COEP Technological University

Department of Computer Science and Engineering

Curriculum Structure

S. Y. B. Tech. Artificial Intelligence and Machine Learning (AIML)

List of Abbreviations

Abbreviation	Title	Credits	Percentage
BSC	Basic Science Course	14	8.75
ESC	Engineering Science Course	16	10.00
PCC	Programme Core Course (PCC)	55	34.37
PEC	Programme Elective Course (PEC)	17	10.62
OE	Open Elective - other than particular program	6	3.75
MDM	Multidisciplinary Minor (MDM)	14	8.75
VSEC	Vocational and Skill Enhancement Course (VSEC)	12	7.50
HSMC	Humanities Social Science and Management	5	3.13
AEC	Ability Enhancement Course	4	2.50
IKS	Indian Knowledge System (IKS)	2	1.25
VEC	Value Education Course (VEC)	2	1.25
RM	Research Methodology (RM)	2	1.25
OJT	Internship	7	4.38
CEA	Community Engagement Activity (CEA)/Field Project	2	1.25
CCA	Co-curricular & Extracurricular Activities (CCA)	2	1.25
	Total	**160**	**100**

Legend	
L	Lecture
T	Tutorial
P	Practical
S	Self-study
Cr	Credits
TA	Teachers' Assessment
MSE	Mid-Semester Evaluation
ESE	End-Semester-Evaluation
CIE	Continuous-Internal-Evaluation

Second Year B. Tech.
Artificial Intelligence and Machine Learning (AIML)
Semester – III

Sr. No.	Course Type	Couse Code	Course Name	L	T	P	S	Cr	Evaluation Scheme (Weightages in %)				
									Theory			Laboratory	
									TA	MSE	ESE	CIE	ESE
01	PCC		Data Engineering	2	-	2	1	3	20	30	50	100	--
02	PCC		Programming language Paradigms	3	-	2	1	4	20	30	50	100	--
03	PCC		Data Structures and Complexity Theory	3	-	2	1	4	20	30	50	100	--
04	OE		Open Elective-1	2	-	-	1	2	20	30	50	--	--
05	AEC		Languages	2	-	-	-	2	--	--	--	100	--
06	VEC		Entrepreneurship	1	-	-	2	1	--	--	--	100	--
07	HSMC		Constitution of India	2	-	-	-	1	--	--	--	100	--
08	CEA		Community Engagement Activity (CEA)/Field Project	-	-	-	-	2	--	--	--	100	--
			Total	**15**	-	**04**	**06**	**19**					

Semester – IV

Sr. No.	Course Type	Couse Code	Course Name	L	T	P	S	Cr	Evaluation Scheme (Weightages in %)				
									Theory			Laboratory	
									TA	MSE	ESE	CIE	ESE
01	PCC		Machine Learning	3	-	2	1	4	20	30	50	100	--
02	PCC		Database Management Systems	3	-	-	1	4	20	30	50	100	--
03	PCC		Automata Theory	3	-	-	1	3	20	30	50	--	--
04	OE		Open Elective-2	2	-	-	1	2	20	30	50	--	--
05	MDM		Multidisciplinary Minor-1	3	-	-	1	3	20	30	50	--	--
06	VSEC		Development Tools Laboratory	-	-	2	-	1	--	--	--	100	--
07	VEC		Environmental Studies	1	-	-	2	1	--	--	--	100	--
08	HSMC		Design Thinking	1	-	-	1	1	--	--	--	100	
09	HSMC		Economics	2	-	-	-	2	--	--	--	100	--
			Total	**18**	**01**	**04**	**08**	**21**					

**** Exit option to qualify for UG Diploma:** Two courses of 3 Credits each on AIML

Second Year B. Tech.
Lateral Entry
Artificial Intelligence and Machine Learning (AIML)
Semester – III

Sr. No.	Course Type	Couse Code	Course Name	L	T	P	S	Cr	Evaluation Scheme (Weightages in %)				
									Theory			Laboratory	
									TA	MSE	ESE	CIE	ESE
01	PCC		Data Engineering	2	-	2	1	3	20	30	50	100	--
02	PCC		Programming Language Paradigms	3	-	2	1	4	20	30	50	100	--
03	PCC		Data Structures and Complexity Theory	3	-	2	1	4	20	30	50	100	--
04	PCC		Matrices, Differential Calculus and Probability	3	-	-	1	3	30	20	50	--	--
05	OE		Open Elective-1	2	-	-	1	2	20	30	50	--	--
06	AEC		Languages	2	-	-	-	2	--	--	--	100	--
07	VEC		Entrepreneurship	1	-	-	2	1	--	--	--	100	--
08	HSMC		Constitution of India	2	-	-	-	1	--	--	--	100	--
			Total	**18**	-	**04**	**07**	**20**					

Semester – IV

Sr. No.	Course Type	Couse Code	Course Name	L	T	P	S	Cr	Evaluation Scheme (Weightages in %)				
									Theory			Laboratory	
									TA	MSE	ESE	CIE	ESE
01	PCC		Machine Learning	3	-	2	1	4	20	30	50	100	--
02	PCC		Database Management Systems	3	-	-	1	4	20	30	50	100	--
03	PCC		Automata Theory	3	-	-	1	3	20	30	50	--	--
04	PCC		Discrete Structures	2	-	-	1	2	20	30	50		
05	OE		Open Elective-2	2	-	-	1	2	20	30	50	--	--
06	MDM		Multidisciplinary Minor-1	3	-	-	1	3	20	30	50	--	--
07	VSEC		Development Tools Laboratory	-	-	2	-	1	--	--	--	100	--
08	VEC		Environmental Studies	1	-	-	2	1	--	--	--	100	--
09	HSMC		Design Thinking	1	-	-	1	1	--	--	--	100	
10	HSMC		Economics	2	-	-	-	2	--	--	--	100	--
11	IKS		Communication Skills	2	-	-	-	2	--	--	--	100	
			Total	**22**	**01**	**04**	**09**	**25**					

**** Exit option to qualify for UG Diploma:** Two courses of 3 Credits each AIML

(PCC) Data Engineering

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs/ Week	Laboratory: CIE: 100 Marks

Course outcomes

1. Understand the fundamentals of data science and analytics.
2. Learn data preprocessing and visualization techniques.
3. Apply statistical and machine learning methods to solve problems.
4. Develop practical skills using Python and data science libraries.
5. Interpret data-driven results for engineering applications.

Course content

Introduction to Data Engineering **[6 Hrs]**

Introduction and Importance of Data Engineering, Types of data, Sources of data, Life cycle of data science, Roles in data science- Data analyst, Data scientist, Data Engineer, Challenges, Applications, Big data concepts.

