimization, Use of different storage structures in query optimization.

Module 6:

Transaction Processing - Transaction and system concepts, Desirable properties of transaction, Schedules and recoverability, serializability of schedules, concurrency control, lock base protocols and time stamp based protocols, read consistency.

Text Books:

1.Fundamentals of Database Systems : Elmasiri and Navathe, Addisson Wesley, 2000

2.Principles of Database Systems : Ullman , Golgotia Publications 1988

<table>
<tr><td>Course Code:</td><td>CSL214</td><td>Course Title:</td><td>Data Handling & Visualization</td></tr>
<tr><td>Category:</td><td>Core</td><td>Credit Assigned</td><td><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>1</td><td>0</td><td>2</td><td>2</td></tr></table></td></tr>
<tr><td>Pre-Requisite (if Any)</td><td>None</td><td>Type of Course</td><td>Computer Science & Engineering</td></tr>
<tr><td colspan="4">Course Outcomes:
By the completion of this course, students will be able to:
1. Design and create data visualizations.
2. Conduct exploratory data analysis using visualization.</td></tr>
</table>

3. Design and evaluate color palettes for visualization based on principles of perception.
4. Apply data transformations for visualization.
5. Acquire, analyze and provide observations from the data using different tools.

Course Contents:
Module 1: Data Acquisition

Gather information from different sources. Web APIs, Open Data Sources, Data APIs, Web Scrapping.

Module 2: Data Pre-processing and Transformation

Data Munging, Wrangling, Cast/Melt, Data imputation, Data Transformation (minmax, log transform, z-score transform etc.). Binning, Classing and Standardization. Outlier/Noise& Anomalies.

Module 3: Overview of Data Visualization, Advantages and applications, visualization design, visualization using Python/R: Scatter Plot, Bar Chart, Vertical & Horizontal Pie Chart and Coxcomb Plot, Line Chart, Area Chart, etc.

Module 4: Visualizing Amounts: Bar Plots, Grouped and Stacked Bars, Dot Plots and Heatmaps, Visualizing Distributions: Histograms and Density Plots- Visualizing a Single Distribution, Visualizing Multiple Distributions at the Same Time, Visualizing Distributions: Empirical Cumulative Distribution Functions and Q-Q Plots-Empirical Cumulative Distribution Functions.

Module 5: Tools for data handling like Power BI and other tools. Case studies.

Text book:

1. Sosulski, K. "Data Visualization Made Simple: Insights into Becoming Visual", New York: Routledge. CRC Press, 2018.
2. Claus Wilke, "Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures", 1st edition, O'Reilly Media Inc, 2019.

Reference Books:

1. Tamara Munzner, Visualization Analysis and Design (VAD), CRC press, 2014.
2. The Visual Display of Quantitative Information by Edward Tufte, Graphics Press.
3. Visualizing Data by Ben Fry, Oreilly

Course Code	**CSP204**	**Course Title**	**AI/ML Workshop-II**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			0	**0**	**4**	**2**
Prerequisite (If any)	**-**	**Type of Course**	**Computer Science**			
Course Outcomes:						

On successful completion of the course, students shall be able to:

1. Understand the usage of Scikit Learn python library for Machine Learning Algorithms
2. Apply the knowledge of Deep learning algorithms for various supervised and unsupervised problems.
3. Use of effective procedures and tools for data analytics using graphical outcomes.
4. Apply text representation techniques for Knowledge based systems.

Course Contents:

Module 1:
Machine Learning Library:
Introduction to Scikit-Learn library; Installation; Machine learning algorithm using Scikit Learn library.

Module 2:
Deep Learning Libraries:
Introduction to TensorFlow library, Installation; TensorFlow Basics — Tensor, Shape, Type, Sessions & Operators; The Computational Graph; Deep learning model using TensorFlow; Introduction to Pytorch; Introduction to Keras; Major differences between these libraries.

Module 3:
Basics of data visualization:
Creating and interpreting outputs of charts, plots and graphs using Matplotlib library.

