Mandsaur University

Bachelor of Technology (Computer Science and Engineering)

Semester-III

L-2T-1P-0C-3

CSE 726 TR1: Data Science using Python

Course Objectives

- To gain hands-on experience and practice using Python to solve real data science challenges
- To familiarize the students to practice Python programming and coding for modeling, statistics, and Storytelling
- To utilize popular libraries such as Pandas, numPy, matplotlib, and SKLearn.
- To enable the students to get hands-on experience creating analytics models and apply those models to real-world problems

Course Outcomes (COs): Upon completion of this unit students will be able to:

1. Understand the need for data science and solve basic problems using Python built-indata types and their method
2. Understand an application with user-defined modules and packages using OOPconcept
3. Apply the data transformation and data manipulation operations using "pandas"
4. Analyze nature of data with help of different tools and visualization
5. Apply statistical analysis techniques for solving practical problems

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	3	-	-	1	-	2	-	-	-	-	-	3	3	-
CO2	2	1	-	3	-	1	2	-	-	-	1	-	2	-	-
CO3	1	-	3	1	-	-	-	-	-	-	1	-	2	2	-
CO4	3	-	-	2	-	1	-	-	-	-	-	-	-	2	-
CO5	2	2	-	-	2	-	-	1	-	-	1	-	2	1	-

High-3 Medium-2 Low-1

UNIT I: Overview of Python and Data Science **7 Hours**

Introduction to Data Science - Why Python? - Essential Python libraries - Python Introduction- Features, Identifiers, Reserved words, Indentation, Comments, Built-in Data types and their Methods: Strings, List, Tuples, Dictionary, Set - Type Conversion- Operators. Decision Making- Looping- Loop Control statement- Math and Random number functions. User defined functions - function

UNIT II: File, Exception Handling and OOP **11 Hours**

User defined Modules and Packages in Python- Files: File manipulations, File and Directory related methods - Python Exception Handling. OOPs Concepts -Class and Objects, Constructors – Data hiding- Data Abstraction- Inheritance

UNIT III: NumPy for Simulation Model **11 Hours**

Introduction to NumPy - Basics of NumPy Arrays, Computation on NumPy Arrays- indexing, slicing, reshaping. Universal Functions, Aggregations.Computation on Arrays – broadcasting, comparisons, Fancy indexing, Sorting Arrays, Structured Arrays

UNIT IV: Data wrangling, Reshaping and Summarizing with panda **7 Hours**

Introducing Pandas Objects – series, data frames, index, Processing CSV, JSON, XLS data, Operations on Pandas Objects – indexing and selection, universal functions, missing data, hierarchical indexing, Combining Dataset – concat and append, merge and join. Aggregation and grouping, Pivot tables, Vectorized string operations, Working with time series, High performance Pandas – eval()

UNIT V: Data Visualization using Matplot **9 Hours**

General MatplotLib, Simple Line Plots, Simple Scatter Plots, Density and Contour Plots, Histograms, Binnin, and Density, Customizing Plot Legends, Customizing Colorbars, Text and Annotation, Three-Dimensional Plotting in Matplotlib, Geographic Data with Basemap, Visualization with Seaborn

Total: 45 Hours

Text Books:

1. Shrawan Kumar Sharma, Bhoomi Devnani, Mastring Data Science , Notion Press, ed.1 2024
2. James, G., Witten, D., Hastie, T., Tibshirani, R. An introduction to statistical learning with applications in R. Springer, 2013.
3. Han, J., Kamber, M., Pei, J. Data mining concepts and techniques. Morgan Kaufmann, 2011.
4. Hastie, T., Tibshirani, R., Friedman, J. The Elements of Statistical Learning, 2nd edition. — Springer, 2009. 4. Murphy, K. Machine Learning: A Probabilistic Perspective. - MIT Press, 201

Reference Books:

1. Practical Data Science with R". Nina Zumel, John Mount. Manning, 2014
2. "Data Science for business", F. Provost, T Fawcett, 2013

Web links and Video Lectures (e-Resources):

https://www.youtube.com/watch?v=ua-CiDNNj30
https://nptel.ac.in/courses/106106179

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Mandsaur University

Bachelor of Technology (Computer Science and Engineering)

Semester-III

L-0 T-0 P-2 C-1

CSE 726 TR1: Data Science using Python

Course Objectives

- An understanding of the data operations
- An overview of simple statistical models and the basics of machine learning techniques of regression.
- An understanding good practices of data science
- Skills in the use of tools such as python, IDE
- Understanding of the basics of the Supervised learning

Course Outcomes (COs): Upon completion of this unit students will be able to:

6. Describe what Data Science is and the skill sets needed to be a data scientist
7. Explain the significance of exploratory data analysis (EDA) in data science
8. Ability to learn the supervised learning, SVM
9. Apply basic machine learning algorithms (Linear Regression)
10. Explore the Networks, PageRank

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	3	-	-	1	-	2	-	-	-	-	-	3	3	-
CO2	2	1	-	3	-	1	2	-	-	-	1	-	2	-	-
CO3	1	-	3	1	-	-	-	-	-	-	1	-	2	2	-
CO4	3	-	-	2	-	1	-	-	-	-	-	-	-	2	-
CO5	2	2	-	-	2	-	-	1	-	-	1	-	2	1	-

High-3 Medium-2 Low-1

Practical's

1. Perform Creation, indexing, slicing, concatenation and repetition operations on Python builtin data types: Strings, List, Tuples, Dictionary, Set
2. Apply Python built-in data types: Strings, List, Tuples, Dictionary, Set and their methods to solve any given problem.
3. Handle numerical operations using math and random number functions.
4. Create user-defined functions with different types of function arguments.
5. Perform File manipulations- open, close, read, write, append and copy from one file toanother.
6. Write a program to implement OOP concepts like Data hiding and Data Abstraction.
7. Create NumPy arrays from Python Data Structures, Intrinsic NumPy objects and RandomFunctions.
8. Manipulation of NumPy arrays- Indexing, Slicing, Reshaping, Joining and Splitting.