Data Preprocessing **[6 Hrs]**

Introduction to data preprocessing, Statistical data type, Data cleaning techniques, Data transformation, Data integration, Data normalization and standardization, Data preprocessing with Pandas- series and data frame in pandas, Loading CSV and Excel files, Data Inspection and Exploration, Data Selection and Filtering, Handling Missing and Duplicate Data, Data Transformation, Data Aggregation and Integration

Data transformation and Integration **[6 Hrs]**

Data normalization and standardization, aggregation and filtering, formatting and type conversion of data, Feature extraction basics, Data enrichment concepts, Data mapping techniques, Data discretization and binning, Data reduction techniques, Concept and architecture of ETL, Data extraction from multiple sources, Data transformation techniques, ETL workflow and scheduling, ETL vs ELT, Error handling in ETL pipelines, data integration, Data flow and pipeline concepts.

Data Visualization and Statistics **[6 Hrs]**

Descriptive and Inferential statistics, Data Visualization using matplotlib, Types of Charts and Graphs- Bar chart, Pie chart, Histogram, Scatter plot, Line graph, Jitter plot, Seaborn Libraries- Heatmap, Violine plot, Pair plot.

Data Modelling **[6 Hrs]**

Introduction to machine learning, Types of machine learning, Regression algorithm, Classification algorithm, Clustering algorithm, Model Evaluation Matrices, Building hypothesis model, p-values, Type I/II errors, t-tests & ANOVA- One-sample, two-sample, one-way and two-way ANOVA, Chi-squared tests Goodness of fit, independence tests for categorical data.

Data science tools and techniques **[6 Hrs]**

Over-fitting, under-fitting and model evaluation metrics: Bias-variance tradeoff, MSE, RMSE, Regularization Techniques (L1 & L2), Over-fitting, under-fitting and model evaluation metrics: Bias-variance tradeoff, MSE, RMSE, Regularization Techniques (L1 & L2) Introduction to Scikit-learn, Model Training and Testing, Introduction to TensorFlow and Keras, data warehousing concepts, Cloud platform for data science.

Self-study **[12 Hrs]**

Data Science Applications in Engineering- Healthcare, Agriculture, Smart cities, Data privacy and security, Bias and fairness in AI, case studies in Data science.

Textbooks

1. "Mining of Massive Datasets", Jure Leskovec, Anand Rajaraman, and Jeffery David Ullman, Cambridge University Press, 2 edition (13 November 2014), ISBN-10: 1107077230, ISBN-13: 978-1107077232
2. “Foundation of Data Science”, Avrim Blum, John Hopcraft, and Ravindran Kannan, Hindustan book agency, (online free version), January 2020, ISBN-10: 9386279800

Reference Books

1. “Foundations of Data Science”, Avrim Blum, John Hopcroft, and Ravindran Kannan, Hindustan Book Agency, (online free version) January 2020, ISBN-10: 9386279800
2. “Introduction to Machine Learning”, Ethem Alpaydin, PHI Learning Pvt Ltd, Third edition, 2016, ISBN-978- 81-203-5078-6

Web references

1. https://www.kaggle.com/code/imoore/intro-to-exploratory-data-analysis-eda-in-python
2. https://www.tensorflow.org/resources/learn-ml
3. https://matplotlib.org/stable/index.html?utm_source
4. https://developers.google.com/machine-learning/crash-course?utm_source

Laboratory assignments

1. Real-world datasets often contain missing values, duplicate records, inconsistent formats, and noisy data that affect the performance of data analysis and machine learning models. Develop a data preprocessing system using Python to clean, transform, and prepare a dataset for further analysis.
2. Organizations collect large amounts of structured data that must be analyzed efficiently for decision making. Develop a Python program using the Pandas library to import, manipulate, analyze, and visualize tabular datasets.
3. Develop a Python-based data analysis system using the Pandas library to perform data cleaning, filtering, grouping, aggregation, and reporting.
4. Develop a Python program using the Matplotlib library to create various graphical representations of datasets for effective data analysis and interpretation.
5. Understanding relationships between variables and analyzing data distribution are important tasks in data science. Develop a Python program using Seaborn to generate Heatmaps and Violin Plots for analyzing correlations and distributions in datasets.
6. Organizations use predictive models to analyze historical data and forecast future outcomes. Develop a machine learning-based data modelling system using Python to train, test, and evaluate predictive models on a real-world dataset.
7. Develop machine learning models and analyze the effects of overfitting and underfitting using training and testing datasets.
8. Machine learning models may overfit due to complex features. Implement L1 and L2 regularization techniques to reduce overfitting and improve model performance.
9. Develop a simple neural network model using TensorFlow and Keras for classification tasks.
10. Develop a Python program to calculate conditional probabilities and apply Bayes’ theorem for probabilistic decision making.
11. Download a CSV dataset from any open-source platform and perform various data transformation and integration tasks to prepare the dataset for analytics and reporting.

This list serves as a guideline and is intended to be continuously refined by the instructor.

Laboratory instructions or disclaimers if any

(PCC) Programming Language Paradigms

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs./ Week	Laboratory: CIE: 100 Marks
Self - study: 1 Hr./week	

Course outcomes

1. Explain fundamental concepts of different programming paradigms
2. Design a class hierarchy using object-oriented thinking for a given problem.
3. Compare, annotate, and comment on various object-oriented programming concepts.
4. Create object-oriented application code for a given problem
5. Demonstrate ability to write simple programs using exception handling and event driven programming concepts.

Course content

Introduction to Principles of Programming Languages:	**[5 Hrs]**

Reasons for Studying Concepts of Programming Languages. Programming Domains. Language Evaluation Criteria. Influences on Language Design: computer architecture. Language Categories. Language Design Trade-Offs.

Implementation Methods:	**[7 Hrs]**

Compilation, Interpretation, Hybrid Implementation, Preprocessors. Programming Environments. Evolution of the Major Programming Languages (Pseudocodes, C, C++, Python, Java).

Language Description: Syntactic Structure	**[8 Hrs]**

Expression Notations, Abstract Syntax Trees (AST), Lexical Syntax, Context-Free Grammars (CFG), top-down parsing, Bottom-up parsing. Names, Bindings, and Scopes, Type checking, Strong Typing. Type Conversions.

Programming Paradigms:	**[6 Hrs]**

Various programming paradigms: Procedural, object-oriented, logic and functional, concurrent programming. Classes, Objects, Methods; Data types provided by OO languages; Abstract Data Types.

Introduction to OOP:	**[8 Hrs]**

Characteristics of OOP: Encapsulation, Polymorphism, Abstraction, Constructors, Destructors, Access specifications: Private, public, protected members. Inheritance, Types of inheritance: single, multilevel, multiple, hierarchical and hybrid, Class hierarchies.

Exception Handling & Multi-Threaded Programming:	**[6 Hrs]**

Exception handling, Exception types, Concurrent Programming, Basic Concepts of Concurrent Programming, Threads, Multi treading.

