Course: BTech **Semester:** 5

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

Subject Syllabus

303105219 - Design and Analysis of Algorithm Laboratory

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:** 4

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	
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:** Study of Different Models, Software Characteristics, Components, Applications, Layered Technologies, Processes, Methods and Tools, Generic View Of Software Engineering, Process Models- Waterfall model, Incremental, Evolutionary process models- Prototype, Spiral And Concurrent Development Model Agile Development : Agility and Agile Process model, Extreme Programming, Other process models of Agile Development and Tools.	10	6
2	**Software Project Management:** Management Spectrum, People ‘Product ‘Process- Project, W5HH Principle, Importance of Team Management Planning a Software Project : Scope and Feasibility, Effort Estimation, Schedule and staffing, Quality Planning, Risk management- identification, assessment, control, project monitoring plan, Detailed Scheduling	10	5
3	**Requirements Engineering:** Problem Recognition, Requirement Engineering tasks, Processes, Requirements Specification, Use cases and Functional specification, Requirements validation, Requirements Analysis	10	5
4	**Structured System Design:** Design Concepts, Design Model, Software Architecture, Data Design, Architectural Styles and Patterns, Architectural Design, Alternative architectural designs, Modeling Component level design and its modeling, Procedural Design, Object Oriented Design. Data Oriented Analysis & Design : Difference between Data and Information, E-R Diagram, Dataflow Model, Control Flow Model, Control and Process Specification, Data Dictionary	15	5
5	**Coding and Unit Testing:** Programming principles and guidelines, Programming practices, Coding standards, Incremental development of code, Management of code evaluation, Unit testing- procedural units, classes, Code Inspection, Metrics- size measure, complexity metrics, Cyclomatic Complexity, Halstead measure,Knot Count, Comparison Of Different Metrics	10	4
6	**Software Testing and Quality Assurance:** Concepts, Psychology of testing, Levels of testing, Testing Process- test plan, test case design, Execution, Black-Box testing ‘Boundary value analysis ‘Pair wise testing- state based testing, White-Box testing criteria and test case generation and tool support Quality Assurance : Quality Control, Assurance, Cost, Reviews, Software Quality Assurance, Approaches to SQA, Reliability, Quality Standards- ISO9000 And 9001	15	7
7	**CASE Tools and Advance Practices of System Dependability and Security:** Computer Aided Software Engineering Tools, SCRUM Developments, Dependable System, Reliability Engineering, Safety Engineering, Security Engineering, Resilience Engineeirng	15	5

8	**Advance Software Engineering:** Software Reuse, Component Based Software Engineering, Distributed Software Engineering, Service-Oriented Software Engineering, Real-Time Software Engineering, Systems Engineering, Systems of System.	**15**	**5**

Reference Books

1.	**Software Engineering (TextBook) R.Pressmen; 6th (TextBook)**
2.	**Software Engineering** By Sommerville
3.	**Fundamentals of Software Engineering** By Rajib Mall | PHI
4.	**Software Engineering** By Pankaj Jalote | Wiley India

Course Outcome

After Learning the Course the students shall be able to:

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

Course: BTech **Semester:** 4

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	
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:

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

List of Practical

1.	**Project Definition and objective of the specified module and Perform Requirement Engineering Process.**
2.	**Identify Suitable Design and Implementation model from the different software engineering models.**
3.	**Prepare Software Requirement Specification (SRS) for the selected module.**
4.	**Develop Software project management planning (SPMP) for the specified module.**
5.	**Do Cost and Effort Estimation using different Software Cost Estimation models.**
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.**
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.**
8.	**Defining Coding Standards and walk through.**
9.	**Write the test cases for the identified module.**
10.	**Demonstrate the use of different Testing Tools with comparison.**
11.	**Define security and quality aspects of the identified module.**

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.