9. Computation on NumPy arrays using Universal Functions and Mathematical methods.
10. Load an image file and do crop and flip operation using NumPy Indexing.
11. Create Pandas Series and Data Frame from various inputs.
12. Import any CSV file to Pandas Data Frame and perform the following:
    (a) Visualize the first and last 10 records
    (b) Get the shape, index and column details
    (c) Select/Delete the records (rows)/columns based on conditions.
    (d) Perform ranking and sorting operations.
    (e) Do required statistical operations on the given columns.
    (f) Find the count and uniqueness of the given categorical values.
    (g) Rename single/multiple columns
13. Import any CSV file to Pandas Data Frame and perform the following:
    (a) Handle missing data by detecting and dropping/ filling missing values.
    (b) Transform data using apply () and map() method.
    (c) Detect and filter outliers.
    (d) Perform Vectorized String operations on Pandas Series.
14. Visualize data using Line Plots, Bar Plots, Histograms, Density Plots and Scatter Plots using Matplotlib.

Total: 30 Hours

Text Books:

5. Shrawan Kumar Sharma, Bhoomi Devnani, Mastring Data Science , Notion Press, ed.1 2024
6. James, G., Witten, D., Hastie, T., Tibshirani, R. An introduction to statistical learning with applications in R. Springer, 2013.
7. Han, J., Kamber, M., Pei, J. Data mining concepts and techniques. Morgan Kaufmann, 2011.
8. Hastie, T., Tibshirani, R., Friedman, J. The Elements of Statistical Learning, 2nd edition. — Springer, 2009. 4. Murphy, K. Machine Learning: A Probabilistic Perspective. - MIT Press, 201

Reference Books:

3. Practical Data Science with R". Nina Zumel, John Mount. Manning, 2014
4. "Data Science for business", F. Provost, T Fawcett, 2013

Web links and Video Lectures (e-Resources):

https://www.youtube.com/watch?v=ua-CiDNNj30
https://nptel.ac.in/courses/106106179

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Mandsaur University

Bachelor of Technology (Computer Science and Engineering)

Semester-III

L-2T-1P-0C-3

CSE882 TR1: Blockchain and Distributed Ledger Technology

Course Objectives

- To Classify Block chain and Distributed Ledger Technologies
- To learn the development in Blockchain functionalities
- To identify alternative techniques to proof of work for Blockchain protocols, proof of Stake/space

Course Outcomes (COs): Upon completion of this unit students will be able to:

11. Understand Comprehend the functionality of block chain
12. Analyze Choose a block chain implementation based on real time scenario
13. Apply the techniques for anonymity preservation
14. Understand the Block chain challenges
15. Understand the use cases of distributed ledger technology

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	-	2	3	-	-	1	-	2	-	-	-	-	-	3
CO2	3	1	2	1	-	-	-	1	2	-	-	-	1	-	2
CO3	-	-	1	-	3	1	-	-	-	-	-	-	1	-	2
CO4	-	1	3	-	-	2	-	1	-	-	-	-	-	-	-
CO5	2	-	2	2	-	-	2	-	-	1	-	-	1	-	2

High-3 Medium-2 Low-1

UNIT I: Blockchain and Distributed Ledger Fundamentals **7Hours**

Blockchain - Distributed Ledger - Cryptographic basics for cryptocurrency - signature schemes, encryption schemes and elliptic curve cryptography - CAP theorem - Categories of Blockchain: Public blockchain, Private Blockchain, Permissioned Ledger, Tokenized blockchain, Tokenless blockchain, and Sidechains.

UNIT II: Blockchain Functionality **11Hours**

Distributed identity: Public and private keys, Digital identification and wallets - Decentralized network - Permissioned distributed Ledger - Blockchain data structure - Double spending - Network consensus - Sybil attacks - Block rewards and miners - Forks and consensus chain - Finality in Blockchain Consensus - Limitation of proof-of-work - Alternatives to Proof of Work

.

UNIT III: Blockchain Implementation **11 Hours**

Bitcoin and Merkle Root - Eventual Consistency and Bitcoin - Byzantine Fault Tolerance -Bitcoin and Secure Hashing - Bitcoin block-size - Bitcoin Mining - Blockchain Collaborative Implementations: Hyperledger, Corda - Ethereum's ERC 20 and token explosion

UNIT IV: Distributed Ledger Technology in Alternative Blockchain **7 Hours**

Kadena, Ripple, Stellar, Rootstock, Drivechain, Quorum – Decentralized Network manager: Tezos, Maidsafe, BigChainDB - Decentralized Cloud Storage: Storj.