Self-study	**[12Hrs]**

Unified Process –UML(Unified Modeling Language) diagram, use case, class diagrams, interaction diagrams, state diagrams – activity diagrams – package, component and deployment diagrams. Design: Unified Modelling language; use case diagrams; Class Diagrams

Textbooks

1. Ravi Sethi, “Programming Languages Concepts and Constructs”,2nd edition, Pearson

Education,ISBN-0201590654/9780201590654
2. Schesta R., "Concepts Of Programming Languages", 4th Edition, Pearson Education, ISBN- 81-7808-161-X
3. Cay S Horstmann and Gary Cornell, Core Java Vol-1 and Vol-2, 9th Edition, Pearson Education India, ISBN-10: 9332518904 and 9332518890
4. Ronald Leach, Object-Oriented Design and Programming with C++: Your Hands-On Guide to C++ , Academic Press, ISBN: 9781483214122

Reference Books

1. Ghezzi C, Milano P., Jazayeri M., "Programming Languages Concepts", 3rd Edition, John Wiley and Sons Pvt. Ltd (WSE), ISBN - 0195113063
2. Roosta S., "Foundations of Programming Languages", Thomson, Brooke/Cole, ISBN 981- 243-141-1
3. Herbert Schilt, “JAVA Complete Reference”, 7th Edition, Tata McGraw Hill, ISBN: 9780070636774
4. E. Balagurusamy, Object Oriented Programming with C++, 6th Edition, McGraw Hill, ISBN10: 125902993X
5. E. Balagurusamy,Programming with JAVA - A Primer, 3rd Edition,McGraw Hill,ISBN 0-07-0617139

Web references

1. https://nptel.ac.in/courses/106102067
2. https://onlinecourses.nptel.ac.in/e-learning/preview/noc25_cs34

Laboratory assignments

1. Design a banking management system to maintain customer account information. The system should support functionalities such as account creation, deposit, withdrawal, balance inquiry, transaction history maintenance, and validation for insufficient balance conditions. Represent the complete solution using suitable classes and relationships.

2. Write a program to implement the following. Create a base class called STUDENT consisting of student name and roll number. Create two child classes:
   a. ACADEMIC with member marks
   b. SPORTS with member sports_grade.
   Inherit both classes into a class RESULT containing student name, roll number, marks, and sports grade. Display the complete student result information using suitable methods.
   a. Write a script to display:
   b. Current system date and time
   c. Current month and year
   d. Number of days in the current month
   e. Current week number of the year
   f. Current weekday name
   g. Day number of the year
   h. Current date of the month and current day of the week.
3. Design an employee payroll management system for an organization. The system should manage employee profiles, departments, attendance records, incentives, deductions, and monthly salary generation. Include role-based employee categories such as manager, technical staff, and administrative staff.

4. Create a transportation reservation system for railway or airline booking. The system should support passenger registration, seat availability checking,

reservation confirmation, ticket cancellation, fare calculation, and waiting list management with proper exception handling.

5. Design a library automation system capable of handling book issue, return, renewal, fine calculation, and inventory management. The system should maintain separate records for students, faculty members, and librarians along with transaction logs.

6. Construct a hospital management system to manage doctor details, patient records, appointments, medical history, laboratory reports, and billing information. The system should also support emergency patient handling and appointment scheduling conflicts.

7. Develop a menu-driven matrix and statistical analysis system capable of performing operations such as matrix addition, subtraction, multiplication, transpose, determinant calculation, mean, median, mode, and standard deviation for datasets entered by the user.

8. Construct a UML Class Diagram for an Online Food Delivery System. The system should include entities such as Customer, Restaurant, Delivery Partner, Order, Payment, and Feedback. Clearly represent associations, multiplicity, inheritance, and aggregation relationships.

9. Construct a State Diagram for an Automated Vending Machine System. The diagram should represent states such as idle, product selection, payment processing, item dispensing, insufficient balance handling, cancellation, and transaction completion with appropriate transitions.

10. Design a smart city parking management system capable of handling vehicle entry/exit, slot allocation, parking duration calculation, automated billing, digital payment, and parking availability monitoring. Represent the complete workflow using suitable UML diagrams and explain the relationships between system components.

    This list serves as a guideline and is intended to be continuously refined by the instructor.

(PCC) Data Structures and Complexity Theory

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs./ Week	Laboratory: CIE: 100 Marks
Self - study: 1 Hr./week	

Course outcomes

1. Understand the fundamental concepts of various data structures (e.g., arrays, linked lists, stacks, queues, trees, graphs, hash tables).
2. Understand the concepts of linear and non-linear data structure and their representation
3. Demonstrate the ability to write abstract data types and the use of basic data structures in solving problems.
4. Discuss, compare and implement algorithms for various operations in different
5. implementations of stacks and queues, tree, graph, matrices, heap, etc.
6. Analyze the time complexity of different searching, sorting, and traversal algorithms.

Course content

Fundamental Concepts	**[6 Hrs]**

Introduction to Data Structures: Data, Data Object, Data types, Abstract Data Types (ADT), Data structures, Concept of primitive and non-primitive, linear and non-linear data structures. Introduction to Algorithms: Definition and Characteristics of an algorithm, Algorithm Specification, Performance Analysis- Time and space complexity, Asymptotic notations, Best, Average and worst cases. Introduction to Data representation and files: Text and binary files, use of various libraries for handling files.

Linear Data Structures and their storage representation	**[6 Hrs]**

Stack and queue as ADT, Operations on stack and queue. Implementations using arrays and linked lists. Application of stack for expression conversion and evaluation, Recursion. Problems like maze and knight's tours

Linear Data Structures: Stacks and Queues	**[8 Hrs]**

Concept of Non-Linear Data Structures, Tree: Basic terminology, representation of trees, Binary Tree: ADT Binary trees, Properties of a Binary Tree and its representations, Binary tree traversals – Inorder, preorder, postorder (recursive and non-recursive) and various operations. Binary Search Tree (BST): Definition, searching a BST, Operations of Insertion and deletion on BST. Heaps: Priority queues, Max and Min Heap, Operations on heap

Non-Linear Data Structures: Trees	**[8 Hrs]**

Concept of Non-Linear Data Structures, Tree: Basic terminology, representation of trees, Binary Tree: ADT Binary trees, Properties of a Binary Tree and its representations, Binary tree traversals – In order, preorder, post order (recursive and non-recursive) and various operations. Binary Search Tree (BST): Definition, searching a BST, Operations of Insertion and deletion on BST. Heaps: Priority queues, Max and Min Heap, Operations on heap

Non-Linear Data Structures: Graphs	**[6 Hrs]**

Graph ADT, Representation of graphs using adjacency matrix, adjacency list, Elementary graph operations- Breadth First Search (BFS), Depth First Search (DFS), Analysis of BFS and DFS Spanning Trees- Introduction, obtaining minimum cost spanning tree using greedy design strategy of Prim's and Kruskal's algorithms, Single source shortest paths using Dijkstra's algorithm.

Searching and Sorting:	**[6 Hrs]**

Linear and binary search algorithms and their analysis. Bubble Sort, Selection sort, Insertion Sort, Quick Sort, Heapsort with time complexity analysis. Hashing: hashing functions, chaining,

overflow handling with and without chaining, open addressing: linear and quadratic probing.

