Semester: 5

Course: BTech

Prerequisite: Data structures, Fundamental of programming

Course Objective: Analyze the asymptotic performance of algorithms. Write rigorous correctness proofs for algorithms. Demonstrate a familiarity with major algorithms and data structures. Apply important algorithmic design paradigms and methods of analysis. Synthesize efficient algorithms in common engineering design situations.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction and Analysis of Algorithms:** Algorithm: Definition, Properties, Types of Algorithms, Writing an AlgoritAlgorithm Analysis: Parameters, Design Techniques of Algorithms Asymptotic Analysis: Big Oh, Big Omega & Big Theta Notations, Lower Bound, Upper Bound and Tight Bound, Best Case, Worst Case, Average Case Analyzing control statement, Loop invariant and the correctness of the algorithm, Recurrences- substitution method, recursion tree method, master method. Sorting Techniques with analysis: Bubble Sort, Selection Sort, Insertion sort.	20	10
2	**Divide & Conquer Algorithms:** Structure of divide-and-conquer algorithms, examples: Binary search, quick sort, Merge sort**,** Strassen Multiplication; Max-Min problem	20	6
3	**Greedy Algorithms:** Introduction, Elements of Greedy Strategy - Minimum Spanning Tree: Kruskal's & Prim's Algorithm, Dijkstra's Algorithm, Knapsack Problem, Activity Selection Problem, Huffman Codes	20	8
4	**Dynamic Programming:** Principal of Optimality, 0/1 Knapsack Problem, Making Change problem, Chain matrix multiplication, Longest Common Subsequence, All pair shortest paths: Warshall's and Floyd's algorithms	20	8
5	**Exploring Graphs:** An introduction using graphs and games, Undirected Graph, Directed Graph, Traversing Graphs, Depth First Search, Breath First Search, Topological sort	5	3
6	**Backtracking and Branch & Bound:** Introduction to Backtracking, Introduction to Branch & Bound, 0/1 Knapsack Problem, N-Queens Problem, Travelling Salesman Problem	5	4
7	**String Matching & NP Completeness:** **String Matching:** - Introduction to String Matching, Naive String Matching, Rabin-Karp Algorithm, Kruth-Morris-Pratt Algorithm, String Matching using Finite Automata **NP Completeness:** - Introduction to NP Completeness, P class Problems, NP Class Problems, Hamiltonian Cycle	10	6

Reference Books

1.	**Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill. (TextBook)**
2.	**Fundamentals of Algorithms – E. Horowitz et al. (TextBook)**
3.	**Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson**
4.	**Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.**
5.	**Algorithms—A Creative Approach,3RD Edition, UdiManber, Addison-Wesley, Reading, MA**

Course Outcome

After Learning the Course the students shall be able to:

Course Outcome: After learning the course the students will be able to:
1. Develop the ability to analyze the running time of any given algorithm using asymptotic analysis and prove the correctness of basic algorithms.
2. Design efficient algorithms for computational problems, using various algorithm design techniques taught in the course.
3. Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.
4. Analyze String matching algorithms.
5. Explain the complexity classes P, NP, and NP-Complete, and demonstrate the NP-Completeness of a specific problems.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc

Course: BTech **Semester:** 5

Prerequisite: Strong programming skills and a solid understanding of algorithms and their analysis are prerequisites for learning and applying Design and Analysis of Algorithms | 203105101 - Fundamentals of Programming

Course Objective: Design and Analysis of Algorithms (DAA) is crucial for efficient problem-solving and algorithm development. It provides tools to measure algorithm performance and make informed decisions on choosing the best algorithms for specific tasks. DAA helps optimize time and space complexities, leading to improved computational efficiency.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
0	0	4	0	2	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:

1. Develop the ability to design and implement efficient algorithms for fundamental problems.
2. Cultivate critical thinking skills to analyze problem requirements and constraints, allowing for the selection and modification of appropriate algorithms to solve specific computational problems.
3. Master the use of essential data structures such as arrays, matrices, graphs, and trees to efficiently store, manage, and manipulate data within algorithm implementations.
4. Learn techniques for optimizing algorithms to improve their efficiency and scalability, focusing on aspects such as time complexity, and space complexity,

List of Practical

1.	write a program to determine whether the given number is Prime or not.
2.	Given a sorted array and a target value,return the index if the target is found. If not,return the index where it would be if it were inserted in order.
3.	There are N children standing in a line with some rating value. You want to distribute a minimum number of candies to these children such that: Each child must have at least one candy. The children with higher ratings will have more candies than their neighbours. You need to write a program to calculate the minimum candies you must give.
4.	There is a new barn with N stalls and C cows. The stalls are located on a straight line at positions x1,xN (0 <= xi <= 1,000,000,000). We want to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. What is the largest minimum distance?
5.	Given an undirected graph with V vertices and E edges, check whether it contains any cycle or not
6.	There are n servers numbered from 0 to n – 1 connected by undirected server-to-server connections forming a network where connections[i] = [ai, bi] represents a connection between servers ai and bi. Any server can reach other servers directly or indirectly through the network.A critical connection is a connection that, if removed, will make some servers unable to reach some other servers. Return all critical connections in the network in any order.
7.	Given a grid of size NxM (N is the number of rows and M is the number of columns in the grid) consisting of '0's (Water) and '1's(Land). Find the number of islands.
8.	Given a grid of dimension N x M where each cell in the grid can have values 0, 1, or 2 which has e following meaning: 0: Empty cell 1: Cells have fres 2. Cells have oranrotten oranges We have to determine what is the minimum time required to rot all oranges. A rotten orange at index [i,j] can rot other fresh oranges at indexes [i-1,j], [i+1,j], [i,j-1], [i,j+1] (up, down, left and right) in unit time'
9.	Given two strings str1 and str2 and below operations that can be performed on str1. Find minimum number of edits (operations) required to convert 'str1' into 'str2'. Insert Remove Replace, All of the above operations are of equal cost.
10.	Minimum Path Sum" says that given a n x m grid consisting of non-negative integers and we need to find a path from top- left to bottom right, which minimizes the sum of all numbers along the path.
11.	Given string num representing a non-negative integer num, and an integer k, return the smallest possible integer after removing k digits from num.
12.	There is a robot on an m x n grid. The robot is initially located at the top-left corner (i.e., grid[0][0]).The robot tries to move to the bottom-right corner (i.e., grid[m - 1][n - 1]). The robot can only move either down or right at any point in time.Given the two integers m and n, return the number of possible unique paths that the robot can take to reach the bottom-right corner.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc

Course: BTech **Semester:** 5

Prerequisite: Calculus, Data Structures, and Algorithms

Course Objective: Formal Language & Automata Theory helps in natural language processing to solve a problem on a model of computation, using an algorithm. It enables to learn in which machine can be made to think.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction:** Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages	5	2
2	**Regular languages and finite automata:** Regular expressions and languages, deterministic finite automata -(DFA) and equivalence with regular expressions,Moore machines and mealy machines, Conversion from Mealy to Moore and vice versa, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata.	30	12
3	**Grammars:** Context-free grammars (CFG) and languages (CFL), Chomsky normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs. , Context-sensitive languages: Context-sensitive grammars (CSG) and languages.	35	15
4	**Turing machines:** The basic model for Turing machines (TM), Turing-recognizable (recursively enumerable) and Turing- decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators.	25	10
5	**Undecidability:**Church Turing thesis, universal Turing machine, the universal and diagonalization languages	5	6

Reference Books

1.	**Introduction to Automata theory, languages and Computation (TextBook)** By John E. Hopcroft, Rajiv Motwani and Jeffery D. Ullman | Pearson
2.	**Elements of the Theory of Computation** By Harry R.Lewis and Christos H. Papadimitriou | Pearson Education Asia
3.	**Introduction to the Theory of Computation** By Michael Sipser | PWS Publishing
4.	**Introduction to Languages and the Theory of Computation** By John C. Martin | McGraw Hill
5.	**Automata and Computability** By Dexter C. Kozen | Undergraduate Texts in Computer Science, Springer

Course Outcome

After Learning the Course the students shall be able to:
After Learning the course, the students shall be able to: 1. Recognize the basic concepts and applications of theory of Computation. 2. Solve Computational Problems using Regular Languages and Finite Automata. 3. Solve Computational Problems using Context free Grammar and Push Down Automata. 4. Design Turing Machine for simple computational Problems. 5. Analyze various concepts of undecidability and Computable Function.

Course: BTech **Semester:** 5

Prerequisite: Good fundamentals in calculations and ability to think logically

Course Objective: The course aims on exploring the fundamentals of Aptitude & reasoning, which involves the ability to analyze and evaluate information logically. Students will learn essential skills such as critical thinking, problem-solving, and decision-making. These skills are vital for software engineers as they navigate complex problems and make sound judgments throughout the development process.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	Number system , LCM & HCF simplifications and approximations	9	4
2	Averages , progessions,	9	4
3	Ratio and proportion,Problems on Ages, Percentages	12	5
4	Profit & loss, partnerships, S.I & C.I	12	5
5	Time & work , pipes and Cisterns, Time speed and distance , Problems on train crossings, Boats & streams ,	18	8
6	Permutations & combinations, probability	11	5
7	Directions, seating arrangements	4	2
8	Clocks, calenders	6	3
9	Cubes & Dice, syllogisms	9	4
10	Blood Relations	5	2
11	Series ,Analogy, odd man out, coding and Decoding	5	3

Reference Books

1.	**Quantitative Aptitude for CAT by Arun Sharma (TextBook)**
2.	**Logical reasoning for CAT by Arun Sharma**
3.	**Quantitative Aptitude by Abhijit Guha**

Course Outcome

After Learning the Course the students shall be able to:
1. Apply Logic & critical thinking skills to analyze information and draw logical conclusions. 2. Solve complex problems by breaking them down into manageable parts & develop effective solutions. 3. Demonstrate the ability to approach problem-solving from various perspectives.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc

Course: BTech **Semester:** 5

Prerequisite: Basic concepts of Software Flaws, Data Structures, and Mathematics including Random numbers, Number theory, and finite fields.

Course Objective: This course provides an introduction to the fundamental principles of cryptography and its applications on the network security domain as well as software development domain. This subject covers various important topics concern to information security like symmetric and asymmetric cryptography, hashing, message and user authentication, digital signatures, key distribution and overview of the malware technologies. The subject also covers the applications of all of these in real life situations

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction:** Computer Security Concept, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanism, A Model for Network Security.	5	2
2	**Classical Encryption Techniques:** Symmetric Cipher Model, Cryptanalysis, Cryptanalysis Attacks, Substitution Techniques: Caesar Cipher, Monoalphabetic Cipher, Hill Cipher, Playfair Cipher, Polyalphabetic Cipher, OTP, Transposition Techniques, Steganography	10	6
3	**Block Ciphers and the Data Encryption Standard:** Stream ciphers and block ciphers, Block Cipher Principles, Data Stream ciphers and block ciphers, Confusion & Diffusion, Block Cipher Principles, Data Encryption Standard (DES), Deferential and Linear Cryptanalysis, Avalanche Effect, strength of DES, Design principles of block cipher.	15	8
4	**Multiple Encryption and Triple DES:** Multiple encryption and triple DES, Electronic Code Book, Cipher Block Chaining Mode, Cipher Feedback mode, Output Feedback mode, Counter mode	10	4
5	**Number theory and Advance Encryption Standard:** The Euclidean Algorithm, Modular Arithmetic, Finite Fields of the Form GF(p), Polynomial Arithmetic, Advance Encryption Standard(AES): structure, key expansion	15	6
6	**Asymmetric Ciphers:** Prime Numbers, Principles of Public-Key Cryptosystems, The RSA Algorithm, Diffie Hellman Key Exchange, Man in the Middle attack	15	4
7	**Cryptographic Data Integrity Algorithms:** Hash Function: Hash Function and its Application, Security Requirements for Cryptographic Hash Functions, Hash Functions Based on Cipher BlockChaining, Secure Hash Algorithm (SHA). MAC: Message Authentication Requirements, Message Authentication Functions, Requirements for Message Authentication Codes, Security of MACs, HMAC Digital Signature: Introduction to Digital Signatures, Digital Signature standard.	20	9
8	**Key Management and Distribution:** Symmetric Key Distribution: Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Asymmetric Key Distribution: Distribution of Public Keys, X.509 certificates Advanced Topics: Firewall, Intruders, Virus, Trojans, Malware, and Ransomware.	10	6

Reference Books	
1.	**Cryptography and Network Security By William Stallings | Pearson Education (TextBook)**
2.	**Cryptography & Network Security By Behrouz A. Forouzan | Tata McGraw-Hill (TextBook)**
3.	**Information Security Principles and Practice By Deven Shah, | Wiley-India**
4.	**Information Security Principles and Practice By Mark Stamp, Willy India Edition**
5.	**Information systems security** By Nina Godbole | Wiley Publications,2008

Course Outcome
After Learning the Course the students shall be able to:
1. Explain the basic concepts of information security. 2. Compare and apply various cipher and data encryption techniques. 3. Explain the fundamental principles of AES and public key cryptosystems. 4. Illustrate use of data integrity algorithms, key management and distribution.