UNIT V: Blockchain Challenges **9 Hours**

Blockchain Governance Challenges: Bitcoin Blocksize Debate, The Ethereum DAO Fork, Ethereum's Move to PoS and Scaling Challenges - Blockchain Technical Challenges: Denial-of-Service Attacks, Security in Smart Contracts, Scaling, Sharding

Total: 45 Hours

Text Books:

1. Fundamental Concept of Blockchain Technology "For Engineers", by Shrawan kumar Sharma , Shiv Kumar , Notion Press Publications
2. Iyer, Kedar, et al. Blockchain: A Practical Guide to Developing Business, Law, and Technology Solutions., 2018, 1st edition, McGraw-Hill Education, United Kingdom
3. Wattenhofer, R. Distributed Ledger Technology: The Science of the Blockchain, 2017, 1st edition, CreateSpaceger Technology: Independent Publishing Platform, United States.

List of e-Learning Resources:

1. Linkhttps://onlinecourses.nptel.ac.in/noc22_cs44/preview

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Mandsaur University

Bachelor of Technology (Computer Science and Engineering)

Semester-III

L-0 T-0 P-2 C-1

CSE882 TR1: Blockchain and Distributed Ledger Technology

Course Objectives

- To Classify Block chain and Distributed Ledger Technologies
- To learn the development in Blockchain functionalities
- To identify alternative techniques to proof of work for Blockchain protocols, proof of Stake/space

Course Outcomes (COs): Upon completion of this unit students will be able to:

16. Understand Comprehend the functionality of block chain
17. Analyze Choose a block chain implementation based on real time scenario
18. Apply the techniques for anonymity preservation
19. Understand the Block chain challenges
20. Understand the use cases of distributed ledger technology

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	-	2	3	-	-	1	-	2	-	-	-	-	-	3
CO2	3	1	2	1	-	-	-	1	2	-	-	-	1	-	2
CO3	-	-	1	-	3	1	-	-	-	-	-	-	1	-	2
CO4	-	1	3	-	-	2	-	1	-	-	-	-	-	-	-
CO5	2	-	2	2	-	-	2	-	-	1	-	-	1	-	2

High-3 Medium-2 Low-1

Practical's

15. Deploy and Implement consensus mechanism (e.g., Proof of Work, Proof of Stake, or Practical Byzantine Fault Tolerance)
16. Implement the mining module of Bitcoin client. The mining module, or miner, should produce blocks that solve proof-of-work puzzle.
17. Compile and test smart contracts on a testing framework using the Ethereum Virtual Machine (EVM).
18. Deploy a chain code using Hyperledger Fabric on a custom network.
19. Create how the Merkle Root is used to verify the authenticity of data in a block..
20. **Design a blockchain-based supply chain solution to track and verify product authenticity**.
21. Deploy and Implement consensus mechanism (e.g., Proof of Work, Proof of Stake, or Practical Byzantine Fault Tolerance)
22. Implement the mining module of Bitcoin client. The mining module, or miner, should produce blocks that solve proof-of-work puzzle.
23. Compile and test smart contracts on a testing framework using the Ethereum Virtual Machine (EVM).

24. Create how the Merkle Root is used to verify the authenticity of data in a block..
25. **Design a blockchain-based supply chain solution to track and verify product authenticity**.

Total: 30 Hours

Text Books:

1. Fundamental Concept of Blockchain Technology “For Engineers”, by Shrawan kumar Sharma , Shiv Kumar , Notion Press Publications
2. Iyer, Kedar, et al. Blockchain: A Practical Guide to Developing Business, Law, and Technology Solutions., 2018, 1st edition, McGraw-Hill Education, United Kingdom
3. Wattenhofer, R. Distributed Ledger Technology: The Science of the Blockchain, 2017, 1st edition, CreateSpaceger Technology: Independent Publishing Platform, United States.

List of e-Learning Resources:

1. Linkhttps://onlinecourses.nptel.ac.in/noc22_cs44/preview

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Mandsaur University

Bachelor of Technology (Computer Science and Engineering)

Semester-IV

L-3 T-1 P-0 C-4

CSE160 TR1: Theory of Computation

Course Objectives

- To give an overview of the theoretical foundations of computer science from the perspective of formal languages.
- To illustrate finite state machines to solve problems in computing.
- To explain the hierarchy of problems arising in the computer sciences.
- To familiarize Regular grammars, context frees grammar.
- To develop an understanding of computational complexity

Course Outcomes (COs): Upon completion of this unit students will be able to:

1. Apply Finite Automata for different Regular Expressions and Languages and basic concepts of formal languages and mathematical preliminaries.
2. Understand the basic aspects of context free grammar and languages with various normal form techniques.
3. Apply the push down automata to solve and represent context free language.
4. Apply the Turing machine to solve and represent various formal language.
5. Understand the basic concept and applications of various problem classes such as P, NP etc.

Articulation Matrix

Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	2	3	1	-	-	-	-	-	-	-	-	-	-	-	2
CO2	3	2	1	-	-	-	-	-	-	-	-	-	-	-	1
CO3	2	2	3	1	-	1	1	-	-	-	-	-	1	-	-
CO4	2	2	3	2	1	1	1	-	-	-	-	1	2	-	1
CO5	3	2	-	1	1	-	-	1	-	-	-	1	2	-	1

High-3 Medium-2 Low-1

UNIT I: Introduction to Theory of Computation and Finite Automata **9 Hours**

Mathematical preliminaries: Alphabet, Strings, Languages, States, Transitions, Grammar, Languages and Automata, Chomsky Hierarchy, Designing Deterministic and Nondeterministic Finite Automata.

Regular Expressions and Languages: Recursive definition of regular expression, regular set, identities of regular expressions, examples and FA. Equivalence of DFA and NDFA, Mealy & Moore machines, Minimization of finite automata, two-way finite automata.

