B.Tech Curriculum for Computer Science and Engineering

SEMESTER-III

Subject Name	Type	Subject Code	L-T-P	Credit	Contact Hours
	HSS 2		3-0-0	3	3
Probability & Statistics	DC2	MA2L011	3-0-0	3	3
Data Structures	DC 3	CS2L004	3-0-0	3	3
Formal Language and Automata Theory	DC 4	CS2L003	3-1-0	4	4
Digital Logic	SDC 3	CS2L008	2-0-2	3	4
Data Structures Lab	DC Lab 3	CS2P002	0-0-3	2	3
Total				18	20

SEMESTER -IV

Subject Name	Type	Subject Code	L-T-P	Credit	Contact Hours
	HSS 3		3-0-0	3	3
Design & Analysis of Algorithms	DC 5	CS2L002	3-1-0	4	4
Computer Organization & Architecture	DC 6	CS2L009	3-1-0	4	4
Open Elective Basket	OE 1		3-0-0	3	3
Object Oriented Programming	SDC 4	CS2L007	2-0-2	3	4
Design & Analysis of Algorithms Lab	DC Lab 4	CS2P001	0-0-3	2	3
Computer Organization & Architecture Lab	DC Lab 5	CS2P003	0-0-3	2	3
Total				21	24

Syllabus for 2nd Year (3rd & 4th Semesters)

Subject Code: CS2L004	Subject Name: Data Structures	L-T-P-C: 3-0-0-3
Pre-requisite(s): Programming in C		

Theory:

Module 1: **Introduction (8 hours)**: Abstract data type, Stacks, Queues (linear queue, circular queue, double-ended queue), Linked Lists (singly linked list, circular linked list, doubly linked list) and their applications.

Module 2: **Priority queues (4 hours)**: Binary heap.

Module 3: **Sorting (7 hours)**: Comparison based sorting (quick sort, heap sort, merge sort); Decision tree model and lower bound on sorting; Sorting in linear time (radix sort, bucket sort, counting sort).

Module 4: **Hashing (3 hours)**: Hash tables, Hash functions (division method, multiplication method, universal hashing), Collision resolution strategies, Open Hashing (linear probing, quadratic probing, double hashing), Chaining.

Module 5: **Trees (14 hours)**: Binary tree, Binary search trees, Threaded Binary Tree, Balanced binary search trees (AVL tree, Red-black tree); Multi-way search tree (2-3 Tree, 2-4 Tree, B-trees).

Module 6: **Advanced Data Structures (6 hours)**: Disjoint sets, Binomial heap, Tries, Compressed tries, Segment tree, Binary Indexed Tree.

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "Introduction to Algorithms", 4*th* Edition, MIT Press.
2. Ellis Horowitz, Sartaj Sahni, and Susan Anderson Freed, "Fundamentals of Data Structures in C", 2*nd* Edition, Universities Press.

Reference Books:

1. Mark Allen Weiss, "Data structures and Algorithm Analysis in C", 2*nd* Edition, Pearson.
2. Yedidyah Langsam, Moshe J. Augenstein, and Aaron M. Tenenbaum, "Data Struc-ture Using C and C++", 2*nd* Edition, Pearson.
3. Narasimha Karumanchi, "Data Structures And Algorithms Made Easy", 2*nd* Edi-tion, Careermonk Publications.
4. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, "Data Structures and Algorithms in Java", 6*th* Edition, John Wiley & Sons Inc.

Subject Code: CS2L008	Subject Name: Digital Logic	L-T-P-C: 2-0-2-3

Pre-requisite(s): None

Theory:

Module 1: **Combinational Circuits:** Logic gates – AND, OR, NOT, XOR, XNOR, NAND, NOR. combinational logic circuits. Switching functions, Boolean alge-bra – axioms and laws, minimization of switching functions – algebraic method, K-map, Quine-McCluskey's method; Number systems & representation – bi-nary, octal, decimal and hexadecimal, code conversion of numbers. Addition and subtraction of binary and decimal numbers; Com-parators, multiplexer- de-multiplexer, codes and code conversion; Universal logic gates – NAND-NOR,

Module 2: **Sequential Circuits:** Different types of flip-flops – RS, JK, D and T flip-flops; gated, master-slave and edge triggered flip-flops; Different types of Regis-ters; Counters - asyn-chronous and synchronous counters, design of synchronous counters with any specified sequence; Finite state machine – Moore and Mealy models – examples; Synchronous sequential circuit syn-thesis: state transition diagrams, state tables, minimization, state assignments, realization with dif-ferent types of flip-flops; Analysis of synchronous sequential circuits; Concepts of asynchronous sequential circuits - races.