Course: BTech **Semester:** 5

Prerequisite: Basic concepts of Software Flaws, Data Structures, and Mathematics including Random numbers, Number theory, and finite fields

Course Objective: This course introduces the fundamental principles of cryptography and its applications in the network security domain as well as the software development domain. This subject covers various important topics concerned with information security like symmetric and asymmetric cryptography, hashing, message and user authentication, digital signatures, key distribution, and an overview of the malware technologies. The subject also covers the applications of all of these in real-lifesituations.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
0	0	2	-	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Analyze the trade-offs between security and complexity in the context of classical ciphers. 2. Apply the principles behind symmetric and asymmetric cryptography. 3. Demonstrate proficiency in hashing algorithms. 4. Apply message authentication techniques and their principles of digital signature and digital certificates. 5. Implement the various key management and remote authentication mechanisms.

List of Practical

1.	**Implement Caesar cipher encryption-decryption.**
2.	**Implement Monoalphabetic cipher encryption-decryption.**
3.	**Implement Playfair cipher encryption-decryption.**
4.	**Implement Polyalphabetic cipher encryption-decryption.**
5.	**Implement Hill cipher encryption-decryption.**
6.	**Implement Simple Transposition encryption-decryption.**
7.	**Implement One time pad encryption-decryption.**
8.	**Implement Diffi-Hellmen Key exchange Method.**
9.	**Implement RSA encryption-decryption algorithm.**
10.	**Demonstrate working of Digital Signature using Cryptool.**

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc

Course: BTech **Semester:** 5

Prerequisite: Basic knowledge of software applications.

Course Objective: This course provides a broad introduction to software engineering. The various process models required to develop software is also being described. Moreover the functional and non-functional requirements are also described.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
2	0	0	0	2	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Foundation of Enterprise Programming:** JDBC, JDBC architecture, JDBC with Oracle, MySql, Maven: integration with eclipse, POM.xml	10	3
2	**Servlets:** Basics of Web, Servlet Lifecycle, Servlets API, HTTP Servlets with XML and annotation, Servlets Configuration, Servlets Context, Servlets Collaboration, Session Tracking, CRUD operations	15	4
3	**JSP: Java Server Programming:** Scripting elements, Directive elements, CRUD operations.	15	4
4	**Hibernate (ORM):** Architecture, JPA, Generator class, Dialects, Mapping, Annotations, Transaction Management, HQL, HCQL, CRUD operations.	20	6
5	**Spring:** Architecture, Modules, Dependency Injection, Autowire, Application Context, annotation-based configuration, MVC CRUD operations	20	7
6	**Spring Boot:** Dependency Injection, Web App using spring boot, Spring boot AOP, spring boot Database, Spring Rest	20	6

Reference Books

1.	**Reference Books: Java Enterprise in a Nutshell" by Jim Farley, William Crawford, and David Flanagan (TextBook)**
2.	**Java EE 8 Design Patterns and Best Practices" by Rhuan Rocha**
3.	**Java EE and HTML5 Enterprise Application Development" by John Brock, Arun Gupta, and Geertjan Wielenga**
4.	**Java 8 Programming Black Book**

Course Outcome

After Learning the Course the students shall be able to:

1. Analyze the structure and operations of JDBC, and apply this knowledge to connect and interact with Oracle and MySQL databases.
2. Perform the concepts of Servlet Configuration and Context, and apply these in practical scenarios.
3. Apply their knowledge to perform CRUD operations using JSP and Hibernate and evaluate the results for correctness and efficiency.
4. Design and create a web application using Spring Boot.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.

Course: BTech **Semester:** 5

Prerequisite: Basic knowledge of software applications | 203105101 - Fundamentals of Programming

Course Objective: This course provides a broad introduction to software engineering. The various process models required to develop software is also being described. Moreover the functional and non-functional requirements are also described.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Prepare and do Software Requirement Specification and Software Project Management Plan by ensuring the quality of software product, different quality standards and software review techniques. 2. Apply the concept of Functional Oriented and Object Oriented Approach for Software Design. 3. Understand modern Agile Development and Service Oriented Architecture Concept of Industry 4. Analyze, design, verify, validate, implement and maintain software systems. 5. Execute a Project Management Plan, tabulate Testing Plans and Reproduce effective procedures.

List of Practical

1.	**Write a program to insert and retrieve the data from database using JDBC.**
2.	**Write a program to demonstrate the use of Prepared Statement and Result Set interface.**
3.	**Servlet Programming Servlet Execution on tomcat A servlet program to print hello world A servlet program to display request details A servlet program to handle user form A servlet program to create a cookie A servlet program to display cookie A servlet program to do session tracking Write a program to implement chat Server using Server Socket and Socket class. Write a Servlet program to send username and password using HTML forms and authenticate the user**
4.	**JSP Programming JSP program to display hello world. JSP program to demonstrate arithmetic operations JSP program to demonstrate jsp: forward action tag JSP program to request implicit object Developing a web application to insert record into Oracle Database using JSP and JDBC**
5.	**Create application to store the data in database to perform Hibernate CRUD operations.**
6.	**Create a application store the data in database to perform Spring CRUD operations.**
7.	**Create a web application to store the data in database with spring boot.**

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc

Course: BTech **Semester:** 5

Prerequisite:

Course Objective: -

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
-	1	-	-	1	100	100	-	-	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Ethics in Engineering** Scope of engineering ethics Accepting & sharing responsibility Responsible professionals and ethical corporations Resolving ethical dilemmas Case studies	20	5
2	**Group Discussion** Communication core Definition, types, process, guidelines Mock round -1	10	3
3	**Introduction to B-School Tests**Students will be able to solve verbal questions from the following exams. In these sessions students will learn to distinguish between national & international level of Management exam. GMAT CAT	15	2
4	**Listening Skills- Advanced Level**Demonstrate ability to listen more than two minutes of audio clips & solve questions based on it.	10	1
5	**Preparing Brochures**Students will learn how to establish the purpose of writing & determine audience they are writing for.	15	2
6	**Agenda & Minutes of Meeting**Students will be able to explain what an agenda & minutes of meeting are and why they are useful.	10	1
7	**Reading Comprehension; Intermediate level**Students will develop their ability to skim for main idea(s). They will able to make use of contextual clues to infer meaning of unfamiliar words from context and will be able to solve questions based on it.	10	1

Course: BTech **Semester:** 5

Prerequisite: Basic understanding of computer concepts and basic programming

Course Objective: This course provides a broad introduction to Azure cloud , infrastructure , services, security and compliance ,also billing , pricing and support plans.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
2	0	0	0	2	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Cloud Concepts:** Understanding cloud computing principles, such as the different types of cloud models (public, private, hybrid), infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS).	15	6
2	**Azure Services:** Familiarity with the various Azure services and their common use cases. This includes services like Azure Virtual Machines, Azure App Services, Azure Storage, Azure Functions, Azure SQL Database, and more	20	7
3	**Security, Privacy, Compliance, and Trust:** Knowledge of Azure security features, identity and access management, Azure Active Directory, data protection, compliance frameworks, and Azure governance methodologies.	25	5
4	**Azure Pricing and Support:** Understanding Azure subscription options, cost management, pricing models, and the different support options available to Azure customers	15	5
5	**Azure SLA and Service Lifecycles:** Familiarity with Azure Service Level Agreements (SLAs) and the Azure service lifecycle, including planned maintenance, updates, and deprecation policies.	25	7

Reference Books

1.	**Microsoft Azure Fundamentals: Understanding Azure" by Michael Collier and Robin Shahan - 3rd Edition (TextBook)**
2.	**Azure for Architects: Implementing cloud design, DevOps, containers, IoT, and serverless solutions on your public cloud" by Ritesh Modi - 2nd Edition**
3.	**Exam Ref AZ-900 Microsoft Azure Fundamentals" by Jim Cheshire - 2nd Edition**

Course Outcome

After Learning the Course the students shall be able to:

1. Describe cloud computing fundamentals, including different cloud models and service types, and become familiar with key Azure services and their typical uses.
2. Apply Azure security, privacy, compliance, and trust measures, covering identity management, data protection, compliance frameworks, and governance.
3. Apply Azure subscription management, cost optimization, pricing models, and support options for efficient utilization of Azure resources.
4. Explain Azure SLAs and service life cycles, including maintenance, updates, and deprecation policies, ensuring reliability and availability of Azure services.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc

Course: BTech **Semester:** 5

Prerequisite: Basic knowledge of Operating systems

Course Objective: This course provides a broad introduction to distributed computing

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	0	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction & Model of Distributed Computations:** What is distributed operating system, Background, need, features, Introduction to Distributed Computing	5	2
2	**Characteristics of Distributed Systems & system models:** Examples of distributed systems, Resource sharing and the web, Architectural models, fundamental model	10	4
3	**Networking and Internetworking:** Types of Networks, Network principles, Internet protocols	10	4
4	**Inter-process communication:** Introduction, External data representation and marshalling, client- server communication, group communication	10	4
5	**Distributed Objects:** Introduction, Communication between objects, Remote procedure call, events and notification	10	4
6	**Operating System support:** Introduction, OS layer, Protection, Processes and threads, communication and invocation, OS architecture	15	7
7	**Security:**Introduction, Overview of security techniques, cryptographic algorithms, digital signatures	5	3
8	**Distributed file system:** Introduction, File Service architecture, Case study: Sun network file system	10	5
9	**ransactions and Concurrency control, Distributed Transactions:**Transactions, nested transactions, Locks, Optimistic concurrency control, Flat and nested distributed transactions, atomic commit protocols, concurrency control in distributed transactions, distributed deadlocks, Transaction recovery	15	7
10	**Authentication in Distributed Systems:**Introduction, Protocols based on Symmetric cryptosystems, protocols based on asymmetric cryptosystems, Password based authentication, Authentication Protocol failures, Self-stabilization.	10	5

Reference Books

1.	**Distributed Systems concepts and Design by George coulouris, Jean Dollimore and Tim Kindberg (TextBook)**
2.	**Distributed Systems Paperback – 31 March 2017 by Coulouris George (Author), Dollimore Jean (Author), Kindberg Tim (Author), Blair Gordon (Author)**
3.	**Distributed Computing by Ajay Kshemkalyani and Mukesh Singhal**

Course Outcome

After Learning the Course the students shall be able to:
1. Explain the design principles in distributed systems and the architectures for distributed systems. 2. Apply various distributed algorithms related to clock synchronization, concurrency control, deadlock detection, load balancing, voting etc. 3. Analyze fault tolerance and recovery in distributed systems and algorithms for the same. 4. Analyze the design and functioning of existing distributed systems and file systems. 5. Implement different distributed algorithms over current distributed platforms.

Course: BTech **Semester:** 5

Prerequisite: Basic knowledge operating system

Course Objective: This course provides a broad introduction distributed computing.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
0	0	2	-	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Explain the design principles in distributed systems and the architectures for distributed systems. 2. Apply various distributed algorithms related to clock synchronization, concurrency control, deadlock detection, load balancing, voting etc. 3. Analyze fault tolerance and recovery in distributed systems and algorithms for the same. 4. Analyze the design and functioning of existing distributed systems and file systems. 5. Implement different distributed algorithms over current distributed platforms.