UNIT II: Context Free Grammars (CFG) and Languages **9 Hours**

Finite automata and regular expression, pumping lemma and regular sets, Application of pumping lemma, Myhill-Nerode Theorem, Linear Grammar, right linear Grammar and Left Linear grammar, Regular

Grammar. **Context Free Grammar**- Definition, Derivation, sentential form, parse tree, Derivation, Parse tree, Ambiguity in Grammar and Language, Simplification of CFG-Elimination of Useless Symbol, Null Production and Unit Production, Normal Forms- Chomsky Normal form, Greibach normal form.

UNIT III: Push Down Automata (PDA) **9 Hours**

Definition, The Language of PDA, Equivalence of PDA 's and CFG- CFG to PDA, PDA to CFG.

Deterministic Push Down Automata (DPDA) – Regular language and DPDA, DPDA and CFL, DPDA and ambiguous grammar, Non-deterministic Push Down Automata (NPDA). The pumping lemma for CFL's, Closure properties of CFL's, Decision problems involving CFL's.

UNIT IV: Turing Machines **9 Hours**

Turing machines (TMs): TM Model and conventions, Formal Definition, TM Instantaneous Description (ID), Transition Function, Languages of TM, Types of TM: Deterministic Turing Machines (DTM) and Non-deterministic Turing Machines (NTM), Extension to Basic TM: TM with Multiple tracks, Multi tape TMs, Universal TM (UTM), Church-Turing hypothesis , Post Machines: Introduction to Post Machines (PMs), Comparison between FA, PDA and TM Undecidable problems about Turing Machines, Properties of recursive & recursively enumerable languages, Universal Turing Machine.

UNIT V: Tractable & Intractable **9 Hours**

Problems Classes: P, NP, NP complete and NP Hard, Examples of these Problems like Satisfiability problems, Vertex Cover Problem, Hamiltonian Path Problem, Traveling sales man problem, Partition problem etc.

Total: 45 Hours

References

1. "Theory of Computer Science", by K.L.P Mishra & N. Chandrasekaran, PHI Learning, 2006
2. "Introduction to Languages and Theory of Computation", by John Martin, McGraw- Hill, 2003.
3. "Introduction to Automata theory, Languages & computation", by John E. Hopcroft, Jeffery Ullman, Narosa Publishers, 2008.
4. "Introduction to the Theory of Computation", by Michael Sipser, Books/Cole Thomson Learning, 2001.
5. "Introduction to Computer Theory", by Daniel I.A. Cohen, Wiley India, 1986.

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Bachelor of Technology (Computer Science and Engineering)
Semester-IV

L-2 T-1 P-0 C-3

CSE170 TR1: Analysis and Design of Algorithms

Course Objectives

- To give comprehend fundamental algorithmic concepts, complexities, and implement various algorithmic strategies effectively.
- To deliver diverse algorithmic techniques such as Greedy Algorithms, Dynamic Programming, Backtracking, and Branch and Bound for problem-solving across different domains.
- To give skills to optimize solutions for a range of problems including optimization, shortest path, combinatorial, and complexity Organize using advanced algorithms.
- To give understanding of complexity theory, NP Completeness, and NP hard problems while discussing their implications and solutions.

Course Outcomes (COs): Upon completion of this course students will be able to:

1. Understand fundamental algorithms, complexities, proficiently implement sorting and matrix multiplication techniques.
2. Apply Greedy Algorithms for optimization, construct Minimum Cost Spanning Trees, solve Shortest Path problems effectively.
3. Apply Dynamic Programming for complex problem-solving, solve Shortest Path problems efficiently.
4. Apply Backtracking and Branch and Bound techniques for combinatorial problem-solving.
5. Understand advanced algorithms, apply Approximation Algorithms, comprehend NP Completeness and NP hard problems.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	3	-	1	1	-	-	1	-	-	1	1	-	2	1
CO2	3	2	2	-	-	-	-	-	-	-	-	-	3	1	1
CO3	3	2	1	-	-	-	-	1	-	-	1	1	3	1	1
CO4	3	-	1	2	-	-	1	-	-	-	-	1	2	3	1
CO5	1	-	1	1	3	-	1	-	-	-	1	-	2	3	1

High-3 Medium-2 Low-1

UNIT I: Introduction **9 Hours**

Algorithm, performance evaluation of algorithms, space & time complexity, asymptotic notations, Master's Theorem. Divide and Conquer: General Concept, Finding the maximum and minimum, Quick Sort, Merge Sort, Max and Min Heap, Heap Sort, Binary Search, Strassen's matrix multiplication.

UNIT II: Greedy Algorithm **9 Hours**

General Concept, Knapsack Problem (Fractional Knapsack), Job Sequencing with Deadline, Huffman's Codes, Minimum Cost Spanning Tree Kruskal's Algorithm, Prim's Algorithm, Single Source Shortest Path Dijkstra's Algorithm. Optimal merge Pattern.

UNIT III: Dynamic Programming **9 Hours**

General Concept, Multistage graph, Reliability design, Matrix Chain Multiplication,0/1 Knapsack Problem DP solution, Activity selection problem DP solution, Single Source Shortest Path ,Bellman Ford Algorithm, All pairs shortest paths, Traveling salesman problem.

UNIT IV: Backtracking **9 Hours**

Backtracking: Basic idea, 4 queen's problem ,8 Queens problem, Graph Coloring, Hamiltonian Cycles.

Branch And Bound: Basic idea, LC search, the 15 puzzle problem, LC Branch and Bound, 0/1 Knapsack Problem.