Text Books:

1. S. Lee, "Digital Circuits and Logic Design" Prentice Hall India.
2. D. P. Leach, A. P. Malvino and G. Saha, "Digital Principles and Applications," Tata McGraw Hill.
3. M. Morris Mano, "Digital Logic and Computer Design," Prentice Hall.

Reference Books:

1. Z. Kohavi and N. K. Jha, "Switching and Finite Automata Theory," Tata McGraw Hill.
2. Thomas L. Floyd, Digital Fundamentals, Pearson, 11th Edition, 2017.

Subject Code: CS2L003	Subject Name: Formal Languages and Automata Theory	L-T-P-C: 3-1-0-4
Pre-requisite(s): None		

Theory:

Module 1: **Introduction:** Motivation for studying theory of computation, a quick overview of the subject. Notion of formal language. Language membership problem.

Module 2: **Finite automata and regular expressions:** DFA, NFA (with and without transitions), their equivalence. Proof that for some languages NFAs can be ex-ponentially more succinct than DFAs. Dentition of regular expressions. Proof that FAs recognize, and regular expressions denote the same class of languages (regular languages).

Module 3: **Properties of regular languages:** Pumping lemma and its use to prove non-regularity of a language, closure properties of class of regular languages, decision properties: converting among representations, testing emptiness, etc. Minimization of DFAs, Myhill-Nerode theorem.

Module 4: **Context-free grammars and languages:** Derivation, parse trees. Language generated by a CFG. Eliminating useless symbols, epsilon-productions, unit productions. Chomsky normal form.

Module 5: **Pushdown automata:** Definition, instantaneous description as a snapshot of PDA computation, notion of acceptance for PDAs. Proof that CFGs generate the same class of languages that PDAs accept.

Module 6: **Properties of context-free languages:** Pumping lemma for context-free languages and its use to prove a language to be not context-free. Closure properties of the class of context- free languages. CYK algorithm for CFL membership, testing emptiness of CFLs.

Module 7: **Turing machines:** Informal proofs of undecidability. Dentition of TM, snap-shot of TM computation, notion of acceptance. Model Robustness, Church-Turing hypothesis.

Module 8: **Undecidability:** R.E. and recursive languages. Turing machine codes, diago-nalization, Universal language, Reducibility. Rices theorem. Posts correspon-dence problem.

Intractability: Class P, Classes NP, co-NP. Polynomial time reductions. NP-completeness, NP-hardness. Cook-Levin theorem. Some NP-completeness proofs.

Text Books:

1. Michael Sipser: Introduction to the Theory of Computation, 3rd edition, PWS Pub-lishing Com-pany, 2012.
2. E. Hopcroft, R. Motwani and J. D. Ullman: Introduction to Automata Theory, Languages and Computation. Low priced paperback edition, published by Pearson Education, 2007.

Reference Books:

1. H. R. Lewis and C. H. Papadimitriou. Elements of the Theory of Computation, Eastern economy edition, 1998.

Subject Code: CS2P002	Subject Name: Data Structures Lab	L-T-P-C: 0-0-3-2

Pre-requisite(s): Programming in C

Applications of the Data Structures discussed in the class: Two-way Stack, Dequeue, Priority queue, Sorting Algorithms, Dictionary data structures for string handling, Cy-cle detection and removal, Application of BST, AVL Trees, Data structure for big data maintenance, Tree traversal algorithms, efficient insertion and deletion algorithm with variations across different applications, Hashing Techniques, Application of Union- Find Data Structure for set theoretic problems.

