BH23 B.Tech. CSE Syllabus	B. Tech. II Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS401PC	**Software Engineering**	**3**	**0**	**0**	**3**

Course Description: This course discusses principles of software engineering, process models and software requirements. Also explores design principles, testing strategies and risks in software development.

Course Outcomes:
After the end of the course, the student will be able to

C401.1	Illustrate software process framework and models for the development of software application.
C401.2	Analyze and validate the requirement engineering strategy for developing software requirement specification documents.
C401.3	Choose an appropriate model to create an architectural design.
C401.4	Apply various testing strategies to verify the software quality.
C401.5	Illustrate the importance of framework for product metrics.
C401.6	Identify the risk strategy and QA techniques for developing quality software.

UNIT – I	**Introduction to Software Engineering:**

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths.

A Generic view of process: Software engineering - a layered technology, a process framework, the capability maturity model integration (CMMI).
Process models: The waterfall model, Spiral model and Agile methodology.

UNIT – II	**Requirement Engineering**

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.
Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

UNIT - III	**Design Engineering**

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT – IV	**Testing**

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.
Metrics for Process and Products: Software measurement, metrics for software quality.

UNIT – V	Risk Management
Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM. **Quality Management**: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.	

TEXT BOOKS:

1. Roger S. Pressman, Software Engineering, A Practitioner's Approach, Sixth Edition, McGraw Hill International Edition.
2. Ian Sommerville, Software Engineering, Seventh Edition, Pearson Education.

REFERENCE BOOKS:

1. Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Addison Wesley.
2. James F. Peters, Witold Pedrycz, Software Engineering, An Engineering approach, John Wiley.
3. Waman S Jawadekar, Software Engineering Principles and Practice, The McGraw Hill Education.
4. Meiler Page-Jones, Fundamentals of Object-Oriented Design using UML, Pearson Education.

BH23 B.Tech. CSE Syllabus	B. Tech. II Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS403PC	**Operating Systems (Common to CSE & IT)**	3	0	0	3
Pre-requisite	**1. Programming for Problem Solving** **2. Elements of Computer Science and Engineering**				

Course Description: This course covers concepts viz., CPU Scheduling, Process Management, Synchronization, Virtual Memory, Memory Allocation Methods, File Management etc.,

Course Outcomes:
After the end of the course, the student will be able to

C403.1	Understand basic concepts of System Structures, Process and Threads.
C403.2	Evaluate CPU scheduling algorithms and deadlock handling mechanisms.
C403.3	Apply various mechanisms to achieve synchronization.
C403.4	Identify suitable mechanism for Inter Process Communication.
C403.5	Choose appropriate Memory Management techniques.
C403.6	Implement efficient File Management techniques through System Calls.

Unit – I	**Introduction**

Introduction: System Structures, Operating System Services, User OS Interface, System Calls and Types, System Programs.
Process: Process Concept, Process Scheduling, Operations on Processes.
Multithreaded Programming: Overview, Multithreading Models.

Unit – II	**CPU Scheduling**

CPU Scheduling: Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling. System Call Interface for Process Management-fork, exit, wait, waitpid, exec
Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, DeadlockPrevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock

Unit – III	**Process management and Synchronization**

Process Management and Synchronization: The Critical Section Problem, Synchronization, Hardware, Semaphores, Classical Problems of Synchronization, Monitors.
Inter Process Communication Mechanisms: IPC using Pipes, FIFOs, Message Queues, Shared Memory.

Unit – IV	**Memory Management and Virtual Memory**

Memory Management: Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation.
Virtual Memory: Demand Paging, Page Replacement, Page Replacement Algorithms.

Unit – V	File System Interface and operations
File System Interface: Access Methods, Directory Structure, Protection, File System Structure, Allocation Methods, Free Space Management. **File Operations:** Usage of open, create, read, write, close, lseek, stat, ioctl system calls	

TEXT BOOKS:

1. Abraham Silberchatz, Peter B. Galvin, Greg Gagne, Operating System Principles - Sixth Edition, John Wiley.
2. W. Richard Stevens, Stephen A. Rago, Advanced Programming in the UNIX Environment, Pearson Education.

REFERENCE BOOKS:

1. William Stallings, Operating Systems - Internals and Design Principles, Fifth Edition-2005, Pearson Education/PHI.
2. Crowley, Operating System A Design Approach, TMH.
3. Andrew S. Tanenbaum, Modern Operating Systems, 2nd edition, Pearson/PHI
4. Kernighan and Pike, UNIX programming environment, PHI/ Pearson Education
5. U. Vahalia, UNIX Internals - The New Frontiers, Pearson Education
6. Andrea Arpaci-Dusseau and Remzi Arpaci-Dusseau, Operating Systems: Three Easy Pieces.