UNIT V: Advanced Algorithms **9 Hours**

Fibonacci Heaps, Network flows, Maximum Flow; Minimum cost circulation, Approximation Algorithms. Introduction to NP Completeness: Basic concepts on NP hard and NP Complete Problems, Discussion on one NP hard graph problem. **Total: 45 Hours**

Reference Books:

1. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", Pearson, 3rd Edition, 2012.
2. S. Sridhar, "Design and Analysis of Algorithms", Oxford University Press, 1st Edition, 2014.
3. Jon Kleinberg and Éva Tardos, "Algorithm Design", Pearson, 1st Edition, 2006.
4. Sara Baase and Allen Van Gelder, "Computer Algorithms: Introduction to Design and Analysis", Pearson, 3rd Edition, 2000.
5. Ellis Horowitz, Sartaj Sahni, and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Universities Press, 2nd Edition, 2008.
6. Parag Himanshu Dave and Himanshu Bhalchandra Dave, "Design and Analysis of Algorithms", Pearson, 2nd Edition, 2014.
7. S.K. Basu, "Advanced Data Structures and Algorithms", PHI Learning, 1st Edition, 2011.
8. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "Introduction to Algorithms", PHI Learning, 3rd Edition, 2009.

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

L-0 T-0 P-2 C-1

CSE170 PR1: Analysis and Design of Algorithms

Course Objectives:

- To give comprehend fundamental algorithmic concepts, complexities, and implement various algorithmic strategies effectively.
- To deliver diverse algorithmic techniques such as Greedy Algorithms, Dynamic Programming, Backtracking, and Branch and Bound for problem-solving across different domains.
- To give skills to optimize solutions for a range of problems including optimization, shortest path, combinatorial, and complexity Organize using advanced algorithms.
- To give understanding of complexity theory, NP Completeness, and NP hard problems while discussing their implications and solutions.

Course Outcomes (COs): Upon completion of this course students will be able to:

6. Understand fundamental algorithms, complexities, proficiently implement sorting and matrix multiplication techniques.
7. Apply Greedy Algorithms for optimization, construct Minimum Cost Spanning Trees, solve Shortest Path problems effectively.
8. Apply Dynamic Programming for complex problem-solving, solve Shortest Path problems efficiently.
9. Apply Backtracking and Branch and Bound techniques for combinatorial problem-solving.
10. Understand advanced algorithms, apply Approximation Algorithms, comprehend NP Completeness and NP hard problems.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	3	-	1	1	-	-	1	-	-	1	1	-	2	1
CO2	3	2	2	-	-	-	-	-	-	-	-	-	3	1	1
CO3	3	2	1	-	-	-	-	1	-	-	1	1	3	1	1
CO4	3	-	1	2	-	-	1	-	-	-	-	1	2	3	1
CO5	1	-	1	1	3	-	1	-	-	-	1	-	2	3	1

High-3 Medium-2 Low-1

PRACTICAL'S

1. Implement a program to demonstrate the effect of input size on time complexity using a logarithmic, linear, and quadratic function.
2. Use a profiler tool to evaluate the performance of recursive and iterative implementations of the Fibonacci sequence.
3. Design and implement a hybrid sorting algorithm combining Merge Sort and Insertion Sort for smaller subarrays.
4. Implement the **Fractional Knapsack Problem** using the greedy approach and analyze its time complexity.
5. Implement **Huffman Coding** to compress a given text string. Display the resulting prefix codes for each character.
6. Implement a program to demonstrate the **Optimal Merge Pattern**. Given a set of files with different lengths, compute the minimum cost to merge them into a single file.
7. Implement the **0/1 Knapsack Problem** using DP. Display the Dynamic Programming table and the items included in the optimal solution.

8. Implement a program to demonstrate the **Floyd-Warshall Algorithm** to find the shortest paths between all pairs of vertices in a graph.
9. Implement a Dynamic Programming-based solution for the **Traveling Salesman Problem**. Use bit masking to represent subsets and optimize the path.
10. Implement the **4 Queens Problem** using backtracking. Display all possible solutions and the decision tree.
11. Implement a backtracking algorithm to find a **Hamiltonian Cycle** in a given graph. Display the path if it exists.
12. Implement a program to solve the **Graph Coloring Problem** using backtracking. Find all valid colorings of a graph for a given number of colors.
13. Implement the **Ford-Fulkerson Algorithm** to compute the maximum flow in a flow network. Display the augmenting paths and the resulting flow.
14. Implement basic operations of a **Fibonacci Heap** (e.g., insert, extract-min, decrease-key, delete) and demonstrate their efficiency.
15. Implement a program for the **Vertex Cover Problem** and implement an approximation algorithm with guaranteed bounds.

Total: 30 Hours

Reference Books:

1. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", Pearson, 3rd Edition, 2012.
2. S. Sridhar, "Design and Analysis of Algorithms", Oxford University Press, 1st Edition, 2014.
3. Jon Kleinberg and Éva Tardos, "Algorithm Design", Pearson, 1st Edition, 2006.
4. Sara Baase and Allen Van Gelder, "Computer Algorithms: Introduction to Design and Analysis", Pearson, 3rd Edition, 2000.
5. Ellis Horowitz, Sartaj Sahni, and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Universities Press, 2nd Edition, 2008.
6. Parag Himanshu Dave and Himanshu Bhalchandra Dave, "Design and Analysis of Algorithms", Pearson, 2nd Edition, 2014.
7. S.K. Basu, "Advanced Data Structures and Algorithms", PHI Learning, 1st Edition, 2011.
8. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "Introduction to Algorithms", PHI Learning, 3rd Edition, 2009.