List of Practical

1.	**Implement concurrent echo client-server application.**
2.	**Implement concurrent day-time client-server application.**
3.	**Incrementing a counter in shared memory.**
4.	**Create CORBA based server-client application.**
5.	**Configure reliability and security options.**
6.	**Program to implement Chat Server.**
7.	**Program to implement locking algorithm.**
8.	**Program to implement Remote Procedure Call.**
9.	**Program to implement edge chasing distributed deadlock detection algorithm.**
10.	**Case Study: CORBA.**

Course: BTech **Semester:** 5

Prerequisite: Basic Electronics and Circuits, Basic Programming Language. | 203105102 - Programming for Problem Solving

Course Objective: This course provides a broad introduction to IoT and its applications. It emphasizes to practically visualizing real-world problems, analyzing them, and then designing the solution for that problem using smart components.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	0	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Basics of Networking & Basics of Network Security:** Network Types, Layered Network Models, Addressing , Internet of Things TCP/IP Transport layer, Security ,Network Confidentiality, Cryptography, Message Integrity and Authenticity, Digital signatures, Key Management, Internet, Security& Firewall.	5	3
2	**Introduction to IoT:** Genesis of IoT, IoT and Digitization, Evolutionary Phases of the Internet, IoT Impact, IoT Applications and examples: Connected Roadways, Connected Factory, Smart Connected Buildings, Smart Creatures, Convergence of IT and OT, IoT Challenges	15	4
3	**IoT Architecture -State of the Art:** Introduction, State of the art, Architecture Reference Model- Introduction, Reference Model, and architecture, IoT reference Model, IoT Reference Architecture- Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views.	15	6
4	**IoT Sensing and Actuation & IoT Processing Topologies and Types:** Introduction, Sensors, Sensor Characteristics, SensorialDeviations, SensingTypes, Sensing Considerations, Actuators, Actuators Types, Actuator Types, Actuator Characteristics, Data Formats, Processing in IoT, Processing Topologies, IoT Device Design and Selection Considerations, Processing Offloading, Offload location, Offload decision making, Offloading considerations.	20	7
5	**IoT Connectivity Technologies:** RFID , NFC, Wi-Fi, Bluetooth low energy, IEEE 802.15.4, Zigbee, Thread, Wireless HART, Z-Wave, LoRa, NB-IoT.	15	8
6	**IoT Communication Technologies:** Introduction, Constrained nodes, Constrained networks, Types of constrained devices, Low power and lossy networks, Infrastructure protocols, Internet protocol version 6 (IPv6), RPL,6LoWPAN, Content-centric networking (CCN), Discovery Protocols, Physical web, Multicast DNS (mDNS), Universal plug and play (UPnP), Data Protocols, MQTT, CoAP, AMQP, XMPP, REST, WebSocket, Identification Protocols, EPC, URIs, Device Management, Semantic Protocols, JSON-LD, Web thing model.	20	10
7	**IoT Case Studies:** Agricultural IoT, Components of an agricultural IoT, Advantages of IoT in agriculture, Case Studies, Vehicular IoT, Components of vehicular IoT, Advantages of vehicular IoT, Healthcare IoT, Components of healthcare IoT, Advantages and risk of healthcare IoT, Case Studies, Evolution of New IoT Paradigms, Challenges Associated with IoT, Emerging Pillars of IoT.	10	7

Reference Books	
1.	**Introduction to IOT (TextBook) By Sudip Mishra, Anandarup Mukherjee, Arijit Roy | Cambridge University Press (TextBook)**
2.	**IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things By David Hanes | Cisco Press**
3.	**Building the Internet of Things with IPv6 and MIPv6 The Evolving World of M2M Communications By Daniel Minoli | Wiley Publications**
4.	**A Practical Guidebook to Learn and Implement IoT using Machine Learning By Purnendu Shekhar Pandey | Perception Publications**
5.	**The Internet of Things: Connecting Objects to the Web By Hakima Chaouchi | Wiley Publications**

Course Outcome

After Learning the Course the students shall be able to:
After Learning the course the students shall be able to: 1. Understand the basics of Networking and Security.. 2. Understand architecture for Internet of Things. 3. Recognize various devices, sensors, actuators, and various processing paradigms for IoT. 4. Design a simple IoT system comprising sensors, wireless network connection, data analytics and display/actuators, and write the necessary control software. 5. Learn working of IoT Connectivity/Medium access protocols

Course: BTech **Semester:** 5

Prerequisite: Basic Electronics and Circuits, Basic Programming Language. | 203105101 - Fundamentals of Programming

Course Objective: This course provides a broad introduction to IoT and its applications. It emphasizes to practically visualizing real- world problems, analyzing them, and then designing the solution for that problem using smart components.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
0	0	2	-	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** – Practical

Course Outcome

After Learning the Course the students shall be able to:
After Learning the course the students shall be able to: 1. Understand the basics of Networking and Security.. 2. Understand architecture for Internet of Things. 3. Recognize various devices, sensors, actuators, and various processing paradigms for IoT. 4. Design a simple IoT system comprising sensors, wireless network connection, data analytics and display/actuators, and write the necessary control software. 5. Learn working of IoT Connectivity/Medium access protocols

List of Practical

1.	**Introduction to Arduino programming.**
2.	**Introduction to Arduino Uno R3**
3.	**To blink the LED with Arduino.**
4.	**To interface push button with Arduino.**
5.	**To interface LCD with Arduino.**
6.	**To read the analog voltage using ADC on Arduino.**
7.	**To detect occupancy of an area using PIR sensors**
8.	**To interface real time clock IC DS1307 with Arduino.**
9.	**To measure the distance of an object using ultrasonic sensor**
10.	**To display temperature and humidity data.**

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.