BH23 B.Tech. CSE Syllabus	B. Tech. II Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS405PC	**Computer Organization and Architecture**	3	0	0	3
Pre - Requisite	**1. Elements of Computer Science and Engineering** **2. Digital Electronics**				

Course Description The course introduces computer design, organization, and architectural concepts, usage of register transfer language, instruction sets. It also consists of hardware algorithms on diverse topics such as computer arithmetic, memory organization, IO organization, Parallel processing and Vector processing enabling students to understand fundamental computing principles effectively

Course Outcomes:
At the end of the course, the student will be able to

C405.1	Apply the knowledge of computer design, organization, and architectural concepts to implement Micro-operations.
C405.2	Design a suitable Control unit for a decided set of Instructions.
C405.3	Design Hardware and Algorithms for manipulation of data, represented in different formats.
C405.4	Implement data transfer with appropriate IO Interface and Interrupt mechanism.
C405.5	Choose suitable type of Memory for a given purpose.
C405.6	Perform Parallel Processing using suitable mechanism.
Unit – I	**Introduction**

Introduction: Definition of Computer Organization, Computer Design and Computer Architecture and types of architecture- Von-Neumann Architecture, Harvard Architecture.
Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.
Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt

Unit – II	**Microprogrammed Control, Central Processing Unit and Reduced Instruction Set Computer**

Microprogrammed Control: Control memory, Address sequencing, micro program example, design of control unit.
Central Processing Unit: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.
Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.

Unit – III	Data Representation and Computer Arithmetic
Data Representation: Fixed Point Representation, Floating Point Representation. **Computer Arithmetic:** Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – Point Arithmetic operations. Decimal Arithmetic unit, Decimal Arithmetic operations.	
Unit – IV	**Input-Output Organization and Memory Organization**
Input-Output Organization: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access. **Memory Organization:** Main Memory, Associate Memory, Cache Memory.	
Unit – V	**Pipeline and Vector Processing and Multi Processors**
Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor. **Multi Processors:** Characteristics of Multiprocessors, Interconnection Structures, Interprocessor arbitration, Interprocessor communication and synchronization, Cache Coherence.	

TEXT BOOK:

1. M. Morris Mano, Computer System Architecture, Third Edition, Pearson/PHI.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer Organization, 5th Edition, McGraw Hill Education.

REFERENCE BOOKS:

1. William Stallings, Computer Organization and Architecture, Sixth Edition, Pearson/PHI.
2. Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition, PHI/Pearson.
3. Computer Organization and Design, The Hardware /Software Interface, David A. Patterson, John L. Hennessy, 4th Edition.

BH23 B.Tech. CSE Syllabus	B. Tech. II Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS406PC	**Operating Systems Lab (Common to CSE & IT)**	**0**	**0**	**2**	**1**

Course Description: This lab compliments the Operating Systems syllabus. Students will be able to simulate and implement Operating systems Concepts such as CPU Scheduling, Process Management through System Calls, Deadlock Management, Inter Process Communication, Semaphores, MemoryManagement, File Management, etc.,

Course Outcomes:

At the end of the course, the student will be able to

C406.1	Evaluate CPU Scheduling Algorithms and Memory management techniques.
C406.2	Construct deadlock detection and avoidance algorithms.
C406.3	Solve classical problems of synchronization using semaphores.
C406.4	Evaluate Inter process communication mechanisms.

LIST OF PROGRAMS

CYCLE 1:

1. Write a program to simulate the following CPU Scheduling algorithms.
   a) FCFS b) SJF c) Round Robin d) priority
2. a) Write a program to implement Process management system calls viz., fork, exit, wait, waitpid, exec.
   b) Write a program to implement I/O system calls viz., open, read, write, close, seek, stat, opendir, readdir.

CYCLE 2:

3. Write a program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.
4. Write a program to implement the Producer – Consumer problem using semaphores using UNIX/LINUX system calls.
5. Write a program to illustrate the following IPC mechanisms
   a) Pipes b) FIFOs c) Message Queues d) Shared Memory
6. Write a program to simulate the following memory management techniques
   a) Paging b) Segmentation
7. Write a program to simulate Contiguous Memory Allocation techniques
   a) First-Fit b) Best-Fit c) Worst-fit
8. Write a program to stimulate Page Replacement Algorithms
   a) FCFS b) LRU c) Optimal

TEXT BOOKS:

1. Abraham Silberchatz, Peter B. Galvin, Greg Gagne, Operating System Principles - Sixth Edition, John Wiley.
2. W. Richard Stevens, Stephen A. Rago, Advanced Programming in the UNIX Environment, Pearson Education.