List of e-Learning Resources:

3. https://nptel.ac.in/
4. https://www.coursera.org/

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Mandsaur University

Bachelor of Technology (Computer Science and Engineering)

Semester-IV

L-2 T-1 P-0 C-3

CSE280 TR1: Operating Systems

Course Objectives

- To have an overview of different types of Operating Systems and Operating System fundamentals.
- To know the components of an Operating System and also able to understand and explore OS design and architectures
- To understand process management, I/O management, memory management including virtual memory, protection and security management.
- To have a thorough knowledge of storage management.
- To learn and analyze process synchronization and deadlock handling and develop skills in memory and system protection

Course Outcomes (COs): Upon completion of this unit students will be able to:

1. Understanding of OS fundamentals, including how OS interacts with hardware and software.
2. Evaluate the performance of various OS components like file systems, memory management, and process scheduling.
3. Apply the key OS functionalities and troubleshoot system performance issues.
4. Analyze the Design and implementation of operating systems.
5. Understand the various memory management schemes.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3	2	1	-	-	-	-	-	-	-	-	-	-	-	2
CO2	3	2	1	-	1	-	1	-	-	-	-	-	-	1	2
CO3	1	3	2	1	1	-	1	-	-	-	1	2	1	2	-
CO4	1	2	3	2	2	1	-	1	-	-	1	2	2	1	1
CO5	3	2	1	1	1	-	-	-	-	-	1	2	2	-	1

High-3 Medium-2 Low-1

UNIT I: Introduction to Operating System — 9 Hours

Introduction: Operating System and its Classification - Batch, Interactive, Multiprogramming, Time sharing, Real Time System, Multiprocessor Systems, Multithreaded Systems, Evolution of Operating System (mainframe, desktop, multiprocessor, Distributed, Network Operating System, Clustered & Handheld System), System Protection, System Calls, Reentrant Kernels, Operating System Structure - Layered structure, Monolithic and Microkernel Systems, Operating system structure Operating System Components, Operating System Functions and Services., System Call & System Boots, Operating system design & Implementations.

UNIT II: File and Device Management — 9 Hours

File management: File concept, types and structures, directory structure, File System Structure, File System Implementation, Allocation methods, Free-space Management, Directory Implementation, access methods and matrices, file security, user authentication, File System Implementation Efficiency and Performance. File system in Linux & Windows.

Device Management : Disk & Drum Scheduling algorithms and policies, Disk Management, Swap-Space Management. I/0 devices organization, I/0 devices organization, I/0 buffering, I/O Hardware, Kernel I/O

subsystem, Transforming I/O request to hardware operations. Device drivers, Path managements, Sub module, Procedure, Scheduler, Handler, Interrupt Service Routine.

UNIT III: Process and Deadlock Management **9 Hours**

Processes: Process Concept, Process States, Process State Transition Diagram, Process Control Block (PCB), Process Scheduling Concepts, CPU Scheduling Concepts, Performance Criteria, Scheduling Algorithms, Multiprocessor Scheduling. Threads and their management.

Process Synchronization: Principle of Concurrency, Producer / Consumer Problem, Inter Process Communication models and Schemes, Critical Section Problem, semaphores, classical problems of synchronization. Synchronization Hardware. Dining Philosopher Problem, Readers Writers Problem. Process Management in Linux.

Deadlock: System model, Deadlock characterization, Methods for deadlock handling,deadlock prevention, deadlock avoidance, deadlock detection, Recovery from deadlock, Combined Approach.

UNIT IV: Memory Management **9 Hours**

Memory Management : Memory Hierarchy, Concepts of memory management, MFT & MVT, logical and physical address space, swapping, contiguous and non-contiguous allocation, paging, segmentation, and paging combined with segmentation. Structure & implementation of Page table. Concepts of virtual memory, Cache Memory Organization, demand paging, page replacement algorithms, allocation of frames, thrashing, Locality of reference. demand segmentation

UNIT V: OS Protections and Distributed operating System **9 Hours**

Protection: System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection.

Distributed operating system:- Types, Design issues, File system, Remote file access, RPC, RMI, Distributed Shared Memory(DSM), Basic Concept of Parallel Processing & Concurrent. Case study of Unix, Linux & Windows

Total: 45 Hours

References

1. "Operating System Principles", by Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 8th Edition, 2008.
2. "Operating systems Internals and Design Principles", by W. Stallings, Pearson, 6th Edition, 2008.
3. "Modern Operating Systems", by Andrew S. Tanenbaum, 2nd Edition, 2001.
4. "Operating Systems: A Design-Oriented Approach", by Charles Crowley, Tata McGraw Hll Education, 1996.
5. "Operating Systems: A Concept-Based Approach", by D M Dhamdhere, Tata McGraw Hill Education, 2nd Edition, 2007.