Self-study	**[12Hrs]**

Textbooks

1. E. Horowitz, S. Sahni, S. Anderson-freed, “Fundamentals of Data Structures in C”, Second Edition, University Press, ISBN 978-81-7371-605-8
2. B. Kernighan, D. Ritchie, “The C Programming Language”, Prentice Hall of India, Second Edition, ISBN 81-203-0596-5
3. Y. Langsam, M. Augenstein and A. Tannenbaum, “Data Structures using C”, Pearson Education Asia, First Edition, 2002, ISBN 978-81-317-0229-1

Reference Books

1.Ellis Horowitz, S. Sahni, D. Mehta “Fundamentals of Data Structures in C++”, Galgotia Book Source, New Delhi 1995 ISBN 16782928 2.

2.Jean-Paul Tremblay, Paul. G. Sorensan, “An introduction to data structures with Applications”, Tata Mc-Graw Hill International Editions, 2nd edition 1984, ISBN-0-07- 462471-7

Web references

1. https://onlinecourses.nptel.ac.in/e-learning/preview/noc25_cs81

Laboratory assignments

1. Write a program to remove duplicate doubles from an array of doubles. In the program, write a function which accepts an array of doubles and removes the duplicates from the array and has return type void.

2. Compare the time complexity of two sorting algorithms, by following the given steps. Create a set of data files with count of integers varying into thousands and millions. Sort the files using both the algorithms. Plot graph of the time taken by both the programs using tools like gnuplot. Compare the graphs and comment on the time complexity theoretically predicted and practically observed.

3. Write a function which evaluates an infix expression, without converting it to postfix. The input string can have spaces, (,) and precedence of operators should be handled.

4. Implement a queue (that is write queue.c and queue.h only) of characters, such that on an enqueue, the char is added at the end of queue, and on a dequeue the first element is taken out, but the queue uses only a 'head' pointer and not a 'tail pointer.

5. Write a data type called "Integer". The data type should represent integers of unlimited length.

6. Write a sorting program with the following features: Reads data from a text file and sorts it alphabetically by default. If the file has data in rows and columns (separated by space or tab) then allows sorting on a particular column. Allows any sort using numeric or alphabetical ordering.

7. Write the following functions for a binary search tree implementation: Searches the maximum value in the tree, preorder traversal without using recursion, Search the string in the tree and returns a pointer to the node, print the binary tree so that it looks like a tree.

8. Write code to list leaf nodes, non-leaf nodes and level of all nodes in a given binary tree.

9. Write a code for level order traversal of a binary tree with and without stack.

10. Start with an empty AVL tree and perform a series of insertions like : December, January, April, March, July, August, October, February, November, May, June. Display the tree.

This list serves as a guideline and is intended to be continuously refined by the instructor.

(OE) Data Analytics

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs/ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Self-study: 1 Hr/ Week	

Course outcomes

1. Know basics of Data Analytics, people associated with it, roles and responsibilities to be a data Analyst.
2. Learn and apply how to interpret data in Python using multi-dimensional arrays in NumPy.
3. Learn and apply to manipulate Data Frames in Pandas.
4. Apply visualization libraries in Python like Matplotlib and Seaborn to gain insight of the dataset.
5. Explore the latest data analytics tools excel, Plotly, Power BI, Tableau.

Course content

Introduction to Data Analytics:	**[6 Hrs]**

Introduction to Data Analytics, Need of data Analytics, types of data analytics, Challenges, different types of data, data sources, data collection, data integration, data wrangling, role of Data Engineers, Data Analysts, Data Scientists, Business Analysts, and Business Intelligence Analysts, roles, responsibilities and skill sets required to be a Data Analyst.

Introduction to Python Libraries-NumPy	**[6 Hrs]**

Installation, Basic operation: Arithmetic, Matrix Product, Increment, Decrement, Aggregate function, Indexing, Slicing and Iterating an array, Conditions and Boolean Arrays, Shape manipulations, Array manipulation-multidimensional array

Pandas Library	**[6 Hrs]**

Installation of Pandas, Testing pandas installation, Introduction to pandas data structure- the series, data frame, Index objects, other functionalities on Indexes-Reindexing, Dropping, Operations on Data Frame and Series- merging, combining, pivoting, removing, data transformation, Data aggregation, Reading and writing data to CSV file, random sampling, group iteration.

Data Visualization with Matplotlib and Seaborn libraries	**[6 Hrs]**

Pyplot, plotting window, adding elements to chart- text, grid, legend, saving charts, Handling data values, Line chart, Histogram, Bar chart, pie chart, scatter plot, box plot, heat map.

Self-study	**[12 Hrs]**

Tools for Data Analytics: Introduction to data analytics tools, collection of maps, diagrams, charts designed to gather, interpret, and visualize data across diverse applications, how to Choose the right data analysis tool, data visualization with different free open-source tools.

Textbooks

1. Fabio Nelli, “Python Data Analytics with Pandas, Numpy and Matplotlib”, ISBN: 978-1-4842- 3913-1, DOI: 10.1007/978-1-4842-3913-1, Publisher: Apress,Year: 2018
2. Bharti Motwani, “Data Analytics using Python “, Publisher: Wiley, ISBN: 8126502959

Reference Books

1. Vincent Granville, “Developing Analytic Talent -Becoming a Data Scientist”, Publisher: Wiley, 2014, ISBN: 9781118810095.
2. Glenn J. Myatt, Wayne Johnson, “Making Sense of Data I”, Second Edition, Publisher: Wiley, ISBN 978-1-118-40741-7
3. Glenn J. Myatt, Wayne Johnson, “Making Sense of Data II”, Publisher: Wiley, ISBN 978-0- 470-22280-5

(PCC) Machine Learning

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs./ Week	Laboratory: CIE: 100 Marks
Self - study: 1 Hr./week	

Course outcomes

1. Understand modern notions in predictive data analysis
2. Apply machine learning techniques for data preprocessing, model training, evaluation, and performance improvement
3. Apply feature engineering and probabilistic learning techniques for feature selection, extraction, and Bayesian classification.
4. Apply regression and classification algorithms for predictive modeling and decision-making tasks.
5. Analyze neural network, deep learning, and unsupervised learning techniques for pattern discovery and real-world machine learning applications.

Course content

Introduction	**[6 Hrs]**

Introduction to Machine Learning: Types of Human learning, machine learning process, Well-posed learning problem, Types of machine learning and comparison, Applications of machine learning, Tools in Machine Learning.

Preparing to Model	**[6 Hrs]**

Machine learning activities, Types of data in machine learning, dataset understanding, plotting and exploration, checking data quality, remediation, data pre-processing, selecting a model, predictive and descriptive models, supervised learning model training, cross-validation and boot strapping, lazy vs eager learner, interpreting the model- underfitting, overfitting, bias-variance trade-off. Parameter for evaluating performance of classification, regression, and clustering model. Improving performance of a model.