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, R.L. Rivest. Introduction to Algorithms, Prentice Hall of India Publications, 3rd Edition 2015.
2. J. Kleinberg and E. Tardos. Algorithm Design, Pearson, 2006.

Reference Books:

1. Fundamentals of Data Structures in C by Horowitz, Sahni, and Anderson-Freed, Universities Press
2. Algorithm Design: Foundations, Analysis and Internet Examples by Goodrich and Tamassia, John Wiley
3. Computer Algorithms by Baase and Van Gelder, Pearson.
4. Algorithms by Sedgewick and Wayne, Addison Wesley, 2011.

Subject Code: CS2L002	Subject Name: Design and Analysis of Algorithms	L-T-P-C: 3-1-0-4

Pre-requisite(s): Programming in C

Learning Outcome(s):

- To estimate the time and space that an algorithm uses, particularly in terms of the input size, using concepts like asymptotic and amortized analysis.
- To apply algorithm design paradigms like divide and conquer, dynamic program-ming, greedy algorithms, and linear programming for a given problem.
- To model several real-world problems as graphs, and implement graph algorithms to find optimal solutions for the problems.
- To design and implement pattern matching algorithms.
- To identify problems that are unlikely to have efficient (polynomial-time) solutions, meaning that finding a solution takes time that grows exponentially with the input size.

Theory:

Module 1: **Analysis Techniques (8 hours)**: Asymptotic notations and their signifi-cance, complexity analysis of algorithms, worst case and average case, solving recurrences (the substitution method, the recursion-tree method, the master method).

Module 2: **Design Techniques (16 hours)**: Divide and Conquer, Greedy Algorithms, Dynamic Programming, Branch and Bound, Backtracking Techniques.

Module 3: **Elementary Graph Algorithms (16 hours)**: DFS, BFS, Topological Sort, Strongly Connected Components, Minimum Spanning Trees, Single source shortest path, All pair shortest path; Network Flows: Ford Fulkerson Algo-rithm for Max Flow-Min Cut; Bipartite Matching.

Module 4: **String matching Algorithms (8 hours)**: The naive string-matching algo-rithm, The Rabin-Karp algorithm, String matching with finite automata, The Knuth-Morris-Pratt algorithm.

Module 5: **Theory of NP-Completeness (8 hours)**: Polynomial-time verification, NP-completeness and reducibility, NP-Completeness proofs, NP-Complete prob-lems.

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "Introduction to Algorithms", 4^{th} Edition, MIT Press.
2. Jon Kleinberg and Eva Tardos. "Algorithm Design", 1^{st} Edition, Pearson.
3. Sanjoy Dasgupta, Christos H. Papadimitriou and Umesh V. Vazirani, "Algorithms", 1^{st} Edition, McGraw-Hill.

Reference Books:

1. Michael T. Goodrich and Roberto Tamassia, "Design and Analysis of Algorithms", 1^{st} Edition, Wiley.
2. U Steven S. Skiena, "The Algorithm Design Manual", 3^{rd} Edition, Springer.
3. Gilles Brassard and Paul Bratley, "Fundamentals of Algorithmics", 1^{st} Edition, Pear-son.
4. Sara Baase and Allen Van Gelder, "Computer Algorithms: Introduction to Design & Analysis", 3^{rd} Edition, Pearson.

Subject Code: CS2P001	Subject Name: Design and Analysis of Algorithms Lab	L-T-P-C: 0-0-3-2
Pre-requisite(s): None		

Implementation of Algorithms covered in Theory Class: Sorting, Matrix Multiplications, Trees, Graph Traversals, Shortest Paths, Network Flows, Algorithms, Network Flows. [This will involve running the algorithms under varying input sets and measuring running times, use of different data structures for the same algorithm (wherever applicable) to see its effect on time and space, comparison of different algorithms for the same problem etc.] Design of Algorithms: This will involve design and implementation of algorithms for prob-lems not covered in class but related to topics covered in class.