Module 4:
Text Processing and Representation:
Introduction to natural language processing; Study and hands-on of word2vec and GloVe models for text representations using Gensim Library.

Text Books:

1. Neural Networks and Deep Learning A Textbook, Charu C. Aggarwal, Springer
2. Deep Learning with Keras: Implementing deep learning models and neural networks with the power of Python, Antonio Gulli, Sujit Pal
3. Sosulski, K. "Data Visualization Made Simple: Insights into Becoming Visual", New York: Routledge. CRC Press, 2018.
4. TensorFlow for Deep Learning, Bharath Ramsundar & Reza Bosagh Zadeh. O'Reilly Media.

Reference Books:

1. Deep Learning from Scratch ,Building with Python from First Principles, Seth Weidman, O'Reilly

3rd Year Syllabus

5th Semester

Course Code:	**CSL 421**	**Course Title:**	**Artificial Intelligence**			
Category :			**L**	**T**	**P**	**C**

	CORE	**Credit Assigned :**	**3**	**0**	**2**	**4**
Prerequisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Formulate problems so that exploratory search can be applied.
2. Apply and implement various search techniques to solve real world problems
3. Apply algorithms for designing games and solving constraint satisfaction problem
4. Represent knowledge using formal logic and apply algorithms to deuce conclusion
5. Design and develop practical algorithms for solving planning and uncertainty problems

Course Contents:

Module 1:

Introduction: What is AI? , History, Overview, Intelligent Agents, Performance Measure, Rationality, Structure of Agents, Problem-solving agents, Problem Formulation, Uninformed Search Strategies

Module2:

Informed (Heuristic) Search and Exploration, Greedy best first search, A* search, Memory bounded heuristic search, Heuristic functions, inventing admissible heuristic functions, Local Search algorithms, Hill-climbing, Simulated Annealing, Genetic Algorithms, Online search

Module 3:

Constraint Satisfaction Problems, Backtracking Search, variable and value ordering, constraint propagation, intelligent backtracking, local search for CSPs, Adversarial Search, Games, The minimax algorithm, Alpha-Beta pruning, Imperfect Real-Time Decisions, Games that include an Element of Chance.

Module 4:

Knowledge Based Agents, Logic, Propositional Logic, Inference, Equivalence, Validity and Satisfiability, Resolution, Forward and Backward Chaining, DPLL algorithm, Local search algorithms, First Order Logic, Models for first order logic, Symbols and Interpretations, Terms, Atomic sentences, complex sentences, Quantifiers, Inference in FOL, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution

Module 5:

Planning, Language of planning problems, planning with state-space search, forward and backward state-space search, Heuristics for state-space search, partial order planning, planning graphs, planning with propositional logic

Module 6:

Uncertainty, Handing uncertain knowledge, rational decisions, basics of probability, axioms of probability, inference using full joint distributions, independence, Baye's Rule and conditional independence, Bayesian networks, Semantics of Bayesian networks, Exact and Approximate inference in Bayesian Networks

Text:

1. Artificial Intelligence a Modern Approach : Russel and Norvig , Pearson Education, 2nd
2. Artificial Intelligence – A Practical Approach : Patterson , Tata McGraw Hill, 3rd

Course Code:	**CSL 433**	**Course Title:**	**Natural Language Processing**			
Category :	**CORE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Prerequisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Overview:

1. Analyse computational treatments of words, sounds, sentences, meanings, and conversations.
2. Analyse Knowledge-based and statistical approaches to language processing for syntax (language structures), semantics (language meaning), and pragmatics/discourse.
3. Develop models for Text Processing and part of Speech Tagging with Hidden Markov Model.
4. Develop models for information extraction, machine translation, automatic summarization, question answering, and interactive dialog systems.

Course Contents:

Module 1:

Introduction: NLP tasks in syntax, semantics, and pragmatics. Applications such as information extraction, question answering, and machine translation. The problem of ambiguity. The role of machine learning. Brief history of the field. Chomsky hierarchy, regular languages, and their limitations. Finite-state automata. Practical regular expressions for finding and counting language phenomena. A little morphology.