Feature Engineering	**[8 Hrs]**

Feature Engineering: Feature transformation - feature construction, feature extraction by PCA, SVD, LDA. Feature subset selection – feature relevancy and redundancy measures. Feature selection process and approaches. Review of Probability concepts: joint probability, conditional probability, bayes rule, Common discrete and continuous distributions, dealing with multiple random variables, central limit theorem. Bayes classifier, Multi-class Classification, Naïve Bayes classifier, Bayesian belief network.

Supervised Learning	**[8 Hrs]**

Regression: Introduction of regression, Regression algorithms: Simple linear regression, Multiple linear regression, Polynomial regression model, Logistic regression, Maximum likelihood estimation.
Classification: Classification model and learning steps, Classification algorithms: Naïve Bayes classifier, Distance measures, k-Nearest Neighbor (kNN), Decision tree, Support vector machines, Kernel trick, Random Forest.

Unsupervised Learning	**[6 Hrs]**

Unsupervised vs supervised learning, Application of unsupervised learning, Clustering and its types, Partitioning method: k-Means and K-Medoids, Hierarchical clustering, Density-based methods – DBSCAN.

Artificial Neural Network:	**[6 Hrs]**

Biological neuron, Artificial neuron, Activation functions, neural network architecture, perceptron, learning process in ANN, Back propagation. Introduction to deep learning, overview of reinforcement learning, Representation learning, Evolutionary learning. Case-study of ML applications: Image recognition, Email spam filtering, Online fraud detection.

Self-study	**[12Hrs]**

Representation Learning: Supervised Neural Network and multilayer perceptron, Active Learning: Heuristics for Active Learning, Active Learning query strategies, Instance-Based Learning: Radial Basis Function.

Textbooks

1. "Mining of Massive Datasets", Jure Leskovec, Anand Rajaraman, and Jeffery David Ullman, Cambridge University Press, 2 edition (13 November 2014), ISBN-10: 1107077230, ISBN-13: 978-1107077232
2. Machine Learning: Theory and Practice · M N Murty and Ananthanarayana V S · ISBN: 9789393330697 | Year: 2024
3. Tom M Mitchell, —Machine Learning, First Edition, McGraw Hill Education, 2013.

Reference Books

1. Stephen Marsland, —Machine Learning – An Algorithmic Perspective, Second Edition,
2. Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.

Web references

1. https://nptel.ac.in/courses/106106139
2. https://nptel.ac.in/courses/106105152

Laboratory assignments

1. Write a python program to compute Central Tendency Measures: Mean, Median, Mode, Measure of Dispersion: Variance, Standard Deviation.
2. Study of Python Basic Libraries such as Statistics, Math, Numpy and Scipy.
3. Study of Python Libraries for ML application such as Pandas and Matplotlib.
4. Write a Python program to implement Simple Linear Regression.
5. Implementation of Multiple Linear Regression for House Price Prediction using sklearn.
6. Implementation of Decision tree using sklearn and its parameter tuning.
7. Implementation of KNN using sklearn.
8. Implementation of Logistic Regression using sklearn.
9. Implementation of K-Means Clustering.
10. Performance analysis of Classification Algorithms on a specific dataset (Mini Project) Develop a simple neural network model using TensorFlow and Keras for classification tasks.

This list serves as a guideline and is intended to be continuously refined by the instructor.

(PCC) Database Management System

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Laboratory: 2 Hrs./ Week	Laboratory: CIE: 100 Marks
Self - study: 1 Hr./week	

Course outcomes

1. Identify and describe various components of DBMS.
2. Construct Entity-Relationship Model for given applications and demonstrate it as a relational model.
3. Design optimal (SQL) query statement.
4. Apply normalization to database design.
5. Improve efficiency in data retrieval by using indexing and solving analytical problems on serializability, concurrency and recovery.

Course content

Introduction	**[6 Hrs]**

Basic Concepts; Database system application, purpose of database systems, view of data, database languages; Database architecture: components of DBMS and overall structure of DBMS; Various types of databases.

E-R and Relational Model:	**[6 Hrs]**

Database design; E-R model: modeling, entity, attributes, relationships, constraints, components of E-R model; Relational model: basic concepts, attributes and domains, concept of integrity and referential constraints, schema diagram.

Relational Algebra and SQL:	**[8 Hrs]**

Relational algebra: fundamental relational algebra operations, additional relational algebra operations, extended relational algebra operations, null values, modification to database; SQL: basic structure and operations, aggregate functions, nested subqueries, complex queries

Relational Database Design:	**[7 Hrs]**

Basic concept of normalization; Decomposition using functional dependencies (1 to BCNF and 3NF).

Indexing and Hashing:	**[7 Hrs]**

Basic concepts, Ordered Indices; Indices: concepts, B+ trees; Static and dynamic hashing.
Self-Study: B tree index file, Bit Map indices.

Transactions and Concurrency control:	**[5 Hrs]**

Transaction: basic concepts, states, concurrent execution, serializability, recoverability, isolation; Concurrency control: timestamps and locking protocols, validation-based protocols, deadlock handling; Recovery.

Self-study	**[12Hrs]**

Network Model, Views, Basics of Multivalued dependencies, Multiple granularity protocols, ARIES algorithm.

Textbooks

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database system concepts”, Fifth Edition, McGraw Hill International Edition, ISBN 978-0073523323.

2. Raghu Ramkrishnan, Johannes Gehrke, “Database Management Systems”, Second Edition, McGraw Hill International Editions, ISBN 978-0072465631.

Reference Books

1. Rob Coronel, “Database systems: Design implementation and management”, Fourth Edition, Thomson Learning Press, ISBN 978-1418835934.
2. Ramez Elmasri and Shamkant B. Navathe, “Fundamental Database Systems”, Third Edition, Pearson Education, 2003, ISBN 978-0321204486.
3. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, "Database Systems: The Complete Book", Second Edition, Pearson, 2008, ISBN 978-0131873254.

Laboratory assignments

1. Basic SQL Statement: Write the simple SQL Queries on the given schema.
2. Assignment:
   a. SQL: Aggregates Statement: Write the SQL queries using aggregates, grouping and ordering statements for given statements on given schema.
   b. Nested Subqueries and SQL Updates Statement: Write the SQL queries forgiven schema using Nested Subqueries and SQL Updates.

Q.2. SQL DDL and updates Statement: Write the DDL and DML statements for the given statements.

Q.3. Schema creation and constraints Statement: Create the schema and constraints on the given relations using given statements.

Q.4. Implement a small database application using appropriate front End and Back end. Select any real time problem for database implementation. (Draw the ER diagram for the selected problem in hand. Normalize the database up to the appropriate normal form.) This list is a guideline.

This list serves as a guideline and is intended to be continuously refined by the instructor.