[The exact set of algorithms to design and implement is to be decided by the instructor. In addition, there will be at least one significantly large design project involving some real world application. An efficient design of the project should require the use of multiple data structures and a combination of different algorithms/techniques.]

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, R.L. Rivest. Introduction to Algorithms, Prentice Hall of India Publications, 3rd Edition 2015.
2. J. Kleinberg and E. Tardos. Algorithm Design, Pearson, 2006.

Reference Books:

1. Sara Baase and Allen Van Gelder. Computer Algorithms: Introduction to Design and Analysis, Pearson education (Singapore) Pvt. Ltd, New Delhi 2007.
2. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman. The Design and Analysis of Computer Algorithms, Pearson Education (Singapore) 2006.
3. Algorithmics: Theory and Practice by Brassard and Bratley, Prentice Hall

Subject Code:CS2L009	Subject Name: Computer Organization & Architecture	L-T-P-C: 0-0-3-2

Pre-requisite(s): Programming in C

Theory:

Module 1: **Fundamentals and Organization of Computers:** Practical look at the Computer Hardware, Major components of a computer, Process technology, performance wall, power wall, Introduction to Assembly Language, Basic Met-rics for evaluation of a better computer, Laws of computer systems: Moore´s Law, Amdahl´s law, and Little´s law, History of Computers, Von-Neuman Ar-chitecture, Flynn´s classifications of Computers, Dennard´s Scaling, Register Or-ganization, Bus and Interconnects, Memory Hierarchy, Internal and External Memory, System-on-Chip

Module 2: **Instruction Set Architecture:** Instruction Formats and Addressing Mode, CISC v RISC, Popular ISAs like X86, MIPS, ARM, and RISC-V, Constants, Pointers, Operands and Data Types, Types of instructions, ALU instructions, Memory instructions: Location, alignment, addresses, and operation, branch instructions, subroutines, memory stack

Module 3: **Data Representation:** signed number representation, fixed and floating point representations, character representation. Computer arithmetic - inte-ger addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication - shift-and-add, Booth multiplier, carry save multiplier, etc. Di-vision - non-restoring and restoring techniques, floating point arithmetic.

Module 4: **Single-cycle computer design:** register files, control register, Instruction Fetch, Decoder, Execution unit, memory unit. non-pipelined data path design, Control Unit Design.

Module 5: **Pipelined Computer Design:** Concept of pipelining, Introduction to In-struction Pipeline, controlling an instruction pipeline, pipelined data path and control path, pipeline hazards, structural hazards, data hazards, control hazards, stalling and bubbles, Interlocking, operand forwarding, dealing with branches.

Module 6: **Cache Memory Design:** Introduction to memory hierarchy, four major ques-tions of cache memory, Cache addressing, data placement, data identification, cache replacement policy, write policy, simple finite state machine and imple-mentation of a cache controller, Cache Optimizations and cache performance, Understanding performance with simulation.

Module 7: **Virtual Memory and Main Memory Design:** Basic main memory archi-tecture, Four major questions of virtual memory, Virtual memory vs physical memory, virtual address vs physical address vs DRAM address, Introductions to TLB, Three major questions on TLB.

Module 8: **Peripheral devices and their characteristics:** Input-output subsystems, I/O transfers - program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes - role of interrupts in process state transitions.

Text Books:

1. R.E.Bryant and D.R.O’Hallaron, Computer Systems: A Programmer’s Perspective, Pearson Education LTD, 2016

2. Computer Organization and Design: RISC-V Edition, The Hardware Software In-terface, 1st Edition by Patterson and Hennessy, Morgan Kaufmann, 2017

Reference Books:

1. Computer Organization and Design: MIPS Edition, The Hardware Software Inter-face, 6th Edition by Patterson and Hennessy, Morgan Kaufmann, 2021
2. Computer Organization and Design: ARM Edition, The Hardware Software Inter-face, Patterson and Hennessy, Morgan Kaufmann, 2016
3. Understanding Pointers in C & C++: Pointers in all its forms: Fully working Ex-amples and Applications of Pointers, Yashvant Kanetkar, BPB Publication, 2019