List of e-Learning Resources:

1. Operating System Fundamentals (nptel.ac.in)
2. https://www.coursera.org/

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Mandsaur University

Bachelor of Technology (Computer Science and Engineering)

Semester-IV

L-0 T-0 P-2 C-1

CSE280 PR1: Operating Systems

Course Objectives

- To have an overview of different types of Operating Systems and Operating System fundamentals.
- To know the components of an Operating System and also able to understand and explore OS design and architectures
- To understand process management, I/O management, memory management including virtual memory, protection and security management.
- To have a thorough knowledge of storage management.
- To learn and analyze process synchronization and deadlock handling and develop skills in memory and system protection

Course Outcomes (COs): Upon completion of this unit students will be able to:

1. Understanding of OS fundamentals, including how OS interacts with hardware and software.
2. Evaluate the performance of various OS components like file systems, memory management, and process scheduling.
3. Apply the key OS functionalities and troubleshoot system performance issues.
4. Analyze the Design and implementation of operating systems.
5. Understand the various memory management schemes.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3	2	1	-	-	-	-	-	-	-	-	-	-	-	2
CO2	3	2	1	-	1	-	1	-	-	-	-	-	-	1	2
CO3	1	3	2	1	1	-	1	-	-	-	1	2	1	2	-
CO4	1	2	3	2	2	1	-	1	-	-	1	2	2	1	1
CO5	3	2	1	1	1	-	-	-	-	-	1	2	2	-	1

High-3 Medium-2 Low-1

Practical's

1. Demonstrate a basic understanding of UNIX commands.
2. Develop a program to analyze the basic functionality of CPU Scheduling algorithms. such as FCFS, SJF, Round Robin, Priority.
3. Create a program to demonstrate the working of file allocation strategies like sequential, Indexed and Linked.
4. Develop a program to implement semaphores.
5. Create a program to implement various file organization techniques such as Single level directory, Two level, Hierarchical and DAG.
6. Develop a program to implement Banker's Algorithm for DeadLock Avoidance.
7. Develop a program to implement an Algorithm for DeadLock Detection.
8. Create a program to implement various page replacement algorithms such as FIFO, LRU and

LFU.

9. Develop a program to implement Shared memory and IPC.
10. Develop a program to implement the Paging Technique of memory management.
11. Develop a program to implement Threading & Synchronization Applications.

Total: 30 Hours

References

1. "Operating System Principles", by Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 8th Edition, 2008.
2. "Operating systems Internals and Design Principles", by W. Stallings, Pearson, 6th Edition, 2008.
3. "Modern Operating Systems", by Andrew S. Tanenbaum, 2nd Edition, 2001.
4. "Operating Systems: A Design-Oriented Approach", by Charles Crowley, Tata McGraw Hll Education, 1996.
5. "Operating Systems: A Concept-Based Approach", by D M Dhamdhere, Tata McGraw Hill Education, 2nd Edition, 2007.

List of e-Learning Resources:

1. Operating System Fundamentals (nptel.ac.in)
2. https://www.coursera.org/

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

BTech (Computer Science & Engineering) (AI)

Semester- IV

L-0 T-0 P-2 C-1

MAT211: Statistical Methods for AI

Course Objectives:

- To teach students the concepts of Python Programming Language with Libraries.
- To learn the basics of math and statistics, and why math is important for AI
- To learn about ethics in making and using AI, and how to make AI safe

Course Outcomes (COs)

1. Understand basics of Python Programming.
2. Create different data structure in Python Programing.
3. Apply NumPy python library for data storage.
4. Apply Panda's python library for data analysis.
5. Create different data graphs in python.

Articulation Matrix

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	1	2	1	1	-	-	-	-	1	-	-	2
CO2	2	2	2	-	1	-	2	2	-	-	-	2	2	2	2
CO3	3	2	-	3	2	2	2	2	1	-	2	-	2	-	-
CO4	3	2	3	2	2	2	2	2	1	1	-	2	-	2	2
CO5	3	2	2	2	1	1	1	-	-	-	1	2	-	1	-

Total: Hours

PRACTICAL'S

1. Implement a python program for Prime Numbers in a range.
2. Implement a python program to check check Perfect numbers.
3. Constract a python program to check check Neon numbers.
4. Implement a python program to check check Palindrome number.
5. Constract a python program for common Prefix and Suffix.
6. Implement a python program to find the closest number in a list.
7. Constract a python program to print patterns using NumPy array.
8. Implement a python program to perform all list operations.

9. Implement a python program to perform all tuple operations.
10. Constract t a python program to perform all dictionary operations.
11. Implement a python program to perform all array operations using Numpy.
12. Implement a python program to convert two reversed list into integer and after sum of integers reverse the result.
13. Implement a python program to perform common data frame operations using Pandas.
14. Constract a python to plot following graph for given data samples.
    (a) Line Plot
    (b) Bar Plot
    (c) Histogram Plot
    (d) Density Plot
    (e) Scatter Plot

Total: 30 Hours

Text books:

1. Achim Klenke, (2014), Probability Theory: A Comprehensive Course Second Edition, Springer, ISBN 978-1-4471-5360- 3

Reference books:

2. Christian Heumann, Michael Schomaker Shalabh (2016), Introduction to Statistics and Data Analysis with Exercises, Solutions and Applications in R, Springer International Publishing, ISBN 978-3-319-46160-1
3. Douglas C. Montgomery, (2012), Applied Statistics and Probability for Engineers, 5th Edition, Wiley India, ISBN: 978- 8-126-53719-8.

BTech (Computer Science & Engineering) (AI)

Semester- IV

L-2 T-1 P-0 C-3

MAT211: Statistical Methods for AI

Course Objectives:

- To teach students the concepts of Python Programming Language with Libraries.
- To learn the basics of math and statistics, and why math is important for AI
- To learn about ethics in making and using AI, and how to make AI safe

Course Outcomes (COs)

6. Understand basics of Python Programming.
7. Create different data structure in Python Programing.
8. Apply NumPy python library for data storage.
9. Apply Panda's python library for data analysis.
10. Create different data graphs in python.