(PCC) Automata Theory

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Self - study: 1 Hr./week	

Course outcomes

1. Design appropriate automata for modeling the solution for various computational problems.
2. Apply transformation between multiple representations of automata/machines.
3. Make use of the pumping lemma to show that a language is not regular/context-free.
4. Distinguish different formal computing languages and classify their respective types.
5. Describe the limitations of a computing machine in terms of language recognition.

Course content

Finite Automata:	**[10 Hrs]**

Computability, and Complexity, Strings and Languages: symbol, alphabet, string, formal languages, Formal definition of a finite automaton, Designing finite automata, Nondeterminism: Formal definition, Equivalence of NFAs and DFAs, Minimization of DFA.

Regular Expressions:	**[4 Hrs]**

Regular Expressions: Formal definition of a regular expression, Equivalence with finite automata, Closure properties of regular languages, the pumping lemma for regular languages.

Context-Free Languages:	**[6 Hrs]**

Context-free Grammars: Formal definition, Designing context-free grammars, Parse Trees, Ambiguity, Chomsky Normal Form. Pushdown Automata: Formal definition, Examples.

Turing Machines and Undecidability:	**[6 Hrs]**

Turing Machines Formal definition, Examples, Decidability and Undecidability: Decidable Languages: Decidable problems concerning regular languages, Decidable problems concerning context-free languages, Undecidable problems, The Halting Problem.

Self-study	**[12Hrs]**

Basic Mathematical notations and terminologies for set theory. Closure Properties of context-free languages, the pumping lemma for context-free languages. Variants of Turing Machine: Multi-tape Turing machines, Nondeterministic Turing machines, Enumerators

Textbooks

1. Michael Sipser, “Introduction to the Theory of Computation”, 3rd Edition, 2013, Cengage Learning Publications, ISBN-13: 978-1133187790.
2. John. E. Hopcroft, Rajeev Motwani, J. D. Ullman, “Introduction to Automata Theory, Languages, and Computations”, 3rd Edition, 2009, Pearson Education Publisher, ISBN-10:0321455363 / ISBN-13: 978-0321455369

Reference Books

1. E. V. Krishnamurthy, “Theory of computer science”, 2004, Affiliated East Press Publications, ISBN-10: 038791255X / ISBN-13: 978-0387912554. Dexter C. Kozen, "Automata and Computability", 1997, Springer Verlag Publications, ISBN 0- 387-94907-0.
2. Harry Lewis, Christos H. Papadimitriou, “Elements of the Theory of Computation”, 2 nd Edition, 1997, Prentice-Hall Publications, ISSN 0891-4516.
3. John Martin, “Introduction to Languages and Theory of Computations”, 4th edition, 2010, McGraw-Hill Publications, ISBN 978–0–07–319146–1 / MHID 0–07–319146–9.

(OE) Fundamentals of Operating Systems

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Self - study: 1 Hr./week	

Course outcomes

1. Understand the structure of OS and basic architectural components involved in OS design.
2. Describe the mechanisms of OS to handle processes, threads, and their communication.
3. Analyze the memory, file, and resource management techniques.
4. Illustrate the role of paging, segmentation, and virtual memory in operating systems.

Course content

Introduction to Operating System (OS):	**[10 Hrs]**

Computer System, OS Operations, Virtualization, Distributed Systems, Kernel Data Structures, Computing Environments, OS Services, User and OS Interface, System Calls, System Services, OS Design, OS Structure.

Process Management:	**[4 Hrs]**

Scheduling, Operations, Interprocess Communication, Models of IPC, Thread, Multicore Programming, Multithreading Models, Implicit Threading, CPU scheduling, Scheduling Algorithms, Thread Scheduling, Multi-Processor Scheduling.

Process Synchronization:	**[6 Hrs]**

Critical-section, Peterson's Solution, Hardware Support, Mutex Locks, Semaphores, Classic Problems of Synchronization, Synchronization within the Kernel, Deadlocks: necessary conditions, Prevention, Avoidance, and Detection.

Memory Management:	**[6 Hrs]**

Logical and Physical Memory, Contiguous Memory Allocation, Paging, Swapping, Virtual Memory, Demand Paging, Page Replacement, Allocation of Frames, Thrashing, Secondary Storage: HDD Scheduling, Swap-Space Management

File Management:	**[6 Hrs]**

File Concept, Access Methods, Directory Structure, Protection, Filesystem Structure, Directory Implementation, Allocation Methods, Free-Space Management, Recovery, Filesystem Mounting, Partitions and Mounting, File Sharing

Self-study:	**[12 Hrs]**

Case Studies of Linux OS: Design Principles, Kernel Modules, Process Management, Scheduling, Memory Management, File Systems.

Textbooks

1. "Operating System Concept" by Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, 10th Edition, 2021, Wiley, ISBN-978-1-119-32091-3.
2. "Modern Operating System", by Andrew S. Tanenbaum, 5th Edition, 2022, Pearson Education, ISBN 13: 978-0137618873.

(OE) Fundamentals of Algorithms

Teaching Scheme	Evaluation Scheme

Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks
Self - study: 1 Hr./week	

Course outcomes

1. Recall and comprehend time and space complexity of algorithms and evaluate algorithmic efficiency in terms of runtime and memory usage using asymptotic analysis.
2. Describe the performance of recursive algorithms and analyze the correctness and efficiency of their solutions.
3. Evaluate and use appropriate algorithmic techniques, and design efficient solutions to various computational challenges.
4. Apply design technique that would be most suitable to solve a given problem, justify the technique used and analyze the resultant algorithm.

Course content

Basics of Algorithm:	**[6 Hrs]**

Introduction to algorithm, Classification of algorithms- Deterministic, nondeterministic, performance analysis of an algorithm, time and space complexity, asymptotic notations (O, Ω, θ notations), complexity analysis with examples.

Design Techniques-I:	**[6 Hrs]**

Greedy Methods: Introduction, Knapsack problem, Job sequencing with deadlines, Dijikstra's Single source shortest paths, Minimum cost spanning tree: Prim's algorithm, Kruskal's algorithm, Optimal merge patterns.

Design Techniques-II:	**[6 Hrs]**

Divide and Conquer: Introduction, Mergesort, Binary Search, Quicksort, Multiplication of two n-bit numbers.

Design Techniques-III:	**[5 Hrs]**

Dynamic Programming: Introduction, all pairs shortest paths, Traveling salesperson problem, longest common subsequence, Bellman-Ford algorithm.

Graph Algorithms:	**[4 Hrs]**

Depth-first search (DFS), breadth-first search (BFS), Topological sort.

Selected Algorithms	**[4 Hrs]**

String Matching: The naïve string-matching algorithm, The Robin-Karp algorithm, The Knuth-Morris-Pratt algorithm.

Self-study:	**[8 Hrs]**

Algorithmic Problem-Solving Platforms: Activity: Solve algorithmic problems on platforms like Leet Code or Hacker Rank.