REFERENCE BOOKS:

1. William Stallings, Operating Systems – Internals and Design Principles, Fifth Edition– 2005, Pearson Education/PHI.
2. Crowley, Operating System - A Design Approach TMH.
3. Andrew S Tanenbaum, Modern Operating Systems, 2nd edition, Pearson/PHI
4. Kernighan and Pike, UNIX Programming Environment, PHI/Pearson Education
5. U. Vahalia, UNIX Internals: The New Frontiers, Pearson Education

BH23 B.Tech. CSE Syllabus	B. TECH II Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS407PC	**Node JS/ React JS/ Django (Common to CSE & IT)**	**0**	**0**	**2**	**1**
Prerequisite	Object Oriented Programming through Java				

Course Description: This course used to develop dynamic web applications using different frameworks and deploy them.

Course Outcomes:

After the end of the course, the student will be able to

C407.1	Build a custom website with HTML, CSS, and Bootstrap and little JavaScript.
C407.2	Demonstrate Advanced features of JavaScript and learn about JDBC.
C407.3	Develop Server – side implementation using Java technologies.
C407.4	Develop the server – side implementation using Node JS.

EXERCISES:

CYCLE - 1

WEEK - 1

1. Build a responsive web application for E-Book management system with registration, login, catalog and cart pages using CSS3 features, flex and grid. Description of application given in Ebook system of Software Engineering.
2. Make the above web application responsive web application using Bootstrap framework for E-ticketing system. Description of application given in E-ticketing of Software Engineering.

WEEK - 2

3. Use JavaScript for doing client – side validation of the pages implemented in experiment 1 and experiment 2.
4. Explore the features of ES6 like arrow functions, callbacks, promises, async/await. Implement an application for reading the weather information from openweathermap.org and display the information in the form of a graph on the web page.

WEEK - 3

5. Develop a java standalone application that connects with the database (Oracle / mySql) and perform the CRUD operation on the database tables.

WEEK - 4

6. Create an xml for the bookstore. Validate the same using both DTD and XSD. Description of application given in Book Bank of Software Engineering.
7. Design a controller with servlet that provides the interaction with application developed in experiment 1 and the database created in experiment 5.

CYCLE-2

WEEK-5

8. Maintaining the transactional history of any user is very important. Explore the various session tracking mechanism (Cookies, HTTP Session)

WEEK-6

9. Create a custom server using http module and explore the other modules of Node JS like OS, path, event.
10. Develop an express web application that can interact with REST API to perform CRUD operations on student data. (Use Postman)

WEEK-7

11. For the above application create authorized end points using JWT (JSON Web Token).
12. Create a react application for the student management system having registration, login, contact, about pages and implement routing to navigate through these pages.

WEEK-8

13. Create a service in react that fetches the weather information from openweathermap.org and the display the current and historical weather information using graphical representation using chart.js
14. Create a TODO application in react with necessary components and deploy it into Github.

REFERENCE BOOKS:

1. Jon Duckett, Beginning HTML, XHTML, CSS, and JavaScript, Wrox Publications, 2010
2. Bryan Basham, Kathy Sierra and Bert Bates, Head First Servlets and JSP, O'Reilly Media, 2nd Edition, 2008.
3. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, A Press.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS501PC	**Design and Analysis of Algorithms (Common to CSE & CSE(AIML))**	3	1	0	4
Pre - Requisite	**Computer Programming and Data Structures**				

Course Description: This course covers performance of algorithms, algorithmic design paradigms, modelling of problems using disjoint sets, priority queues and graphs, classification of problems into P & NP classes.

Course Outcomes:

At the end of the course, the student will be able to

C501.1	Analyze the performance of the algorithms and represent using relevant notations.
C501.2	Apply the concepts of disjoint sets and priority queues to solve real world problems.
C501.3	Choose appropriate algorithmic design paradigms to solve various real world problems.
C501.4	Identify the issues in graph connectivity and resolve them.
C501.5	Reduce the search space of a problem using bounding functions.
C501.6	Categorize problems into NP hard & NP Complete.

Unit – I **Introduction Algorithms and Sets**

Introduction: Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notations**,** Amortized Analysis.

Disjoint Sets: Disjoint set operations, union and find algorithms, Priority Queue - Heaps, Heapsort.

Unit – II **Divide and Conquer & Greedy Method**

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication.

Greedy method: General method, applications-Job sequencing with deadlines, knapsack problem, Single source shortest path problem.

Unit – III **Dynamic Programming**

Dynamic Programming: General method, Multistage Graph problem, Applications- Optimal binary search tree, 0/1 knapsack problem, All pairs shortest path problem, Traveling salesperson problem, Reliability design.

Unit – IV **Traversals** & **Backtracking**

Basic Traversal and Search Techniques: Techniques for Binary Trees, Techniques for Graphs, Connected components, Biconnected components.

Backtracking: General method, applications, n-queen's problem, sum of subsets problem, graph Coloring, hamitonian cycles.