Module 2:

The role of language models. Simple N-gram models. Estimating parameters and smoothing. Evaluating language models. Key algorithmic tool: dynamic programming, first a simple example, then its use in optimal alignment of sequences. String edit operations, edit distance, and examples of use in spelling correction, and machine translation. Constituency, CFG definition, use and limitations. Chomsky Normal Form. Top-down parsing, bottom-up parsing, and the problems with each. The desirability of combining evidence from both directions.

Module 3:

Lexical syntax. Hidden Markov Models (Forward and Viterbi algorithms and EM training). Probabilistic language modeling and its applications. Markov models. N-grams. Estimating the probability of a word, and smoothing. Generative models of language. Their application to building an automatically-trained email spam filter, and automatically determining the language. The concept of parts-of-speech, examples, usage. The Penn Treebank and Brown Corpus. Probabilistic (weighted) finite state automata.

Module 4:

Spelling correction, Information Retrieval, Machine translation, Vector representation of words, Bag of words, CBOW, skip gram models, Recurrent Neural networks, Parts of speech tagging, Word sense disambiguation, Text summarization, Question answering systems.

Text Books:

1. Allen, James, Natural Language Understanding, Second Edition, Benjamin/Cumming, 1995.
2. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.
3. Jurafsky, Dan and Martin, James, Speech and Language Processing, Second Edition, Prentice Hall, 2008.
4. Manning, Christopher and Heinrich, Schutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.

Reference Books:

1. Radford, Andrew et. al., Linguistics, An Introduction, Cambridge University Press, 1999.
2. Journals: Computational Linguistics, Natural Language Engineering, Machine Learning, Machine Translation, Artificial Intelligence.
3. Conferences : Annual Meeting of the Association of Computational Linguistics (ACL), Computational Linguistics (COLING), European ACL (EACL), Empirical Methods in NLP (EMNLP), Annual Meeting of the Special Interest Group in Information Retrieval (SIGIR), Human Language Technology (HLT).

Course Code	CSL308	Course Title	Computer Vision Techniques			
Category	CORE	Credit Assigned	L	T	P	C
			3	0	2	4
Prerequisite (If any)	---	Type of Course	Computer Science & Engineering			

Course Outcomes:

1. identify basic concepts, models and methods in the field of computer vision
2. assess which methods to use for solving a given problem, and analyse the accuracy of the methods
3. develop and apply computer vision techniques for solving real world problems
4. choose appropriate image processing methods for image filtering, image restoration, image reconstruction, segmentation, classification and representation

Course Contents:

Module 1:
Digital Image Formation and low-level processing: Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.

Module 2:
Depth estimation and Multi-camera views: Perspective, Binocular Stereopsis: Camera and Epipolar Geometry; Homography, Rectification, DLT, RANSAC, 3-D reconstruction framework; Auto-calibration.

Module 3:
Feature Extraction: Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT.
Image Segmentation: Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection.

Module 4:
Motion Analysis: Background Subtraction and Modeling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

Module 5:
Shape from X: Light at Surfaces; Phong Model; Reflectance Map; Albedo estimation; Photometric Stereo; Use of Surface Smoothness Constraint; Shape from Texture, color, motion and edges.

Text Books:

1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.
2. Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 200

Reference Books:

1. Christopher M. Bishop; Pattern Recognition and Machine Learning, Springer, 2006.
2. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004.

3. K. Fukunaga; Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990.
4. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison- Wesley, 1992.

Course Code :	**CSL302**	**Course Title :**	**Computer Networks**			
Category :	**CORE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Prerequisite (if Any) : Nil	**None**	**Type of Course :**	**Computer Science & Engineering**			

Course Outcomes:

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

1. Analyse various issues and their solutions at different layers of network architecture
2. Design and develop various networking algorithms
3. Apply networking protocols on a given network to analyze their working
4. Apply networking concepts to build real world networking systems using most important protocols in use today

Course Contents:

Module 1:
Introduction to Computer Networks, Network Architecture: Layering and Protocol, Internet architecture, Implementing Network Software: Application Programming Interface (Socket), Delay x bandwidth product.