Subject Code: CS2P003	Subject Name: Computer Organization & Architecture Lab	L-T-P-C: 0-0-3-2
Pre-requisite(s): Programming in C		

Implement Multiplier Circuit and Divider Circuit in System Verilog
Implement RISC-V based Single Cycle Computer using C++ and System Verilog Convert the single cycle computer into pipelined computer using C++ and System Verilog Implement a simple cache controller with timing model using C++ and System Verilog Implement a simple TLB controller using C++ and System Verilog
Simulation result analysis with gem5, Generating performance statistics using profiling, Tracing with pin tool or dynamo rio, Source code analysis and performance tuning

Text Books:

1. Computer Organization and Design: RISC-V Edition, The Hardware Software In-terface, 1st Edition by Patterson and Hennessy, Morgan Kaufmann, 2017

Reference Books:

1. Computer Organization and Design: MIPS Edition, The Hardware Software Inter-face, 6th Edition by Patterson and Hennessy, Morgan Kaufmann, 2021
2. Computer Organization and Design: ARM Edition, The Hardware Software Inter-face, Pat-terson and Hennessy, Morgan Kaufmann, 2016
3. Computer Architecture: A Quantitative Approach, 6th Edition, by Patterson and Hennessy, Morgan Kaufmann, 2017

Subject Code: CS2L007	Subject Name: Object Oriented Programming	L-T-P-C: 2-0-2-3
Pre-requisite(s): Programming in C		

Module 1: **Object-Oriented Programming with C++ / Java**: Abstraction, Classes and Objects, Encapsulation, Polymorphism, Inheritance, Exception Handling, Files and Streams, I/O Redirection, Templates, Standard Template Library (STL): Algorithms, Containers, Iterators, Vector, Stack, Queue, Map, Set, Multithreading, Handling Databases.

Module 2: **Methods and tools**: Design principles and patterns that support the de-velopment of maintainable, reusable and extensible software, UML: use case diagrams, class diagram, sequence diagrams, and state diagrams.

Module 3: **Design Patterns**: Creational Design Patterns(Factory Method, Abstract Factory Method, Singleton Method, Prototype Method, Builder Method), Structural Design Patterns (Adapter Method, Bridge Method, Compos-ite Method, Decorator Method, Facade Method, Flyweight Method, Proxy Method), Behavioral Design Patterns (Chain Of Responsibility Method, Com-mand Method, Interpreter Method, Mediator Method, Memento Method, Ob-server Method, State Method, Strategy Method, Template Method, Visitor Method), Design Patterns in Different Languages.

Text Books:

1. Grady Booch, Object Oriented Analysis and Design ,3rd Edition, Pearson, 2015.
2. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: El-ements of Reusable Object-Oriented Software. Awl, 2010.
3. Alan Dennis, Barbara Haley Wixom, David Tegarden, Systems Analysis and Design: an Object-oriented Approach with UML (5th edition).

Reference Books:

1. Martin Fowler, UML Distilled With Access codes: A Brief Guide to the Standard Object Modeling Language, Pearson, 2015.
2. Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Pearson, 2nd Edition, 2006.

Subject Code: MA2L011	Name: Probability and Statistics	L-T-P-C: 3-0-0-3

Prerequisite: None

Module 1: (Classical and axiomatic definitions) (6 hours)
Classical and axiomatic definitions of probability, addition rule, Conditional probability, Multiplication rule, Bayes' theorem, Independence

Module 2: (Discrete and continuous random variables) (9 hours)
Discrete and Continuous random variables, Standard distributions: Uniform, Binomial, Geometric, Poisson, Exponential, Gamma, Normal

Module 3: (Jointly distributed random variables) (7 hours)
Jointly distributed random variables, Marginal and conditional distributions, Expectation, Moments, Probability generating functions, Moment generating functions

Module 4: (Laws of large numbers) (7 hours)
Chebyshev's inequality, weak and strong laws of large numbers, functions of random variables and their distributions

Module 5: (Central Limit Theorem) (9 hours)
Central Limit Theorem, distributions of sample mean and variance, Chi-square, t and F distributions, point estimation, interval estimation, maximum likelihood estimation