Subject Syllabus

303105307 - Artificial Intelligence

Course: BTech **Semester:** 5

Prerequisite: Data structure, Formal Languages and automata Theory, Mathematics

Course Objective: This course provides a broad introduction to Artificial Intelligence. AI techniques for search and knowledge representation also Apply knowledge of AI planning and machine learning techniques to real-world problems.

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:** Definition of an AI, Major Areas of Artificial Intelligence, AI Techniques, History, AI problems, Production Systems, Problem characteristics, Intelligent Agents, Agent Architecture, AI Application (E-Commerce, & Medicine), AI Representation, Properties of internal representation, Future scope of AI, Issues in the design of search algorithms. Introduction to AI Problems and Applications, Defining Problems as a State Space Search, Problem Characteristics, Production Systems.	15	7
2	**Search techniques:** Generate-And-Test, Hill Climbing, Best-First Search, Problem Reduction, Constraint Satisfaction, Means-Ends Analysis. Heuristic search, Hill Climbing, Best first search, mean and end analysis, Constraint Satisfaction, A* and AO* Algorithm, Knowledge Representation: Basic concepts, Knowledge representation Paradigms, Propositional Logic, Inference Rules in Propositional Logic, Knowledge representation using Predicate logic, Predicate Calculus, Predicate and arguments, ISA hierarchy, Frame notation, Resolution, Natural Deduction	20	8
3	**Knowledge Representation:** Knowledge Representation – Representation and Mappings, Different Approaches, Issues in knowledge representation. Predicate Logic - Representation Simple Facts in Logic, Representing Instance and Isa Relationships, Computable Functions and Predicates, Resolution. Propositional Logic: Representation, Inference, Reasoning Patterns, Resolution, First-order Logic: Representation, Inference, Reasoning Patterns, Resolution	15	8
4	**Uncertainty:** Non-Monotonic Reasoning, Logics for Non-Monotonic Reasoning, Forward rules, and Backward rules, Justification based Truth Maintenance Systems, Semantic Nets Statistical Reasoning, Probability and Bayes’ theorem, Bayesian Network, Markov Networks, Hidden Markov Model, Basis of Utility Theory, Utility Functions.	15	4
5	**Fuzzy Sets and Fuzzy Logic:** Fuzzy Set Operations, Membership Functions, Fuzzy Logic, Hedges, Fuzzy Proposition and Inference Rules, Fuzzy Systems.	10	5
6	**Natural Language Processing:** Introduction, Syntactic Processing, Semantic Analysis, Semantic Analysis, Discourse and Pragmatic Processing, Spell Checking.	10	5
7	**Neural Networks and Expert systems:** Introduction to neural networks and perception-qualitative Analysis, Neural net architecture and applications, Utilization and functionality, the architecture of the expert system, knowledge representation, two case studies on expert systems	15	8

Reference Books

1.	**Artificial Intelligence: A New Synthesis, Harcourt Publishers (TextBook)** By N. J. Nilsson | Harcourt Publishers
2.	**Artificial Intelligence (TextBook)** By Elaine Rich and Kevin Knight | TMH
3.	**Artificial Intelligence-Structures and Strategies For Complex Problem Solving** By George F. Luger | Pearson Education / PHI
4.	**Artificial Intelligence-A Modern Approach** By Stewart Russell and Peter Norvig | Pearson Education/ Prentice Hall of India | 2
5.	**Artificial Intelligence – A Practical Approach** By Patterson | Tata McGraw Hill | 3

Course Outcome

After Learning the Course the students shall be able to:

1. Discuss AI fundamentals, history, and future trends to develop solutions for problem-solving, inference, perception, knowledge representation, and learning tasks.
2. Utilize knowledge representation methods like propositional logic, predicate logic, and frame notation to effectively represent knowledge within AI systems.
3. Discover methods for solving AI problems, including diverse search algorithms and techniques like non-monotonic reasoning, probability theory, Bayesian networks, and fuzzy logic for effective decision-making in uncertain scenarios.
4. Apply Natural Language Processing (NLP), Neural Networks and Expert Systems technologies effectively in real-world scenarios.

