Semester–IV

S.No.	Course Code	Course Title	L	P	Credits
1	CSE260C	Design and Analysis of Algorithms	3	0	3
2	CSE261C	Computer Architecture and Organization	3	0	3
3	CSE262C	Web Technologies	3	0	3
4	CSE263C	Software Engineering	3	0	3
5	CSE264C	Data Communication	3	0	3
6	CSE265C	Design and Analysis of Algorithms Lab	0	2	1
7	CSE266C	Web Technologies Lab	0	2	1
8	CSE268C	Project-I	0	2	1
Total			**15**	**6**	**18**

Course Code: **CSE260C**	Course Title: **Design and Analysis of Algorithms**	Credits: **03** L – **3** P – **0**

Course Outcomes (COs):

- Understand and use asymptotic notations to analyze the performance of algorithms
- Understand and analyze the design of algorithms using Brute force, Divide and Conquer, Dynamic
- Programming, Greedy technique, Backtracking, Branch and Bound techniques.
- Compare and contrast various search and sorting techniques.
- Apply the various algorithms to solve problems and analyze their efficiency.

Unit – I **(10)**

Introduction: Algorithm Design paradigms- motivation, Concept of algorithmic efficiency, Run time analysis of algorithms, Asymptotic Notations.
Divide and Conquer: Structure of divide and conquer algorithms: examples, Binary search, Merge Sort, Quick sort, Analysis of divide and conquer run time. Recurrence Relations, Master Theorem for solving Recurrence Relations.

Unit – II **(8)**

Greedy method: Overview of the greedy paradigm, examples of exact optimization solutions (minimum cost spanning tree), approximate solution (Knapsack problem), Huffman coding, Single source shortest path.

Unit – III **(9)**

Dynamic Programming: Overview, difference between dynamic programming and divide and conquer, applications: Shortest Path in Multistage Graph, Non-fractional (0/1) Knapsack problem, Matrix Chain Multiplication, Travelling salesman problem, Longest common sequence.

Unit – IV **(8)**

Graph searching and traversal: Overview, traversal methods, depth first and breadth first search. Dijkstra's and Bellman-Ford Algorithm for finding Single source shortest paths. All pair shortest paths and matrix multiplication, Floyd-Warshall algorithm for all pair shortest paths.

Unit – V **(10)**

Back Tracking: Overview, 8-queen problem and Knapsack problem.
Branch and Bound: LC searching, bounding, FIFO branch and bound, Applications: 0/1 Knapsack problem, Travelling salesman problem.
Computational complexity: Complexity measures, Polynomial vs non-polynomial time complexity; NP hard and NP complete classes, examples.

Textbook:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, Clifford Stein, "*Introduction to Algorithms*", 2nd Ed., PHI, 2004.

Reference Books:

1. Ellis Horowitz and Sartaz Sahani, "*Computer Algorithms*", Galgotia Publications, 1999.
2. V. Aho, J. E. Hopcroft, J. D. Ullman, "*The Design and Analysis of Computer Algorithms*", Addition Wesley, 1998.
3. D. E. Knuth, "*The Art of Computer Programming*", 2nd Ed., Addison Wesley, 1998.

Course Code: **CSE261C**	Course Title: **Computer Architecture and Organization**	Credits: **03** L – **3** P – **0**

Course Outcomes (COs):

- Describe how different functional units in a computer system operate, interact and communicate.
- Describe the detailed architecture of the central processing unit, control unit, input-output unit and memory unit.
- Describe the representation, arithmetic and computation of data at machine level.
- Describe how various memory units and input-output devices are accessed in a computer system.
- Describe how the throughput of a computer system can be increased using pipelining.

Unit – I (8)

Register Transfer and Micro-operations: Defining computer architecture and computer organization, Register Transfer Language, Register Transfer, Bus and Memory Transfers, Bus system construction, Arithmetic Micro-operations, Logic Micro-operations, Shift Micro-operations, Arithmetic Logic Shift Unit.

Unit – II (8)

Basic Computer Organization and Control Unit Design: Instruction Codes, Computer Registers, Computer Instructions, Hardwired Control, Timing and Control, Instruction Cycle, Memory Reference Instructions, Input-Output and Interrupt, Complete Computer Description, Design of Basic Computer, Design of Accumulator Logic, Data-path, Microprogrammed Control: Control Memory, Address Sequencing, Micro Program Example, Design of Control Unit.

Unit – III (9)

Central Processing Unit: Introduction, General Register Organization, Stack Organization, Machine Instructions, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer.

Computer Arithmetic: Addition and Subtraction, Multiplication Algorithms, Division Algorithms, Floating-Point Arithmetic Operations, Decimal Arithmetic Unit, Decimal Arithmetic Operations.