Textbooks

1. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Universities Press, 2nd edition (2008) , ISBN-13: 978-8173716126 Thomas Cormen, Charles Leiserson, Ronald Rivest and Cliford Stein, "Introduction to Algorithms", PHI, 3rd edition, ISBN-13: 978-8120340077

Reference Books

1. Gilles Brassard and Paul Bratley, "Fundamentals of Algorithmics", PHI, ISBN-13: 978 8120311312

2. Jon Kleinberg and Éva Tardos, “Algorithm Design”, Pearson Education India, ISBN-13: 978- 9332518643

(MDM-01) The mathematics of quantum mechanics

Teaching Scheme	Evaluation Scheme
Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Self - study: 1 Hr./week

Course outcomes

1. Analyze simple states of superposition and the effect of doing the measurement in different basis states.
2. Design and analyze quantum circuits with single and two-qubit gates.
3. Equip students with the linear algebra background required for this course.
4. Analyze quantum circuits with superposition and entanglement.
5. Analyze quantum algorithms and complexity.
6. Understand QAOA and error correction scheme.

Course content

Introduction and Mathematical postulates of quantum mechanics: **[8 Hrs]**

Concept of quantum mechanics, Quantum Technologies, need, Complex numbers, Euler's Formula and Polar form, vector Spaces. Linear Combination of Vectors, Uniqueness of a spanning set, basis & dimensions, inner Products, orthonormality, bra-Ket formalism, Quantum bits, Bloch sphere representation of a qubit, multiple qubits .

Matrices and Operations: **[8 Hrs]**

Observables, The Pauli Operators, Outer Products, The Closure Relation, Representation of operators using matrices, outer products & matrix representation, matrix representation of operators in two dimensional spaces, Pauli Matrix, Hermitian unitary and normal operator, Eigen values & Eigen Vectors, Spectral Decomposition, Trace of an operator, important properties of Trace, Expectation Value of Operator, Projection Operator, Positive Operators, Commutator Algebra, Heisenberg uncertainty principle, polar decomposition & singular values, Postulates of Quantum Mechanics.

Tensor Products and Density Operator: **[6 Hrs]**

Representing Composite States in Quantum Mechanics, Computing inner products, Tensor products of column vectors, operators and tensor products of Matrices. Density Operator of Pure & Mix state, Key Properties, Characterizing Mixed State, Practical Trace & Reduce Density Operator, Density Operator & Bloch Vector.

Quantum Measurement Theory, Noise and error correction: **[8 Hrs]**

Distinguishing Quantum states & Measures, Projective Measurements, Measurement on Composite systems, Generalized Measurements, Positive Operator-Valued Measures, Graph states and codes, Quantum error correction.

Quantum Algorithms **[7 Hrs]**

Classical computations on quantum computers, Relationship between quantum and classical complexity classes, Deutsch's algorithm, Deutsch's-Jozsa algorithm, Shor factorization, Grover search, Quantum teleportation.

Self-study: **[8 Hrs]**

Explore advanced quantum algorithms like Grover's, Shor's, Deutsch-Jozsa, and Simon's, and practice analyzing quantum circuits with entanglement and superposition. Study Quantum Approximate Optimization Algorithm (QAOA), quantum error correction codes (Shor, Steane), and experiment with single- and multi-qubit gate operations.

Textbooks

1. P. Wittek, Quantum Computing. Education, 2007

2. Phillip Kaye, Raymond Laflamme et. al., An introduction to Quantum Computing, Oxford University press, 2007.
3. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press, Cambridge, 2020

Reference Books

1. “Quantum Computing: A Gentle Introduction” by Eleanor Rieffel and Wolfgang Polak, MIT press (2014)
2. Quantum Computation and Quantum Information, M. A. Nielsen &I. Chuang, Cambridge University Press (2013)

(VSEC) Development Tools Laboratory

Teaching Scheme	**Evaluation Scheme**
Laboratory: 2 Hrs/ Week	CIE: 100 Marks
Course outcomes	

1. Develop an application in a group using GIT, demonstrating ability to work remotely, push, and pull.
2. Write a report in a specified format using LaTEX.
3. Demonstrate programming ability using Unix Shell

Course content

LaTEX:

Basic syntax, compiling and creating documents; Document structure, sections, paragraphs; packages, Math, Adding Images, Drawing images (using tools like Inkscape) Table of contents; Source code, graphs (using tools like Graphviz), Adding references, different templates, IEEE format, Bibliography.

Shell Programming:

Introduction to Linux commands, concept of shell, shell variables, getcwd() and pwd; Introduction to shell programming features: Variables declaration & scope, test, return value of a program, if-else and useful examples, for and while loop, switch case; Shell functions, pipe and redirection, wildcards, escape characters; Awk script: Environment and workflow, syntax, variables, operators, regular expressions, arrays, control flows, loops, functions, output redirections.

GIT:

Creating a project using git locally, add, commit, status, diff; branch and merge, GIT: cloning
a remote repo, working with a remote repo – git push, pull, fetch; creating issues and pull requests;
working on a project in a distributed fashion.

Suggested List of Assignments:

The lab incharge will impart instructions during a part of the lab session and assign the assignments based on it for an appropriate duration of days.

1. Format a given essay using sections, paragraphs, headings in LaTEX.
2. Format a given report in IEEE format using LaTEX.
3. Write a shell program which reads a set of unspecified count of numbers and prints their sum and average.
4. Write a shell program to extract a compressed file in any format (zip, tar.gz, tar.gz2, .tar, .bz2, .gz, .rar, .Z, .7z, etc)
5. Write a shell program to convert a CSV file of contacts, into a VCF file.
6. Write a shell program to sort all files stored in a given folder hierarchy, on their size.
7. Write a shell script to manage a todo list from the command line. The script should be able to add, remove, list, sort, prepend, append, deduplicate todo-items
8. Write a program that scans a file line by line, splits each input line into fields later, compares input line/fields to pattern and performs action(s) on matched lines
9. Develop a program using git locally. E.g. add the exponent operator to the calculator program that you wrote. Demonstrate the ability to do git add, commit, status, diff.
10. Create a branch in your calculator program to do hexadecimal calculation. Write code and develop two branches. Merge the two to have a decimal/hex calculator. Demonstrate git branch, merge capability.
11. In a group of 3, create a github/gitlab repo. Raise issues, send pull requests, do the local and remote merges and finally get a synced local repo. For example, in the calculator project one student will become the developer, the other two will create issues and send pull requests for features like adding an operator, developing and pulling those requests. Rotate the roles and repeat.