Module 2:

Classes of Links, Framing, Error Detection: cyclic redundancy check, Internet checksum algorithm. Reliable transmission: Stop-and-wait, Sliding Window, Ethernet and Multiple Access Networks (802.3), Wi-Fi/802.11

Module 3:

Datagram and virtual circuit switching, Bridges and LAN switches, Basic Internetworking (IP): Internetwork, Global addresses, Datagram forwarding in IP, Subnetting and Classless addressing,

Module 4:

Address Translation (ARP), Host Configuration (DHCP), Error Reporting (ICMP), Virtual Networks and tunnels, Distance-vector (RIP), Link-state (OSPF), Routing areas, Interdomain routing (BGP).

Module 5:

Simple demultiplexer (UDP), Reliable byte stream (TCP): End-to-end issues, segment format, Connection establishment and termination, Sliding window, Triggering transmission and Adaptive retransmission, TCP Congestion Control: Additive increase/ Multiplicative decrease, Slow start, Fast retransmission and fast recovery. Resource allocation in TCP

Module 6:

Introduction to applications and related Protocols: Electronic mail (SMTP, MIME, IMAP), World Wide Web (HTTP), Name service (DNS).

Text Books:

1. Larry L. Peterson, Bruce S. Davie, "Computers Networks: A systems approach", Morgan Kaufmann, 5th Edition.

Reference Books:

1. Tanenbaum A. S, "Computer Networks", PHI
2. James F. Kurose and Keith W. Ross: Computer Networking: A Top-Down Approach Featuring the Internet, 3rd Edition.
3. William Stallings, "Data and Computer Communications", PHI 6th Edition
4. Behrouz A Forouzan, "Data Communication and Networking", 4th Edition
5. Simon Haykin, "Communication Systems", John Wiley 4th Edition
6. Douglas Comer, "Computer Networks and Internets", Addison Wesley 2nd Edition
7. Peterson, Simon, "Computer Networks: A Systems Approach", Pearson Education, Asia

6th Semester

Course Code	**CSL454**	**Course Title:**	**Reinforcement Learning**
Category:	**ELECTIVE**	**Credit Assigned**	**L** | **T** | **P** | **C** **3** | **0** | **2** | **4**
Pre-Requisite (if Any)	**ML (CSL 422)**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:

1. To incorporate and analyse various elements & characteristics of reinforcement learning
2. To Formulate decision problems and set up, run, and analyse computational experiments

3. To Apply reinforcement learning for various real-life problems

Course Contents:

Module 1:

Course logistics and overview. Origin and history of Reinforcement Learning research. Its connections with other related fields and with different branches of machine learning. Brush up of Probability concepts - Axioms of probability, concepts of random variables, PMF, PDFs, CDFs, Expectation. Concepts of joint and multiple random variables, joint, conditional and marginal distributions. Correlation and independence.

Module 2:

Introduction to RL terminology, Markov property, Markov chains, Markov reward process (MRP). Introduction to and proof of Bellman equations for MRPs along with proof of existence of solution to Bellman equations in MRP. Introduction to Markov decision process (MDP), state and action value functions, Bellman expectation equations, optimality of value functions and policies, Bellman optimality equations.

Module 3:

Overview of dynamic programming for MDP, definition and formulation of planning in MDPs, principle of optimality, iterative policy evaluation, policy iteration, value iteration, Banach fixed point theorem, proof of contraction mapping property of Bellman expectation and optimality operators, proof of convergence of policy evaluation and value iteration algorithms, DP extensions.

Module 4:

Overview of Monte Carlo methods for model free RL, First visit and every visit Monte Carlo, Monte Carlo control, On policy and off policy learning, Importance sampling. Getting started with the function approximation methods, Revisiting risk minimization, gradient descent from Machine Learning, Gradient MC and Semi-gradient TD(0) algorithms, Eligibility trace for function approximation, Afterstates, Control with function approximation, Least squares, Experience replay in deep Q-Networks.