Subject Code:- 303105306
Subject Name:- Theory of Computation

Sr. No	Lecture No.	Unit No.	TOPIC	Lecture Hours
1	1	1	Sets, Functions, Logical statements, Proofs, Relations,	1
2	2		Languages, Principal of Mathematical Induction, Strong Principle,	1
3	3		Recursive Definitions, Structural Induction, Alphabet,	1
4	4		languages and grammar,	1
5	5		productions and derivation,	1
6	6		Chomsky hierarchy of languages.	1
7	7	2	Regular expressions and languages,	2
8	8		deterministic finite automata -(DFA) and equivalence with regular expressions,	1
9	9		Moore machines and Mealy machines,	1
10	10		Conversion from Mealy to Moore and vice versa,	1
10	10		nondeterministic finite automata (NFA) and equivalence with DFA,	1
11	11		regular grammar and equivalence with finite automata	1
12	12		properties of regular languages,	1
13	13		pumping lemma for regular languages,	1
14	14		minimization of finite automata.	1
15	15	3	Context-free grammars (CFG) and languages (CFL),	2
16	16		Chomsky normal forms,	1
17	17		nondeterministic pushdown automata (PDA) and equivalence with CFG,	2
18	18		parse trees, ambiguity in CFG,	1
19	19		pumping lemma for context-free languages,	2
20	20		deterministic pushdown automata	2
21	21		closure properties of CFLs.	1
22	22		Context-sensitive languages: Context-sensitive grammar (CSG) and languages.	2
23	23	4	The basic model for Turing machines (TM),	2
24	24		Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties,	2
25	25		variants of Turing machines,	2
26	26		nondeterministic TMs and equivalence with deterministic TMs,	1
27	27		unrestricted grammar and equivalence with Turing machines,	2
28	28		TMs as enumerators.	1
29	29	5	Church Turing thesis,	2
30	30		universal Turing machine,	2
31	31		the universal and diagonalization languages	2

Subject Code: 303105218
Subject Name: Design and Analysis of Algorithm

Sr. No	Lecture No.	Unit No.	TOPIC	Lecture Hours
1	1		Algorithm: Definition, Properties	1
2	2		Data analytics and its types, Why data analytics is important,	1
3	3		Types of Algorithms	1
4	4		Writing an AlgoritAlgorithm Analysis: Parameters	1
5	5		Design Techniques of Algorithms Asymptotic Analysis: Big Oh, Big Omega & Big Theta Notations, Lower Bound, Upper Bound and Tight Bound, Best Case, Worst Case, Average Case	1
6	6		Analyzing control statement	1
7	7		Loop invariant and the correctness of the algorithm	1
8	8		Recurrences- substitution method	1
9	9		recursion tree method,master method	1
10	10	1	Sorting Techniques with analysis: Bubble Sort, Selection Sort, Insertion sort.	1
11	11		Structure of divide-and-conquer algorithms	1
12	12		examples: Binary search,	1
13	13		quick sort	1
14	14		Merge sort	1
15	15	2	Max-Min problem	1
16	16		Introduction	1
17	17		Elements of Greedy Strategy	1
18	18		Minimum Spanning Tree	1
19	19	3	Kruskal's & Prim's Algorithm,	1

20	20		Dijkstra's Algorithm,	1
21	21		Knapsack Problem	1
22	22		Activity Selection Problem	1
23	23		Huffman Codes	1
24	24		Principal of Optimality,	1
25	25		0/1 Knapsack Problem,	2
26	26		Making Change problem,	1
27	27		Chain matrix multiplication,	1
28	28		Longest Common Subsequence,	1
29	29		All pair shortest paths: Warshall's algorithms	1
30	30	4	Floyd's algorithms	1
31	31		An introduction using graphs and games, Undirected Graph, Directed Graph,	1
32	32		Traversing Graphs, Depth First Search,	1
33	33	5	Breath First Search, Topological sort	1
34	34		Introduction to Backtracking,	1
35	35		Introduction to Branch & Bound,	1
36	36		0/1 Knapsack Problem,	1
37	37	6	N-Queens Problem, Travelling Salesman Problem	1
38	38		Introduction to String Matching, Naive String Matching, Rabin-Karp Algorithm,	1
39	39		Kruth-Morris-Pratt Algorithm, String Matching using Finite Automata	1
40	40		Introduction to NP Completeness, P class Problems,	1
41	41	7	NP Class Problems, Hamiltonian Cycle	1

Subject Code: 303105219
Subject Name: Design and Analysis of Algorithm Laboratory

Sr. No	Practical No.	Practical Title	Lab Hours
1	1	write a program to determine whether the given number is Prime or not.	2
2	2	Given a sorted array and a target value,return the index if the target is found. If not,return the index where it would be if it were inserted in order.	2
3	3	There are N children standing in a line with some rating value. You want to distribute a minimum number of candies to these children such that: Each child must have at least one candy. The children with higher ratings will have more candies than their neighbours. You need to write a program to calculate the minimum candies you must give.	2
4	4	There is a new barn with N stalls and C cows. The stalls are located on a straight line at positions x1,xN (0 <= xi <= 1,000,000,000). We want to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. What is the largest minimum distance?	2
5	5	Given an undirected graph with V vertices and E edges, check whether it contains any cycle or not	2
6	6	There are n servers numbered from 0 to n – 1 connected by undirected server-to-server connections forming a network where connections[i] = [ai, bi] represents a connection between servers ai and bi. Any server can reach other servers directly or indirectly through the network.A critical connection is a connection that, if removed, will make some servers unable to reach some other servers. Return all critical connections in the network in any order.	2
7	7	Given a grid of size NxM (N is the number of rows and M is the number of columns in the grid) consisting of '0's (Water) and '1's(Land). Find the number of islands.	2
8	8	Given a grid of dimension N x M where each cell in the grid can have values 0, 1, or 2 which has e following meaning: 0: Empty cell 1: Cells have fres 2. Cells have oranrotten oranges We have to determine what is the minimum time required to rot all oranges. A rotten orange at index [i,j] can rot other fresh oranges at indexes [i-1,j], [i+1,j], [i,j-1], [i,j+1] (up, down, left and right) in unit time'	2

9	9	Given two strings str1 and str2 and below operations that can be performed on str1. Find minimum number of edits (operations) required to convert ‘str1’ into ‘str2’. Insert Remove Replace, All of the above operations are of equal cost.	2
10	10	Minimum Path Sum” says that given a n x m grid consisting of non-negative integers and we need to find a path from top- left to bottom right, which minimizes the sum of all numbers along the path.	2
11	11	Given string num representing a non-negative integer num, and an integer k, return the smallest possible integer after removing k digits from num.	2
12	12	There is a robot on an m x n grid. The robot is initially located at the top-left corner (i.e., grid[0][0]).The robot tries to move to the bottom-right corner (i.e., grid[m - 1][n - 1]). The robot can only move either down or right at any point in time.Given the two integers m and n, return the number of possible unique paths that the robot can take to reach the bottom-right corner.	2