Miscellaneous

Exam Requirement

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

Course: BTech **Semester:** 5

Prerequisite: Data structure, automata, and languages, Mathematics

Course Objective: This course provides a broad introduction to Artificial Intelligence. AI techniques for search and knowledge representation also Apply knowledge of AI planning and machine learning techniques to real-world problems.

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. Analyze real-world problems and apply appropriate AI techniques to solve them.
2. Create AI systems using heuristic search and knowledge representation techniques.
3. Implement core AI algorithms to solve problems and understand their functionalities.
4. Apply programming skills to build functional AI applications.
5. Analyze complex data and design neural network architectures for pattern recognition and problem-solving.

List of Practical

1.

Develop an AI-based medical diagnosis system using expert systems architecture and knowledge representation techniques.

2.

Build an intelligent agent for optimizing e-commerce inventory management using search algorithms like hill climbing and best-first search.

3.

Implement a constraint satisfaction algorithm to solve scheduling problems in healthcare facilities

4.

Create a recommendation system for personalized learning using means-end analysis and heuristic search techniques.

5.

Develop a problem-solving agent for optimizing resource allocation in logistics using A* and AO* algorithms.

6.

Develop a fuzzy logic-based system for predicting stock market trends considering uncertain market conditions.

7.

Write a program to implement BFS (Water Jug problem or any AI search problem).

Write a program to implement DFS (Water Jug problem or any AI search problem).

8.

Define a predicate brother(X,Y) which holds iff X and Y are brothers.

Define a predicate cousin(X,Y) which holds iff X and Y are cousins.

	Define a predicate grandson(X,Y) which holds iff X is a grandson of Y. Define a predicate descendent(X,Y) which holds iff X is a descendent of Y. Consider the following genealogical tree: father(a,b). father(a,c). father(b,d). father(b,e). father(c,f). Say which answers, and in which order, are generated by your definitions for the following queries in Prolog: ?- brother(X,Y). ?- cousin(X,Y). ?- grandson(X,Y). ?- descendent(X,Y).
9.	Write a program to implement Tic-Tac-Toe game using python.
10.	Create a spell-checking application utilizing natural language processing (NLP) techniques, including syntactic and semantic analysis.
11.	Design a neural network architecture for pattern recognition in medical imaging for disease diagnosis.

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: 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	**UNIT-1**Number system , LCM & HCF simplifications and approximations	9	4
2	**UNIT-2**Averages , progessions,	9	4
3	**UNIT-3**Ratio and proportion,Problems on Ages, Percentages	12	5
4	**UNIT-4**Profit & loss, partnerships, S.I & C.I	12	5
5	**UNIT-5** Time & work , pipes and Cisterns, Time speed and distance , Problems on train crossings, Boats & streams ,	18	8
6	**UNIT-6**Permutations & combinations, probability	11	5
7	**UNIT-7**Directions, seating arrangements	4	2
8	**UNIT-8**Clocks, calenders	6	3
9	**UNIT-9**Cubes & Dice, syllogisms	9	4
10	**UNIT-10**Blood Relations	5	2
11	**UNIT-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: Data analytics and Data analysis, Data visualization techniques and Statistical measures, Basics of Programing Languages, Understanding of Python.

Course Objective: Data Analytics helps small and large organizations maximize the value of their data, unearth insights, build plans and respond in real-time to customer demand.