Module 6: (Hypothesis Testing) (4 hours)
Null and alternative hypotheses, critical and acceptance regions, Neyman–Pearson lemma, one-sample and two-sample tests for normal populations

Text Books:

1. Probability and Random Processes by Grimmett G. R. and Stirzaker D. R., Oxford University Press, 2001
2. An Introduction to Probability and Statistics by Vijay K. Rohatgi, A. K. M. E. Saleh, Wiley Series in Probability and Statistics, 2015
3. Introduction to Probability and Statistics for Engineers and Scientists by Ross S. M., Academic Press, 2014

Reference Books:

1. Probability and Statistics in Engineering by Hines W. W., Montgomery D. C., Goldsman D. M., Borror C. M., 4th Edition, John Wiley and Sons, 2002
2. Introduction to Probability by Bertsekas D. P. and Tsitsiklis J. N., Athena Scientific, 2002
3. Probabilistic Methods by Alon N. and Spenser J., 4th Edition, John Wiley and Sons, 2016

Subject Code: CS3L010	Subject Name: Introduction to Information Security	L-T-P: 3-0-0-3

Pre-requisite(s): Data Structures

Learning Outcome(s):

- Analyze and classify security incidents using the CIA triad and assess their likely business impact.
- Perform a qualitative risk assessment with a risk matrix and recommend prioritized mitigation strategies.
- Design and justify an access control scheme for a given system, including authenti-cation and MFA requirements.
- Apply cryptographic primitives (symmetric/asymmetric encryption, hashing, and digital signatures) to secure data in transit and at rest, and explain appropriate protocol choices.

Theory:

Module 1: **Security Fundamentals (12 hours)**: Introduction to Network Security, CIA triad, security models, and common threats like malware, phishing, and social engineering.

Module 2: **Risk Management and Policies (6 hours)**: Introduces risk assessment, risk matrices, qualitative and quantitative risk evaluation, security policies, standards (e.g., ISO 27001), contingency planning, and business continuity. .

Module 3: **Access Control and Authentication (6 hours)**: Identification, authen-tication methods (passwords, biometrics, Multifactor Authentication), autho-rization models, and physical security, basics of Mandatory Access Control, Discretionary Access Control

Module 4: **Cryptography Basics (12 hours)**: Symmetric/Asymmetric encryption such as AES and RSA, hashing, digital signatures, PKI, and common protocols.

Module 5: **Network and System Security (6 hours)**: Firewalls, VPNs, intru-sion detection, OS hardening, software vulnerabilities (e.g., buffer overflows, SQL/NoSQL injection, XSS), and web/mobile security.

Text Books:

1. William Stallings, "Cryptography and Network Security: Principles and Practice", Pearson

Reference Books:

1. Mike Rosulek, "The Joy of Cryptography", MIT Press

Subject Code: CS3L011	Subject Name: Computer Graphics	L-T-P-C: 3-0-0-3
Pre-requisite(s): None		

Module 1: **Graphics devices**: Display devices, Input devices, Rendering pipeline.

Module 2: **Raster Graphics**: Line and Circle drawing algorithms, Windowing.

Module 3: **Clipping**: Cohen and Sutherland line clipping, Cyrus-beck clipping method, 2D and 3D Geometrical Transformations,

Module 4: **Viewing Transformations**: parallel and perspective projection, Curves and Surfaces: Cubic splines, Bezier curves, B-splines, Tensor product sur-faces, Surface of revolution Sweep surfaces, Fractal curves and surfaces, Hid-den line/surface removal methods, Illu-mination model, Polygon Shading: Gouraud, Phong, Introduction to Ray-tracing, Ani-mation.

Module 5: **Graphics Programming**: Getting started with OpenGL, Input and In-teraction in OpenGL, Geometrical Objects and Transformations in 2D and 3D, homogeneous coor-dinates, matrix representation, windows and viewports, Viewing in 3D, projections, hidden surface removal, Light, shading and ma-terials. Illumination and Shading, light sources, (surface detail, ray tracing, radiosity).