Subject Code: 303105253
Subject Name: Software Engineering

Sr. No	Lecture No.	Unit No.	TOPIC	Lecture Hours
1	1	1	Study of Different Models, Software Characteristics, Components, Applications, Layered Technologies	1
2	2		Processes, Methods and Tools, Generic View Of Software Engineering	1
3	3		Process Models - Waterfall model, Incremental	1
4	4		Evolutionary process models- Prototype, Spiral and Concurrent Development Model	1
5	5		Agile Development: Agility and Agile Process model	1
6	6		Extreme Programming, Other process models of Agile Development and Tools	1
7	7	2	Management Spectrum, People Product Process- Project	1
8	8		W5HH Principle, Importance of Team Management	1
9	9		Planning a Software Project: Scope and Feasibility, Effort Estimation, Schedule and staffing	1
10	10		Quality Planning, Risk management- identification	1
11	11		Assessment, control, project monitoring plan, Detailed Scheduling	1
12	12	3	Problem Recognition, Requirement Engineering tasks	1
13	13		Processes, Requirements Specification	1
14	14		Use cases and Functional specification - 1	1
15	15		Use cases and Functional specification - 2	1
16	16		Requirements validation, Requirements Analysis	1
17	17	4	Design Concepts, Design Model, Software Architecture, Data Design, Architectural Styles and Patterns	1
18	18		Architectural Design, Alternative architectural designs	1
19	19		Modelling Component level design and its modelling, Procedural Design, Object Oriented Design	1
20	20		Data Oriented Analysis & Design: Difference between Data and Information, E-R Diagram	1
21	21		Dataflow Model, Control Flow Model	1
22	22		Control and Process Specification, Data Dictionary	1
23	23	5	Programming principles and guidelines, Programming practices, Coding standards	1
24	24		Incremental development of code, Management of code evaluation, Unit testing- procedural units, classes	1
25	25		Code Inspection, Metrics- size measure, complexity metrics, Cyclomatic Complexity	1
26	26		Halstead measure, Knot Count, Comparison Of Different Metrics	1
27	27	6	Concepts, Psychology of testing, Levels of testing, Testing Process- test plan, test case design	1
28	28		Execution, Black-Box testing Boundary value analysis	1
29	29		Pair wise testing- state based testing	1
30	30		White-Box testing criteria and test case generation and tool support	1
31	31		Quality Assurance: Quality Control, Assurance, Cost	1

32	32		Reviews, Software Quality Assurance	1
33	33	7	Computer Aided Software Engineering Tools	1
34	34		SCRUM Developments	1
35	35		Dependable System, Reliability Engineering	1
36	36		Safety Engineering, Security Engineering	1
37	37		Resilience Engineering	1
38	38	8	Software Reuse, Component Based Software Engineering	1
39	39		Distributed Software Engineering	1
40	40		Service-Oriented Software Engineering	1
41	41		Real-Time Software Engineering	1
42	42		Systems Engineering, Systems of System	1

Subject Code: 303105253

Subject Name: Software Engineering Laboratory

Sr. No	Practical No.	Practical Title	Lab Hours
1	1	Project Definition and objective of the specified module and Perform Requirement Engineering Process.	2
2	2	Identify Suitable Design and Implementation model from the different software engineering models.	2
3	3	Prepare Software Requirement Specification (SRS) for the selected module.	2
4	4	Develop Software project management planning (SPMP) for the specified module.	2
5	5	Do Cost and Effort Estimation using different Software Cost Estimation models.	2
6	6	Prepare System Analysis and System Design of identified Requirement specification using structure design as DFD with data dictionary and Structure chart for the specific module.	2
7	7	Designing the module using Object Oriented approach including Use case Diagram with scenarios, Class Diagram and State Diagram, Collaboration Diagram, Sequence Diagram and Activity Diagram.	2
8	8	Defining Coding Standards and walk through.	2
9	9	Write the test cases for the identified module.	2
10	10	Demonstrate the use of different Testing Tools with comparison.	2
11	11	Define security and quality aspects of the identified module.	2

Subject Code: 303105309
Subject Name: Enterprise Programming Using Java

Sr. No	Lecture No.	Unit No.	TOPIC	Lecture Hours
1	1	1	Introduction to Foundation of Enterprise Programming	1
2	2		XML	1
3	3		JDBC	1
4	4		JDBC architecture	1
5	5		JDBC with Oracle	1
6	6		MySql in detail	1
7	7		MySql with programming	1
8	8		Maven: integration with eclipse	1
9	9		Maven: integration with eclipse	1
10	10		POM.xml	1
11	11		POM.xml in detail	1
12	12	2	Introduction to servlets	1
13	13		Basics of Web	1
14	14		Servlet Lifecycle	1
15	15		Servlets API	1
16	16		HTTP Servlets with XML and annotation	1
17	17		Servlets Configuration	1
18	18		Servlets Context	1
19	19		Servlets Collaboration	1
20	20		Session Tracking, CRUD operations	1
21	21		servlets CRUD operations	1
22	22	3	Introduction to JSP: Java Server Programming	1
23	23		Scripting elements	1
24	24		Directive elements	1
25	25		JSP CRUD operations	1
26	26	4	Introduction to Hibernate (ORM)	1
27	27		Architecture	1
28	28		JPA	1
29	29		Generator class	1
30	30		Dialects	1
31	31		Mapping	1
32	32		Annotations	1
33	33		Transaction Management	1
34	34		HQL	1
35	35		HCQL	1
36	36		Hibernate CRUD operations	1
37	37	5	Introduction to Spring	1
38	38		Architecture	1
39	39		Modules	1
40	40		Dependency Injection	1
41	41		Autowire	1
42	42		Application Context	1
43	43		Annotation-based configuration	1
44	44		MVC CRUD operations	1
45	45	6	Introduction to Spring Boot	1

46	46		Dependency Injection	1
47	47		Web App using spring boot	1
48	48		Spring boot AOP	1
49	49		spring boot Database, Spring Rest	1
50	50		Introduction of Microservices Architecture with Spring Boot and Docker	1
51	51		Spring Security for authentication and authorization in enterprise applications	1