Unit – IV (10)

Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access (DMA), Input-Output Processor, Serial Communication.

Pipelining: Pipelining Basics, Arithmetic Pipeline, Instruction Pipeline, Pipeline Hazards.

Unit – V (10)

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory, Memory Management Hardware.

Introduction to Multiprocessors.

Textbook:

1. Mano, M. Morris. *Computer system architecture*. 3rd Edition, Prentice-Hall.

Reference Books:

1. Hayes, John P. *Computer architecture and organization*. McGraw-Hill.
2. Stallings, William. *Computer organization and architecture: designing for performance*. Pearson.
3. Hamacher, V. Carl, Zvonko G. Vranesic, Safwat G. Zaky, Zvonko Vransic, and Safwat Zakay. *Computer organization*. McGraw-Hill.

Online Resources:

1. https://youtube.com/playlist?list=PLdl2B3KkY5upvcVCwXImlzDhop9Xhua9M&si=0xqCH_iNXn6urxM1
2. https://archive.nptel.ac.in/courses/106/105/106105163/
3. https://onlinecourses.nptel.ac.in/noc23_cs67/preview

Course Code: **CSE262C**	Course Title: **Web Technologies**	Credits: **03** L – **3** P – **0**

Course Outcomes (COs):

- To understand the basic concepts of web programming and internet,
- To understand how the client-server model of Internet programming works,
- To understand interactive web applications,
- To learn the latest technologies, including JavaScript, Node, and Web3 development,
- To understand the latest frontend and backend web development technologies.

Unit – I (5)

Introduction to Web System
Internet Overview, WWW, Web Protocols, Web Browsers and Web Servers, Web System Architecture. URL, Domain Name, Client and Server-side Scripting. MVC, Implementing Model-View-Controller.

Unit – II (10)

HTML and CSS
HTML5 Basics: Formatting, Colours, Images, Links, Tables, Lists, Layout, Forms, Canvas, Media.
CSS3 Basics: Selectors, Box Model, Backgrounds and Borders, Text Effects, Advanced Features.

Unit – III (10)

JavaScript and jQuery
JavaScript Basics: Functions, Arrays, DOM, Built-in Objects, Regular Expressions, Event handling, JavaScript Form Validation.
jQuery Basics: jQuery with Websites, Selecting Elements with jQuery, Manipulating Styles, Text, and Attributes with jQuery, Adding Event Listeners with jQuery, Adding and Removing Elements with jQuery, Website Animations with jQuery.

Unit – IV (9)

Backend Web Development (Node.js)
Node.js Basics: Node JS Modules, Functions, Buffer, Module, Modules Types, Core Modules, Local Modules, Modules Exports, The Node REPL (Read Evaluation Print Loops), The NPM Package Manager.

Unit – V (9)

MongoDB and Mongoose:
MongoDB: Introduction to NoSQL Databases, MongoDB Ecosystem, Replica Sets and Clusters, Advantages of MongoDB Databases, MongoDB Query Languages, MongoDB CRUD Operations: Create, Read, Update, and Delete.
Mongoose: Introduction to Mongoose, reading from the database, Data Validation, Updating and Deleting Data, Establishing Relationships and Embedding Documents using Mongoose.

Textbooks:

1. *"HTML, CSS, and JavaScript All in One*," Julie C. Meloni and Jennifer Kyrnin, Pearson Education India.
2. *"Node.js Web Development,"* David Herron, Packt Publishing Limited.
3. *"MongoDB the Definitive Guide,"* Shannon Bradshaw (Author), Eoin Brazil, and Kristina Chodorow, Shroff/O'Reilly.

Reference Books:

1. *"JavaScript and jQuery: Interactive Front-End Web Development,"* Jon Duckett, Wiley.
2. *"Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node,"* Scott Meyers, Shroff/O'Reilly
3. *"MERN Quick Start Guide,"* Eddy Wilson Iriarte Koroliova, Packt Publishing Limited.

Online Resources:

1. https://www.udemy.com/course/complete-web-development-course/
2. https://www.udemy.com/course/mongodb-the-complete-developers-guide/
3. https://www.coursera.org/learn/web-development

Course Code: **CSE263C**	Course Title: **Software Engineering**	Credits: **03** L – **3** P – **0**

Course Outcomes (COs):

- Understanding and applying the software engineering lifecycle models by demonstrating competence in communication, planning, analysis, design, construction, and deployment
- Translate a requirements specification into an implementable design, following a structured and organized process.
- Defining the basic concepts and importance of Software project planning concepts like cost estimation, and scheduling.
- Applying different testing and debugging techniques and analyzing their effectiveness.
- Defining software maintenance concepts, software quality and reliability on the basis of international quality standards.