Articulation Matrix

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	1	2	1	1	-	-	-	-	1	-	-	2
CO2	2	2	2	-	1	-	2	2	-	-	-	2	2	2	2
CO3	3	2	-	3	2	2	2	2	1	-	2	-	2	-	-
CO4	3	2	3	2	2	2	2	2	1	1	-	2	-	2	2
CO5	3	2	2	2	1	1	1	-	-	-	1	2	-	1	-

UNIT I: Python Programming Basic **9 Hours**

Python interpreter, IPython Basics, Tab completion, Introspection, %run command, magic commands, matplotlib integration, python programming, language semantics, scalar types. Control flow.

UNIT II: Data Structure, Functions, Files **9 Hours**

tuple, list, built-in sequence function, dictionary, set, functions, namespace, scope, local function, returning multiple values, functions are objects, lambda functions, error and exception handling, file and operation systems

UNIT III: NumPy **9 Hours**

Array and vectorized computation: Multidimensional array object. Creating Nd arrays, arithmetic with NumPy array, basic indexing and slicing, Boolean indexing, transposing array and swapping axes, universal functions, array-oriented programming with arrays, conditional logic as arrays operations, file input and output with array

UNIT IV: Pandas **9 Hours**

Pandas' data structure, series, Data Frame, Index Object, Reindexing, dropping entities from an axis, indexing, selection and filtering, integer indexes, arithmetic and data alignment, function application and mapping, sorting and ranking, correlation and covariance, unique values, values controls and membership, reading and writing data in text format

UNIT V: Visualization with Matplotlib **9 Hours**

Figures and subplots, colors, markers, line style, ticks, labels, legends, annotation and drawing on sublots, matplotlib configuration Plotting with pandas and seaborn: line plots, bar plots, histogram, density plots, scatter and point plots, facet grids and categorical data

Total: 45 Hours

Text books:

4. Achim Klenke, (2014), Probability Theory: A Comprehensive Course Second Edition, Springer, ISBN 978-1-4471-5360- 3

Reference books:

5. Christian Heumann, Michael Schomaker Shalabh (2016), Introduction to Statistics and Data Analysis with Exercises, Solutions and Applications in R, Springer International Publishing, ISBN 978-3-319-46160-1
6. Douglas C. Montgomery, (2012), Applied Statistics and Probability for Engineers, 5th Edition, Wiley India, ISBN: 978- 8-126-53719-8.

B.Tech(Computer Science Engineering)

MUC 010 Quantitative Aptitude- I
Sem –III (CSE-Plain, AI, BCT)

L-4 T-0 P-0 C-4

Course Objectives

- To cater to the needs of outgoing students.
- To prepare students for various examinations and campus interviews.
- To acquaint students with frequently asked patterns in quantitative aptitude and logical reasoning.
- To provide students with strategies and techniques to excel in these areas.
- To give exposure on Logical reasoning.
- To create deeper understanding in data interpretation and progression

Course Outcomes:

1. Understand the concepts of quantitative ability.
2. Understand the basic concepts of logical reasoning Skills.
3. Apply the concept of logical reasoning.
4. Analyze campus placements aptitude papers covering Quantitative Ability, Logical Reasoning Ability and compete in various competitive exams like CAT, CMAT, GATE, GRE, GATE, UPSC, GPSC etc..
5. Evaluate the concept of Logical and Verbal Reasoning.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation).

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	1	-	-	-	-	-	-	-	-	1	-	-
CO2	3	2	1	1	2	-	-	-	-	-	1	-	1	1	-
CO3	2	3	2	2	1	1	-	-	-	-	1	1	2	1	-
CO4	1	3	2	2	1	1	-	-	-	1	1	1	1	2	1
CO5	1	3	2	2	2	1	-	-	-	1	1	1	2	1	1

High-3 Medium-2 Low-1

Unit- I: Quantitative Ability-I **12 Hours**

Number Systems, LCM and HCF, Decimal Fractions, Simplification, Square Roots and Cube Roots, Average, Problems on Ages, Surds & Indices, Percentages, Problems on Numbers

Unit- II: Quantitative Aptitude-Number Theory **12 Hours**

Place value, Face Value, Divisibility test, Prime and Co- prime numbers, number series based on basic formula, Fractions and factors

Unit- III: Basic Data Interpretation **12 Hours**

Tables, Column, Graphs, Bar Graphs, Line Charts, Pie Char,Venn Diagrams

Unit- IV: Logical Reasoning level -I **12 Hours**

Analogy, Blood Relation, Directional Sense, Number and Letter Series, Coding – Decoding

Unit- V: Verbal Reasoning **12 Hours**

Verification of Truth, Logical sequence of Words, Character Puzzels, Series Competition

Total: 60 Hours

Reference Books:

1. Aggarwal, R. S. (2022). A Modern Approach to Verbal & Non Verbal Reasoning. *S. Chand & Company Pvt Limited* .
2. Aggarwal, R. S. (2012). Quantitative Aptitude for Competitive Examinations. *S. Chand & Company Pvt Limited (Unit II, III)*.
3. Praveen, R. V. (2016). *Quantitative Aptitude and Reasoning*. PHI Learning Pvt. Ltd..
4. Allwein, G., & Barwise, J. (Eds.). (1996). *Logical reasoning with diagrams*. Oxford University Press.
5. Sharma, M., & Basu, S. (2024). BELL THE "CAT". *Managing India: The Idea of IIMs and its Changing Contexts*.

List of e-Learning Resources:

1. https://prepinsta.com/
2. https://www.indiabix.com/
3. https://www.javatpoint.com/