Module 5:

Getting started with policy gradient methods, Log-derivative trick, Naive REINFORCE algorithm, bias and variance in Reinforcement Learning, Reducing variance in policy gradient estimates, baselines, advantage function, actor-critic methods.

Text Books:

1. Reinforcement Learning: An Introduction, Sutton and Barto, 2nd Edition.

2. Algorithms for Reinforcement Learning. C. Szepesvari. Morgan and Claypool Publishers, 2010

Reference Books:

1. Reinforcement Learning: State-of-the-Art, Marco Wiering and Martijn van Otterlo, Eds.
2. Deep Learning, Ian Goodfellow, Yoshua Bengio, and Aaron Courville.

Course Code:	**CSL 446**	**Course Title:**	**Neural Networks and Deep Learning**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Prerequisite (if Any) :	**Machine Learning**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Ability to understand the fundamental concepts of deep neural networks and underlying error convergence algorithms such as backpropagation and gradient descent.
2. Acquire the depth knowledge of various deep models, architectural aspects, comparative analysis and their applications.
3. Ability to implement, analyze, optimize deep models or the computer vision and natural language processing tasks.
4. Ability to apply the advancements of deep models for various problems in different domain.

Course Contents:

Course Contents:

Module 1:

Neural network working, Introduction of deep neural network. Backpropagation and Gradient Descent algorithms in deep networks. Role of vectorization in various operations in deep learning. Comparisons of shallow and deeper networks.

Module 2:

Bias variance trade-off, overfitting and its remedies, regularization: L1, L2, and dropout. Hyperparameter and its tuning in converging deep networks. Exponentially weighted moving averages, Gradient descent optimization algorithms and its importance such as momentum, RMSProp, Adam's algorithm, Nesterov, Adagrad, Adadelta, and Adamax Introduction and importance of domain knowledge in deep learning.

Module 3:

Convolutional neural networks, its architecture, deep CNN, parameter sharing, and applications. Generalized parameters computation. Introduction and parameters computation of several pretrained CNN architectures such as LeNet, AlexNet, Inception and its successors, ResNet architectures. Depth-wise pooling: 1x1 convolution.

Module 4:
Recurrent neural networks and its architectural variants such as LSTM, GRU. Architectural aspects and applications. Comparative analysis of RNN, LSTM, GRU.
Word representation: character and word level embedding, CBoW and skip gram model, Word2Vec, negative sampling, GloVe, Attention Model.
Hybridization of deep models.

Module 5:
Applications/Case studies of deep learning models
Computer vision: Introduction to object detection, YOLO, regression formulation, process, network design, and loss function.
Natural language processing: Sentiment Analysis.

Research direction in deep learning. Advanced convolutional: tiled convolution, dilated convolution, transpose or deconvolution, networks in networks. Introduction to bidirectional, multidimensional, stacked sequence models.

Text Books:

1. Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville
2. Deep Learning: A Practitioner's Approach, by Adam Gibson and Josh Patterson

Reference Books:

1. Neural Networks and Deep Learning by Michael Nielsen.

Course Code:	**CSL317**	**Course Title:**	**Parallel and Distributed Computing**			
Category:	**Core**	**Credit Assigned:**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Prerequisite (If any):	**NONE**	**Type of Course:**	**Computer Science and Engineering**			

Course Outcomes:

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

1. Understand fundamental concepts in parallel computing.
2. Identify and implement common parallel computing patterns on shared and distributed memory architectures.
3. Compare efficiency and scalability of a parallel algorithm/application.
4. Think parallel and solve different computing patterns effectively.

Course Contents:

Module 1: Introduction to Parallel Computing

Overview of parallel processing landscape: why and how, microprocessor trend over last 6 decades, types of parallelism, Flynn's taxonomy, and brief overview of parallel architectures – shared memory and distributed memory. Basic concepts in parallel processing: formal definition of parallelism, concepts of work, speedup, efficiency, overhead, strong and weak scaling (Amdahl's law, Gustafson's law), practical considerations using parallel reduction and parallel prefix.