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 to Data Analytics:** Introduction, Data and its importance, Data analytics and its types, Why data analytics is important, Data analysis Vs Data analytics, Classification of data analytics, Elements of Data analytics, Data analyst Vs. Data scientist	25	9
2	**Introduction to Python Fundamentals and Statistics:** Introduction, Importance of Python, Levels of Data measurement, Central tendency and Dispersion, Distribution of Sample Means, Population and Variance, Confidence interval estimation	15	8
3	**Probability and Types of Testing:**Probability and Probability distribution, Sampling and Sampling distribution, Hypothesis testing, Anova test, Chi-square test	20	9
4	**Regression, Classification and Clustering:** Linear and Logistic regression, Clustering: K-Means clustering and Hierarchical clustering, Classification: Decision tree, Confusion matrix	25	10
5	**Data Visualization Using PowerBI:** Introduction to visualization and analytic tool: Power BI, Getting Data from different sources, data transformations, introduction to data modeling, types of data visualizations in PowerBI, Publishing and sharing reports, Use cases of Dashboard and Analytical Reports Creation.	15	9

Reference Books

1.	**Data Analytics using Python By Bharati Motwani, Wiley Publications. (TextBook)**
2.	**Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data.(TextBook) Wiley Publications**
3.	**Statistics 101: From Data Analysis and Predictive Modeling to Measuring Distribution and Determining Probability, Your Essential Guide to Statistics By David Borman, Adams Media**
4.	**Machine Learning, A Probabilistic Approach. By Kevin P. Murphy**

Course Outcome
After Learning the Course the students shall be able to:
1. Explain basics of data analytics lifecycle and visualization. 2. Compare different analytics techniques and visualization using Python. 3. Apply various testing methods and techniques using probability 4. Apply different regression, classification, clustering techniques. 5. Create an interactive data visualization using PowerBI.

Course: BTech **Semester:** 5

Prerequisite: Data analytics tools like PowerBI, Different techniques of visualization and data analytics | 203105251 - Database Management System

Course Objective: Data Analytics helps small and large organizations to maximize the value of their data, unearth insights, build plans and respond in real-time to customer demand.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	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. Apply statistical measures to calculate mean, median and mode.
2. Compare and apply different regression, classification algorithm on the given dataset .
3. Perform clustering and detect outliers
4. Create an interactive data visualization dashboard using PowerBI.

List of Practical

1.	**Perform Exploratory Data Analysis on the given dataset using Python.**
2.	**Calculate mean, median and mode of the first 50 records in the given dataset using python.**
3.	**Perform Multiple Linear Regression on data.**
4.	**Perform the Logistic Regression on a dataset.**
5.	**Use a dataset & apply K means clustering to get insights from data.**
6.	**Perform the Decision tree classification algorithm using a dataset.**
7.	**Study and installation of the tools like PowerBI tool for data Visualization.**
8.	**Load a dataset from different sources in PowerBI and apply transformations to it.**
9.	**Study and Plot various graphs for Data Visualization on PowerBI.**
10.	**Given a case study: Interactive Data Analytics with Power BI Dashboard.**

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
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	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: Fundamentals of web applications, Understanding of PHP

Course Objective: Web application security is the practice of protecting websites, applications, and APIs from attacks. It is a broad discipline, but its ultimate aims are keeping web applications functioning smoothly and protecting business from cyber vandalism, data theft, unethical competition, and other negative consequences.

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 HTTP & HTTPs:** HTTP Request, Response - Header Fields and HTTPS - Understanding Same Origin – Cookies – Sessions - Web Application Proxies, Understanding Burp-Suite.	20	6
2	**Information Gathering:**whois, nsLookup, netcraft - web server fingerprinting - subdomain enumeration - fingerprinting frameworks - hidden resource enumeration - security misconfigurations - google hacking database - Shodan HQ. OSINT Framework, NMAP: Scanning.	20	9
3	**SQL Injections & Authentication Vulnerabilities:** SQL Statements, Finding SQL Injections, Exploiting SQL Injections, Bypass Authentication, Xpath Injection, Error Based Injection, Double Query Injection, Time Based injections, Union Based Injections, SQL Map, Mitigation plans.	20	10
4	**Advance Web Application Attacks:** Anatomy of an XSS Exploitation, Reflected XSS, Persistent XSS, DOM based XSS, Browsers and XSS, Blocking malicious request, user enumeration, random password guessing, remember me functionality, no limit attempts, password reset feature, logout flaws, CAPTCHA.	20	10
5	**Advance Web Application Attacks-2:** Security Misconfiguration, Sensitive data exposure, Insecure direct object reference and security, CSRF (Cross Site Request Forgery), HTTP Response Splitting, Using Components With Known Vulnerabilities, Unvalidated Redirects and Forwards	20	10