Subject Code: 303105310
Subject Name: Enterprise Programming Using Java

Sr. No	Practical No.	Practical Title	Lab Hours
1	1	Write a program to insert and retrieve the data from database using JDBC.	2
2	2	Write a program to demonstrate the use of Prepared Statement and Result Set interface.	2
3	3	Servlet Programming Servlet Execution on tomcat A servlet program to print hello world A servlet program to display request details A servlet program to handle user form A servlet program to create a cookie A servlet program to display cookie A servlet program to do session tracking Write a program to implement chat Server using Server Socket and Socket class.	2
4	4	Write a program to implement chat Server using Server Socket and Socket class.	2
5	5	Write a Servlet program to send username and password using HTML forms and authenticate the user	2
6	6	Write a program to implement chat Server using Server Socket and Socket class.	2
7	7	JSP program to display hello world. JSP program to demonstrate arithmetic operations.	2
8	8	JSP program to demonstrate jsp: forward action tag JSP program to request implicit object.	2
9	9	Developing a web application to insert records into Oracle Database using JSP and JDBC.	2
10	10	Create an application store the data in database to perform Spring CRUD operations.	2
11	11	Create a web application to store data in database with spring boot.	2
12	12	Develop an application by Using Enterprise programming code (Any).	2

Subject Code: 303193304
Subject Name: Professionalism & Corporate Ethics

Sr. No	Lecture No.	Unit No.	TOPIC	Lecture Hours
1	1	**1-Ethics in Engineering**	Scope of engineering ethics	1
2	2		Accepting & sharing responsibility	1
3	3		Responsible professionals and ethical corporations	1
4	4		Resolving ethical dilemmas	1
5	5		Case studies	1
6	6	**2-Group Discussion**	Communication core	1
7	7		Definition, types, process, guidelines	1
8	8		Mock round -1	1
9	9	**3-Introduction to B-School Tests**	GMAT	1
10	10		CAT	1
11	11	**4-Listening Skills- Advanced Level**	Listening Skills- Advanced Level	1
12	12	**5-Preparing Brochures**	Students will learn how to establish the purpose of writing	1
13	13		Determine the audience they are writing for	1
14	14	**6-Agenda & Minutes of Meeting**	Agenda & Minutes of Meeting.	1
15	15	**7-Reading Comprehension; Intermediate level**	Reading Comprehension: Intermediate level	1

Subject Code: 303105377
Subject Name: IOT Based System

Sr. No	Lecture No.	Unit No.	TOPIC	Lecture Hours
1	1	1	**Network Types, Layered Network Models, Addressing , Internet of Things TCP/IP Transport layer,**	1
2	2		**Security, Network Confidentiality, Cryptography, Message Integrity and Authenticity, Digital Signatures,**	1
3	3		**Key Management, Internet,Security & Firewall.**	1
4	4	2	**Genesis of IoT, IoT and Digitization, Evolutionary Phases of the Internet,**	1
5	5		**IoT Impact, IoT Applications and examples: Connected Roadways, Connected Factory,**	2
6	6		**Smart Connected Buildings, Smart Creatures, Convergence of IT and OT, IoT Challenges**	1
7	7		**Introduction, State of the art,**	1

8	8	3	**Architecture Reference Model- Introduction, Reference Model, and Architecture,**	1

9	9		**IoT Reference Model,**	1
10	10		**IoT Reference Architecture- Introduction,**	1
11	11		**Functional View, Information View, Deployment**	1
12	12		**Operational View, Other Relevant architectural views.**	1
13	13	4	**Introduction, Sensors, Sensor Characteristics,**	1
14	14		**Sensorial Deviations, Sensing Types, Sensing Considerations,**	1
15	15		**Actuators, Actuators Types, Actuator Characteristics,**	1
16	16		**Data Formats, Processing in IoT, Processing Topologies,**	1
17	17		**IoT Device Design and Selection Considerations,**	1
18	18		**Processing Offloading, Offload location,**	1
19	19		**Offload Decision Making,, Offloading considerations.**	1
20	20	5	**RFID ,**	1
21	21		**NFC, Wi-Fi,**	2
22	22		**Bluetooth low Energy,**	1
23	23		**IEEE 802.15.4,**	2
24	24		**Zigbee, Thread, Wireless HART,**	1
25	25		**Z-Wave, LoRa, NB-IoT.**	1
26	26	6	**Introduction, Constrained Nodes, Constrained Networks,**	1
27	27		**Types of constrained Devices, Low Power and lossy Networks,**	2
28	28		**Infrastructure protocols, Internet protocol version 6 (IPv6), RPL,6LoWPAN,**	1
29	29		**Content-Centric Networking (CCN), Discovery Protocols, Physical web,**	1
30	30		**Multicast DNS (mDNS), Universal plug and play (UPnP),**	1
31	31		**Data Protocols,MQTT, CoAP,**	1
32	32		**AMQP, XMPP, REST, WebSocket,**	1
33	33		**Identification Protocols, EPC, URIs, Device Management,**	1
34	34		**Semantic Protocol, JSON-LD, Web Thing Model.**	1
35	35	7	**Agricultural IoT, Components of an Agricultural IoT,**	1
36	36		**Advantages of IoT in agriculture, Case Studies,**	1
37	37		**Vehicular IoT,Components of vehicular IoT, Advantages of vehicular IoT,**	1
38	38		**Healthcare IoT, Components of healthcare IoT,**	1
39	39		**Advantages and risk of healthcare IoT, Case Studies,**	1
40	40		**Evolution of New IoT Paradigms, Challenges Associated with IoT, Emerging Pillars of IoT.**	2

Subject Code: 303105319
Subject Name: IoT Based System Laboratory

Sr. No	Practical No.	Practical Title	Lab Hours

1	1	Introduction to Arduino Programming.	2
2	2	Introduction to Arduino Uno R3.	2
3	3	To blink the LED with Arduino.	2

4	4	To interface push button with Arduino.	2
5	5	To interface LCD with Arduino.	2
6	6	To read the analog voltage using ADC on Arduino.	2
7	7	To detect occupancy of an area using PIR sensors.	2
8	8	To interface real time clock IC DS1307 with Arduino.	2
9	9	To measure the distance of an object using ultrasonic sensor.	2
10	10	To display temperature and humidity data.	2