Module 2: Shared memory Programming using OpenMP

Shared memory and shared address space, multi-core architecture, data and task parallelism, OpenMP Fork-Join Model, Pragmas, race conditions, hyperthreading, False sharing, schedules, and data regions, Parallel Pi calculation.

Module 3: Shared memory programming using CUDA/OpenCL

GPGPU architecture, GPU programming model, Thread organization, parallel Vector addition,

Matrix multiplication, GPU memory model, GPU memory optimization.

Module 4: Distributed memory Programming using MPI

Multi-node computing architecture, MPI programming model, performance limiting factors, MPI send receive functions, MPI ping-pong and ring program, MPI broadcast, MPI scatter, MPI gather, MPI reduce.

Module 5: Parallel computing patterns on shared and distributed memory

Reduction, prefix sum, odd-even sort, radix sort, histogram, convolution, matrix transpose, matrix multiplication.

Text Books:

1. "Parallel Programming in C with MPI and OpenMP", Michael Jay Quinn" McGraw-Hill, 2003.
2. "Introduction to Parallel Computing", Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, 2nd Edition, Addison-Wesley, 2003.

Reference Books:

1. "Programming Massively Parallel Processors A Hands-on Approach", Wen-Mei W Hwu, David B Kirk, 4th Edition, Morgann Kaufmann, 2012.
2. "Introduction to Parallel Computing From Algorithms to Programming on State-of-the-Art Platforms", Roman Trobec, Bostjan Slivnik, Patricio Bulić, Borut Robic, 1st Edition, Springer International Publishing, 2018.
3. "Structured Parallel Programming: Patterns for Efficient Computation", M. McCool, J. Reinders, A. Robison, Morgan Kaufmann, 2012.

Course Code:	CSL 318	Course Title:	Optimization Techniques in ML			
Category:	Core	Credit Assigned:	L	T	P	C
			3	0	0	3
Prerequisite (if Any):	None	Type of Course:	Computer Science and Engineering			

Course Outcomes:

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

1. Develop foundational knowledge of optimization techniques and their applications in machine learning.
2. Formulate and solve different optimization problems relevant to machine learning.
3. Analyze and compare various optimization algorithms, focusing on convergence and efficiency.
4. Implement and apply optimization techniques using programming libraries and assess their performance in ML models.
5. Gain insight into advanced optimization methods and recent research trends in machine

Course Contents:

Module 1:

Introduction to Optimization in Machine Learning: Overview of optimization in ML, types of optimization problems, loss functions, and their role in ML, examples in supervised and unsupervised learning.

Module 2:

Convex Optimization Techniques: Convex sets and functions, gradient descent (GD) and variants (batch, stochastic, mini-batch GD), convergence analysis for convex functions.

Module 3:

Constrained Optimization and Regularization: Constrained optimization, Lagrange multipliers, KKT conditions, duality theory, regularization techniques (L1, L2), impact on optimization and overfitting.

Module 4:

Stochastic and Advanced Gradient-Based Methods: Stochastic Gradient Descent (SGD) and variants (Adam, RMSprop, Adagrad), Momentum-based optimization (Momentum, Nesterov Accelerated Gradient).

Module 5:

Large-scale and Recent Advances in Optimization: Distributed optimization frameworks (e.g., Parameter Server, federated learning), large-scale optimization challenges, hyperparameter optimization (Grid Search, Random Search, Bayesian Optimization), AutoML, neural architecture search (NAS).

Text Books:

1. *"Convex Optimization",* Stephen Boyd & Lieven Vandenberghe, Cambridge University Press, 2004.
2. "*Numerical Optimization*", Nocedal, Jorge and Wright, Stephen J, Springer, 1999.

References:

1. "*Convex Optimization Algorithms*", Dmitri P. Bertsekas, first edition, Athena Scientific, 2015.
2. "*Deep Learning*", Goodfellow, I., Bengio, Y., & Courville, A, MIT Press, 2016.