Unit – I **(9)**

Introduction: Software engineering discipline-Evolution and Impact, program vs software product. Software Crisis, Software engineering a layered technology – processes, methods and tools. Software life cycle models: Waterfall, Prototype, Evolutionary and Spiral models, Overview of object oriented software development and Agile software.

Unit – II **(9)**

Software Requirement Analysis and Specifications: Problem Analysis, Data Flow Diagrams, Data Dictionaries, Entity-Relationship diagrams, Software Requirement and Specifications. Software Project Planning, Project Manager Responsibilities, Cost estimation, static, single and multivariate models, COCOMO model. Risk Management, Software Configuration Management.

Unit – III **(9)**

Software Design: Cohesion and Coupling, Classification of Cohesiveness and Coupling, Function Oriented Design, Object Oriented Design. Software Reliability: Introduction, Failure and Faults, Reliability Models: Basic Model, Logarithmic Poisson Model, Calendar time Component.

Unit – IV **(9)**

Software Testing: Software process, Functional testing: Boundary value analysis, Equivalence class testing, Decision table testing, Cause effect graphing, Structural testing: Path testing, Data flow and mutation testing, Unit testing, Integration and System testing, Black box testing, White- box testing, Debugging, Testing Tools and Standards, Using testing tools in lab.

Unit – V **(9)**

Software Quality, Software Quality Management System, ISO 9000, 9001, Capability Maturity Model, Personal Software Process, Six Sigma, Software Maintenance, Reverse Engineering, Software Reengineering, Configuration Management.

Textbooks:

1. "*Software Engineering A. Practitioner's Approach*", by Pressman, MGH.
2. "*Fundamental of Software Engineering",* by Rajib Mall, PHI.

Reference Books:

1. *"Software Engineering",* by James F. Peters, Wiley.
2. *"Software Project Management from Concept to Development",* by Kieron Conway, Dreamtech Press.
3. "*Software Engineering*", by Sommerville, Pearson Education.
4. "*Software Engineering",* by Jawadekar, TMH.

Course Code: **CSE264C**	Course Title: **Data Communication**	Credits: **03** L – **3** P – **0**

Course Outcomes (COs):

- Understand the model of data communication
- Understand the working of various Transmission Media.
- Understand various signal conversion techniques.
- Understand various error detection and correction techniques.
- Understand multiplexing techniques

Unit – I (8)

Introduction to Data Communications and Networking, Communications Model, Data Communications, Networks, The Internet, Data Transmission: Concepts and Terminology, Analog and Digital Data Transmission, Transmission Impairments, Channel Capacity.

Unit – II (8)

Guided Transmission Media, Wireless Transmission, Wireless Propagation, Line-of-Sight Transmission, Data transmission: simplex, half duplex and full duplex, Asynchronous and Synchronous Transmission.

Unit – III (10)

Digital Data and Digital Signals, Digital Data and Analog Signals, Analog Data and Digital Signals, Analog Data and Analog Signals.

Unit – IV (9)

Types of Errors, Error Detection, Error Correction, Line Configurations.
Flow Control, Error Control, High-Level Data Link Control (HDLC)

Unit – V (10)

Multiplexing Techniques: Frequency Division Multiplexing, Synchronous Time-Division Multiplexing Statistical Time-Division Multiplexing, Asymmetric Digital Subscriber Line
Spread Spectrum, Frequency Hopping Spread Spectrum, Direct Sequence Spread Spectrum, Code-Division Multiple Access

Textbooks:

1. William Stallings: Data and Computer Communications, 9th Ed, PHI
2. Data Communications and Networking: Behrouz A. Forouzan

Reference Books:

1. Andrew Tanenbaum, “Computer Networks” PHI
2. Sklar, “Digital Communications fundamentals and Applications”
3. Keizer, “Local Area Networks” McGraw Hill

Course Code: **CSE265C**	Course Title: **Design and Analysis of Algorithms Lab**	Credits: **01** L – **0** P – **2**

Course Outcomes (COs):

- Understand and use asymptotic notations to analyze the performance of algorithms
- Understand and analyze the design of algorithms using Brute force, Divide and Conquer, Dynamic
- Programming, Greedy technique, Backtracking, Branch and Bound techniques.
- Compare and contrast various search and sorting techniques.
- Apply the various algorithms to solve problems and analyze their efficiency