Reference Books

1.	**The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws Dafydd Stuttard, Marcus Pinto (TextBook)**
2.	**The Tangled Web: A Guide to Securing Modern Web Applications" by Michal Zalewski**
3.	**Web Application Security, A Beginner's Guide" by Bryan Sullivan and Vincent Liu**
4.	**OWASP Testing Guide" by The Open Web Application Security Project (OWASP)**
5.	**Web Hacking 101" by Peter Yaworski**

Course Outcome
After Learning the Course the students shall be able to:
1. Describe the potential security implications of decentralized technologies in Web application. 2. Identify potential attack vectors through information gathering methods. 3. Differentiate between white-box, grey-box, and black-box penetration testing methodologies. 4. Evaluate the effectiveness of identified vulnerabilities based on the OWASP Top 10 web application security risks. 5. Demonstrate the key phases of a Secure Development Life Cycle (SDLC) and their role in security.

Course: BTech **Semester:** 5

Prerequisite: Basic knowledge of operating systems, Social Networking platforms, Types of web application functionality | 203105215 - Computer Networks

Course Objective: Learning web application security will give insights into the various types of cyber threats, compliance requirements, career options in like security analyst, penetration tester, security consultant, and security engineer.

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:

After Learning the course, the students shall be able to:
1. Identify common web application vulnerabilities like SQL injection, XSS, and CSRF and also learn to utilize security testing tools and manual testing methods.
2. Implement secure coding principles in their code to prevent vulnerabilities.
3. Create a threat model for a web application, identifying potential threats and attack vectors and analyze case studies of web application security breaches along with its legal implications
4. Develop mitigation strategies to address identified security risks and propose solutions to improve overall web application security.

List of Practical

1.	**Cross-site scripting (XSS) attacks: This practical could involve testing a web application for XSS vulnerabilities and demonstrating how an attacker can exploit them.**
2.	**SQL injection attacks: Students can be given hands-on experience in exploiting SQL injection vulnerabilities to access or modify sensitive data in a web application.**
3.	**CSRF (Cross-Site Request Forgery) attacks: This practical could involve demonstrating how an attacker can use CSRF vulnerabilities to trick a user into performing an unwanted action on a web application.**
4.	**Broken authentication and session management: Students can be trained to identify and exploit vulnerabilities in authentication and session management mechanisms in a web application.**
5.	**Web application firewall (WAF) evasion techniques: This practical could involve testing a web application firewall and demonstrating how an attacker can bypass it using different techniques.**
6.	**Information leakage and sensitive data exposure: Students can be given hands-on experience in identifying and exploiting vulnerabilities that expose sensitive data or information.**
7.	**File inclusion attacks: This practical could involve demonstrating how an attacker can exploit file inclusion vulnerabilities to execute arbitrary code on a web server**
8.	**Clickjacking attacks: Students can be trained to identify and exploit clickjacking vulnerabilities in a web application to trick users into clicking on malicious links.**
9.	**Security configuration issues: This practical could involve identifying and exploiting vulnerabilities resulting from insecure web application configurations.**
10.	**Input validation and sanitization: Students can be given hands-on experience in testing the input validation and sanitization mechanisms of a web application and identifying vulnerabilities.**

Miscellaneous

Exam Requirement

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

Course: BTech **Semester:** 5

Prerequisite: Fundamentals of Android and iOS architecture Mobile rooting and Jailbreaking, Understanding of IPA and APK

Course Objective: The objective of this subject is to train the students about various types of pen testing methodology for mobile devices, basic concepts of penetration testing of mobile applications.