Text Books:

1. Donald Hearn, M. Pauline Baker, Computer Graphics, 2nd edition, C version, Pren-tice Hall, 1996.
2. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics : Principles & Practices, Addison Wesley Longman, 2nd edition in C, 1994

Reference Books:

1. M. de Berg, M. Van Kreveld, M. Overmars, and O. Schwarzkopf, Computational Geome-try: Algorithms and Applications (3rd Edition), Springer, 2008.
2. Computational Geometry Algorithm Library (CGAL): http://www.cgal.org

Subject Code: CS3L012	Subject Name: Introduction to Data Science	L-T-P-C: 3-0-0-3

Pre-requisite(s): Programming in C, Probability and Statistics

Learning Outcome(s):

- Explain the fundamental concepts and applications of Data Science.
- Perform data collection, preprocessing, analysis, and visualization on structured and unstructured datasets.
- Apply statistical and Machine Learning techniques for data-driven decision making.
- Analyze patterns and extract insights from real-world datasets.
- Build predictive models and evaluate their performance.
- Explain the ethical, social, and practical aspects of Data Science systems.

Theory:

Module 1: **Introduction to Data Science (4 hours)**: Data science lifecycle, types of data, structured and unstructured data, data sources, applications of Data Science, overview of data-driven decision making.

Module 2: **Data Collection, Preprocessing and Visualization (6 hours)**: Data acquisition, data cleaning, handling missing values and outliers, data transfor-mation, feature engineering, exploratory data analysis, data visualization.

Module 3: **Statistical Foundations for Data Science (8 hours)**: Descriptive statis-tics, probability distributions, sampling techniques, correlation and covariance, hypothesis testing, confidence intervals, statistical inference.

Module 4: **Predictive Analytics and Data Mining (12 hours)**: Classification tech-niques (Decision Trees, Naive Bayes, k-NN), clustering techniques (K-Means, Hierarchical Clustering, DBSCAN), association rule mining, anomaly detec-tion, recommendation systems, evaluation metrics and analytics applications.

Module 5: **Big Data and Data Engineering Basics (6 hours)**: Introduction to Big Data, Hadoop ecosystem, and MapReduce concepts, distributed data process-ing.

Module 6: **Modern Data Science Applications (6 hours)**: Time-series analysis, text analytics, social network analytics, dashboards and business intelligence, in-troduction to Generative AI assisted analytics, ethics, privacy, and responsible data science.

Text Books:

1. Joel Grus, “Data Science from Scratch”, 2nd Edition, O’Reilly Media.
2. Cathy O’Neil and Rachel Schutt, “Doing Data Science”, O’Reilly Media.
3. Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, “An Introduc-tion to Statistical Learning”, Springer.
4. Jiawei Han, Micheline Kamber and Jian Pei, “Data Mining: Concepts and Tech-niques”, 4th Edition, Morgan Kaufmann.

Reference Books:

1. Trevor Hastie, Robert Tibshirani and Jerome Friedman, “The Elements of Statistical Learning”, Springer.

2. Jake VanderPlas, “Python Data Science Handbook”, O’Reilly Media.
3. Foster Provost and Tom Fawcett, “Data Science for Business”, O’Reilly Media.
4. Wes McKinney, “Python for Data Analysis”, O’Reilly Media.
5. Anand Rajaraman and Jeffrey Ullman, “Mining of Massive Datasets”, Cambridge University Press.

Subject Code: CS3L013	Subject Name: Operations Research	L-T-P-C: 3-0-0-3
Pre-requisite(s): None		
Linear Programming, Transportation Problem, Assignment Problem, Integer Linear Programming, Travelling Salesperson Problem, PERT-CPM, Queueing, Sequence and Scheduling, Dynamic Programming, Non-linear Programming.		
Text Books: 1. G. Srinivasan, Operations Research, 3rd Edition, PHI, 2017. 2. Vasek Chvatal, Linear Programming, W H Freeman, 1983. **Reference Books:** 1. Hamdy A. Taha, Operations Research, 10th Edition, Pearson, 2017.		