References:

LaTeX

1. Leslie Lamport, “LaTeX: A document preparation system”, User’s guide and reference manual,
   2nd Edition, 1994, by Addison-Wesley Professional. ISBN 0201529831, 9780201529838
2. Stefan Kottwitz, “LaTeX Beginner's Guide: Create High-quality and Professional-looking Texts, Articles, and Books for Business and Science Using LaTeX, Packt Publishing, 2011. ISBN: 1847199860, 9781847199867” https: //www.latex-project.org/
3. Introduction to LaTEX, MIT
   http: //web.mit.edu/rsi/www/pdfs/new-latex.pdf
4. A simple guide to LaTeX - Step by Step
   https: //www.latex-tutorial.com/tutorials/

Shell

1. Bash Guide for Beginners: https://www.tldp.org/LDP/Bash-BeginnersGuide/Bash-Beginners-Guide.pdf
2. Bash Reference Manual https://www.gnu.org/software/bash/manual/bash.pdf
3. Tutorials on Shell Programming
   https://www.shellscript.sh/
   https://www.tutorialspoint.com/unix/shell_scripting.html

(HSMC) Design Thinking

Teaching Scheme	Evaluation Scheme
Lectures: 1 Hr/ Week	CIE: 100 Marks

Self - study: 1 Hr./week

Course outcomes

1. Outline various learning styles and psychological principles and Infer Design Thinking principles & methodology.
2. Explain the levels of designs and Experiment with the process using human centric tools.
3. Propose real-time innovative engineering product designs and choose appropriate frameworks, strategies, techniques for prototype development.
4. Appraise user feedback and propose corrective innovative solutions to meet project requirements using critical thinking skills.

Course content

An Insight to Learning:	**[3 Hrs]**

Experiential Learning Styles, Self-assessment Psychological Principles in Design Thinking Perception & Observation, Imagination & Creative Confidence (lateral thinking & 6 thinking hats).

Design Thinking Framework:	**[2 Hrs]**

Introduction to different frameworks of DT, Stanford d. school framework Empathize, Define, Ideate, Prototype, Test Case Study: IDEO Shopping Cart, etc.

User-Design Relationship:	**[2 Hrs]**

Levels of Designs Understanding users through interviews, personas, empathy maps/affinity diagrams/journey maps and need identification.

Introduction to Human Centric Tools in Design Thinking Process:	**[2 Hrs]**

Brainstorming & Mind mapping, POV and HMW

Process of Product Design	**[2 Hrs]**

Process of simple Product Design using real life problem statements from our daily routine activities, Design Thinking Approach, Stages of Product Design, Examples of best product designs and functions. Hands on Lab Assignment –Simple routinely used Product Design. Hands-on demonstration of how to translate the ideas into physical objects. Better visualization of the ideas and concepts using, IDEA LAB/FAB LAB facilities such as Wood router, Laser cutting of thin plastic sheets, clay modelling, Expandable Polystyrene etc

Prototyping	**[2 Hrs]**

What is Prototype? Understanding necessity of making prototypes by building the prototypes for preselected Engineering problem, using one or combinations of the digital fabrication techniques & electronics fabrication systems.

Self-Study	**[12 Hrs]**

Testing, Sample Example, Test Group Marketing Feedback, Re-Design & Re-Create Final Presentation – “Solving Practical Engineering Problem through Innovative Product Design & Creative Solution”.

Textbooks

1. Norman, D. (2013). The Design of Everyday Things. Basic Books, NY.
2. Norman, D. (2004). Emotional Design. Basic Books, NY.
3. Brown, T. (2019). Change by Design. HarperCollins Publishers, NY.
4. Lal, D. M. (2021). Design Thinking- Beyond the Sticky Notes. Sage Publications India Pvt. Ltd.

Reference Books

1. Malik, A. D. M. (2019). Design Thinking for Educators. Notion Press, Chennai, India.
2. E. F. Crawley, "Creating the CDIO Syllabus, a universal template for engineering education," 32nd Annual Frontiers in Education, Boston, MA, USA, 2002, pp. F3F-F3F, doi: 10.1109/FIE.2002.1158202.
3. Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., & Leifer, L. J. (2005). Engineering design thinking, teaching, and learning. Journal of engineering education, 94(1), 103-120.
4. Panke, S. (2019). Design thinking in education: Perspectives, opportunities and challenges. Open Education Studies, 1(1), 281-306.
5. Parmar, A. J. (2014, October). Bridging gaps in engineering education: Design thinking a critical factor for project-based learning. In 2014 IEEE frontiers in education conference (FIE) proceedings (pp. 1-8). IEEE.

(PCC) Discrete Structures

Teaching Scheme	**Evaluation Scheme**
Lectures: 3 Hrs./ Week	Theory: MSE: 30 Marks, TA: 20 Marks, ESE: 50 Marks

Self - study: 1 Hr./week

Course outcomes

1. State real life situations using predicate/propositional calculus and do logical arguments leading to conclusions.
2. Solve problems involving sets, functions, sequences and summation
3. Apply principles of counting and induction to solve real life problems.
4. Solve problems involving algebraic systems.
5. Apply principles of relations to solving real life problems

Course content

The Foundation Logic and Proofs: **[6 Hrs]**

Propositions, Conditional Propositions, Logical Connectivity, Propositional equivalences, predicates and Quantifiers, First order logic, Proofs: Proof Techniques, Rule of inference.

Sets, Functions Sequences and Sums: **[6 Hrs]**

Set, set operation, Functions, Types of Functions, Functions: Definition, Domain, Range, Image, etc. Types of functions: Surjection, Injection, Bisection, Inverse, Identity, Composition of Functions.

Counting: **[6 Hrs]**

Basic Counting Techniques (sum, product, subtraction, division, exponent), Pigeonhole Principle, Permutations and Combinations and numerical problems, Binomial Coefficients, Pascal's Identity and Triangle.

Relations: **[6 Hrs]**

Definitions, Properties of Binary Relations, Representing relation, closures of relations, Equivalence Relations and partitions, Partial ordering relations.

Self-study: **[12 Hrs]**

Algebraic Systems, Groups, Semi Groups, Monoids, Subgroups, Permutation Groups, Codes and Group codes, Isomorphism and Automorphisms, Homomorphism and Normal Subgroups, Ring, Field.

Textbooks

1. “Discrete Mathematics and Its Applications”, Kenneth H. Rosen, 7th Edition, Tata McGraw-Hill, 2017, ISBN: 9780073383095.
2. “Elements of Discrete Mathematics”, C. L. Liu, 4th Edition, Tata McGraw-Hill, 2017, ISBN-10: 1259006395, ISBN-13: 9781259006395.

Reference Books

1. “Discrete Mathematical Structures”, G. Shanker Rao, 2nd Edition, 2009, New Age International, ISBN-10: 8122426697, ISBN-13: 9788122426694.
2. “Discrete Mathematics”, Lipschutz, Lipson, 2nd Edition, 1999, Tata McGraw-Hill, ISBN: 007463710X.
3. “Graph Theory”, K. Balakrishnan, 1st Edition, 2004, Tata McGraw-Hill, ISBN-10: 0-07-058718-3, ISBN-13: 9780070587182.
4. “Discrete Mathematical Structures”, B. Kolman, R. Busby and S. Ross, 4th Edition, Pearson Education, 2002, ISBN: 8178085569.
5. “Discrete Mathematical Structures with Application to Computer Science”, J. Tremblay, R. Manohar, Tata McGraw-Hill, 2002, ISBN: 0070651426.