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	**Fundamentals of Android OS and Applications:**History of Android, Understanding Android Hardware and Software Architecture, Understanding Android Security Model	20	6
2	**IOS & IPA Architecture:** History of iOS, Understanding iOS Hardware and Software Architecture, Understanding iOS Security Model, Understanding iOS Permission Model for Application Security, Sandboxing, Jailbreaking Devices, Understanding IPA	20	9
3	**Mobile App Security:** Understanding Android Permission Model for Application Security, Sandboxing, Codesigning, Encryption, rooting Devices, Understanding APK Understanding Directories and Files on an APK	20	10
4	**Setting up Mobile Vulnerabilities System and Devices:** Setting up Mobile App Pen testing Environment, interact with the Devices, Starting with Drozer, Understanding AndroidManifest.xml,Configuring, Burp and Traffic Interception, Traffic Interception Bypass	20	10
5	**Mobile Application Attacks:** Weak Server-Side Controls (M1), Insecure Data Storage (M2), Insufficient Transport Layer Protection (M3), Unintended Data Leakage (M4), Poor Authentication & Authorization (M5), Broken Cryptography (M6), Client-Side Injections (M7), Security Decisions via Untrusted Input (M8), Improper Session Handling (M9), Lack of Binary Protection (M10)	20	10

Reference Books

1.	**"iOS Application Security: The Definitive Guide for Hackers and Developers" by David Thiel (TextBook)**
2.	**Android Security Internals: An In-Depth Guide to Android's Security Architecture" by Nikolay**
3.	**The Mobile Application Hacker's Handbook" by Dominic Chell, Tyrone Erasmus, Shaun Colley, Ollie Whitehouse, and Georg Wicherski**
4.	**Mobile Application Security: Protecting Mobile Devices and Their Applications" by Manoranjan (Mano) Paul**

Course Outcome

After Learning the Course the students shall be able to:

1. Describe the core components of the Android hardware and software architecture.
2. Evaluate the security mechanisms of iOS, including its data protection model, sandboxing's impact on app security, and the potential risks of device jailbreaking.
3. Apply the Android permission model to configure app access and identify security risks, code-signing verifies app authenticity and origin.
4. Evaluate mobile vulnerability assessment tools and preparing secure testing environments, automate repetitive tasks and streamline the mobile security testing process.
5. Analyze the common vulnerabilities exploited in mobile application attacks, identify different attack types and implement effective mitigation strategies to protect your devices.

Miscellaneous

Exam Requirement

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

Course: BTech **Semester:** 5

Prerequisite: Fundamentals of Android and iOS architecture Mobile rooting and Jailbreaking Understanding of IPA and APK. | 203105251 - Database Management System

Course Objective: The objective of this subject is to train the students about various types of pen testing methodology for mobile devices, basic concepts of penetration testing of mobile applications.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	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. Debug and troubleshoot common issues encountered during mobile app development.
2. Implement user authentication and authorization mechanisms securely.
3. Analyze the security vulnerabilities specific to each type of mobile application
4. Conduct penetration testing exercises to exploit identified vulnerabilities and assess their impact.
5. Develop a mobile application security policy based on the OWASP Mobile Top 10 guidelines.