List of Experiments

S.No.	Topic
1.	Simple Experiments on time and space complexity of a program
2.	Sort a set of elements using the Quick sort, Merge sort
3.	Implement Knapsack Problem using Greedy Algorithm
4.	Implement Huffman Codes using Greedy Algorithm
5.	Implement 0/1 Knapsack problem using Dynamic Programming.
6.	Implement Matrix Chain Multiplication using Dynamic Programming
7.	Implement Traveling Salesman Problem using Dynamic Programming
8.	Implement Longest common subsequence using Dynamic Programming
9.	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.
10.	Print all the nodes reachable from a given starting node in a digraph using BFS method
11.	Check whether a given graph is connected or not using DFS method.
12.	Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm
13.	Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.
14.	Implement N Queen's problem using Backtracking

Course Code: **CSE266C**	Course Title: **Web Technologies Lab**	Credits: **01** L – **0** P – **2**

Course Outcomes (COs):

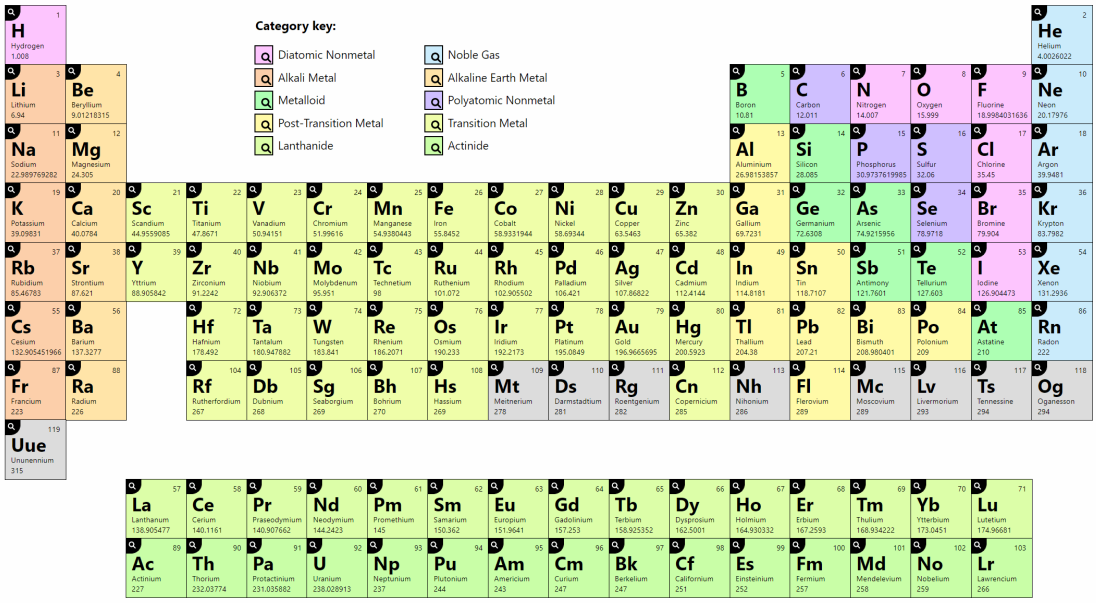
- To develop web-based application using suitable client side and server-side web technologies,
- To develop solution to complex problems using appropriate method, technologies, frameworks, web services and content management,
- To learn why server-side JavaScript is useful,
- To learn how Node.js is architected to allow high scalability with asynchronous code,
- To develop basic web applications with Node.js.

List of Experiments

1. Design an English alphabet chart such that on clicking the alphabet the appropriate example must be displayed using HTML client-side image mapping.

2. Develop and demonstrate the usage of inline, internal and external style sheet using CSS3.

3. Design the online periodic table of elements as shown below using HTML and CSS.



4. Validate the ISBN number of a given book using regular expressions in JavaScript.

5. Write JavaScript to validate the following fields of the Registration page.
   a. *First Name* (Name should contain alphabets and the length should not be less than 6 characters).
   b. *Password* (Password should not be less than 6 characters' length).
   c. *E-mail id* (should not contain any invalid and must follow the standard pattern name@domain.com)
   d. *Mobile Number* (Phone number should contain 10 digits only).
   e. *Last Name and Address* (Should not be Empty).

6. Develop an online application to find the transpose of the given matrix. Obtain the number elements from the user based on the number of rows and columns using JavaScript.

7. Build a basic CRUD application with Node.js and MongoDB.

8. Use Express.js to Create Node.js Web Apps.

Course Code: **CSE268C**	Course Title: **Project I**	Credits: **01** L – **0** P – **2**

Description:

- During fourth semester, students have to take Project-I (Mini Project) of one credit.
- Students need to identify area of their interest in which they would opt (not necessarily) their Project–II, Project–III and Project–IV in later semesters.
- Students have to do extensive literature survey of their field of interest and also learn certain tools / software required.