List of Practical

1.	**Study the architecture of Android and APK using dex2jar command line.**
2.	**Perform APK reversing using JADX.**
3.	**Perform IPA reversing.**
4.	**Setting up burp suite to intercept mobile application traffic.**
5.	**Setting up MobSF and extract the source code of the apk.**
6.	**Install Genymotion/NOX player and configure it with the ADB to analyze the apk.**
7.	**Installing DIVA on the virtual platform to perform OWASP TOP 10 mobile vulnerabilities.**
8.	**Perform Client-side injection on the apk.**
9.	**Demonstrate the Hard-coded issue in the apk file.**
10.	**Demonstrate improper session handling in apk.**

Course: BTech **Semester:** 5

Prerequisite: Basic knowledge of system and mobile devices, Social Networking platforms, Types of web application functionality, Operating System, Computer Ports and services, Solid understanding of networking fundamentals, Familiarity with operating systems (Linux and Windows)

Course Objective: The objective of this subject is to train the students about various types of pentesting methodology, basic concepts of red teaming and use of Metasploit for penetration testing.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	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	**Meterpreter-2** Setting up multiple communication channels with the target, Meterpreter anti-forensics, the get-desktop and keystroke sniffing, Meterpreter resource scripts, Meterpreter timeout control, Meterpreter Sleep Control, Meterpreter transports, Interacting with the registry, Meterpreter API and mixins, Injecting VNC server remotely, Enabling remote Desktop	20	9
2	**Server Side Exploitation :** Exploiting a Linux server, Exploiting a Windows machine, Exploiting Common services	30	12
3	**Client Side Exploitation :** Bypassing antivirus and IDS/IPS, Human interface device attacks, HTA attack, Backdooring executables using a MITM attack, Creating a Linux trojan, File format based Exploitation-PDF and Word, Creating an Android backdoor	20	12
4	**Wireless Network penetration Testing:** Metasploit and wireless, understanding an evil twin attack, Configuring karmetasploit, Wireless MITM attacks, SMB relay attacks	30	12

Reference Books

1.	**Metasploit: The Penetration Tester's Guide David Kennedy, Jim O'Gorman, Devon Kearns, Mati Aharoni (TextBook)**
2.	**Hacking: The Art of Exploitation Jon Erickson**
3.	**Network Security Essentials William Stallings**
4.	**Metasploit Penetration Testing Cookbook Packt Publishing**
5.	**Metasploit Revealed - Secrets of the Expert Pentester - Build your Defense against Complex Attacks Packt Publishing**

Course Outcome
After Learning the Course the students shall be able to:
1. Explain the difference between penetration testing and vulnerability assessments. 2. Analyzing the structure and anatomy of Metasploit, including an in-depth exploration of its core components. 3. Utilize Metasploit to conduct client-side attacks, generate payloads with msfvenom, and exploit Windows machines using social engineering techniques. 4. Assess the effectiveness of various post-exploitation modules in Linux environments to gather comprehensive system information. 5. Apply post-exploitation modules for Windows, including capture, gather, and manage functionalities. In addition, gain a foundational understanding of cryptography and its various types.

Course: BTech **Semester:** 5

Prerequisite: Understanding the basic concepts of the Linux operating system, Navigating the Linux file system and directory structure, File and directory permissions in Linux, Operating System, Experience using the Linux terminal for executing commands, Critical thinking and problem-solving skills for addressing practical challenges

Course Objective: The objective of this subject is to train the students about various types of pentesting methodology, basic concepts of red teaming and use of Metasploit for penetration testing.

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:

At the end of the course, you will be able to:
1. List the different types of manual penetration testing methodologies.
2. Compare and contrast Red Teaming with other penetration testing methodologies.
3. Apply Identify vulnerabilities in a real-world environment using manual testing techniques.
4. Differentiate between symmetric and asymmetric encryption techniques.
5. Develop custom malicious files to exploit specific vulnerabilities in a target system.

List of Practical

1.	**Meterpreter anti-forensics**
2.	**The getdesktop and keystroke sniffing**
3.	**Interacting with the windows registry**
4.	**Meterpreter API and mixins**
5.	**Injecting VNC server remotely**
6.	**Enabling remote Desktop**
7.	**Exploiting a Linux server**
8.	**Exploiting a Windows machine**
9.	**Exploiting Common Network services**
10.	**Bypassing antivirus and IDS/IPS**

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