Unit – V	Branch and Bound & P and NP Problems
Branch and Bound: General method, applications - Traveling salesperson problem, 0/1 knapsack problem - LC Branch and Bound solution, FIFO Branch and Bound solution. **NP-Hard and NP-Complete problems:** Basic concepts, non-deterministic algorithms, NP-Hard and NP-Complete classes, Cook's theorem.	

TEXT BOOK:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharan, University press, 1998.

REFERENCE BOOKS:

1. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson education.
2. Introduction to Algorithms, Third edition, T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein, PHI Pvt. Ltd./ Pearson Education.
3. Algorithm Design: Foundations, Analysis and Internet Examples, M.T. Goodrich and R. Tamassia, John Wiley and sons.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS502PC	**Computer Networks** **(Common to CSE & CSE(AIML))**	3	0	0	3
Pre - Requisites	**1. Elements of Computer Science and Engineering** **2. Data Structures**				

Course Description:

This course covers concepts viz., basic taxonomy and terminology of the computer networking and enumerates the layers of OSI model and TCP/IP model. Design issues and Protocols related to Data link layer, Network layer, Transport layer and Application layer.

Course Outcomes:

At the end of the course, the student will be able to

C502.1	Analyze pros and cons of the components, reference models and various transmission media.
C502.2	Analyze various link control and access control mechanisms available in the data link layer.
C502.3	Grasp the foundational principles, challenges, and mechanisms of the network layer in computer networks.
C502.4	Choose the appropriate routing algorithm suitable for the given network topology
C502.5	Manage the networks to ensure efficient, reliable, and high-quality communication.
C502.6	Assess the Transport layer protocols and the features of Application layer.

Unit – I **Introduction to Networking Concepts**

Network hardware, Network software, Reference Models, Example Networks: Internet, Wireless LANs: 802.11x.

Physical Layer: Guided Transmission media: twisted pairs, coaxial cable and fiber optics. Wireless Transmission: Electromagnetic Spectrum, Radio Transmission, Microwave Transmission and Infrared Transmission.

Data link layer: Design issues, Error detection and correction.

Unit – II **Data Link Layer & Medium Access Control Sub layer**

Elementary data link protocols: A Simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channels.

Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocol: SONET

Medium Access Control Sublayer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, Collision free protocols, Ethernet; Data link layer switching.

Unit – III	Network Layer -I
Network Layer Design Issues, Network Layer in the Internet: The IPv4 Protocol, IP Addresses, IPv6. **Routing algorithms:** Shortest Path Algorithm, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing, Broadcast, Multicast.	
Unit – IV	**Network Layer –II**
Congestion Control Algorithms, Quality of Service, Internetworking, BGP.	
Unit – V	**Transport Layer & Application Layer**
Transport Layer: Transport Services, Elements of Transport protocols, The Internet Transport protocols: TCP and UDP protocols. **Application Layer:** The Domain Name System, Electronic Mail, The WORLD WIDE WEB, Streaming Audio and Video.	

TEXT BOOK:

1. Andrew S Tanenbaum, David. j. Wetherall, Computer Networks, 5th Edition, Pearson Education/PHI

REFERENCE BOOKS:

1. S. Keshav, An Engineering Approach to Computer Networks, 2nd Edition, Pearson
2. Behrouz A. Forouzan, Data Communications and Networking, Third Edition TMH.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS503PC	**DevOps**	3	0	0	3
Pre - Requisite	**Software Engineering**				

Course Description: This course contains an advanced Software development model - DevOps. Work with Project Management using Git server and Docker. Integrating the system with Jenkins and finally testing using Selenium.

Course Outcomes:

At the end of the course, the student will be able to

C503.1	Explore the various components of the DevOps environment.
C503.2	Identify Software development models and architectures of DevOps
C503.3	Work with Source code management.
C503.4	Choose a project management tool.
C503.5	Use the Jenkins integration tool to build the application.
C503.6	Choose appropriate testing tools deployment model for the project

Unit – I **Introduction to DevOps**

Introducing DevOps, Agile development model, DevOps and ITIL. DevOps process and Continuous Delivery.

Release management, Scrum, Kanban and delivery pipeline, Identifying bottlenecks.

Unit – II **Software development models and DevOps**

DevOps Lifecycle for Business Agility, DevOps and Continuous Testing.

DevOps influence on Architecture: Introducing software architecture, The monolithic scenario, Handling database migrations, Micro services and the data tier.

Unit – III **Introduction to project management**

The need for source code control, The history of source code management, Roles and code, source code management system and migrations, shared authentication.

Hosted Git servers, Different Git server implementations, Docker intermission, Gerrit, The pull request model, Git Lab.

Unit – IV **Integrating the system**

Build systems, Jenkins build server, Managing build dependencies, Jenkins plugins and file system layout, The host server, Build slaves, Triggers.

Job chaining and build pipelines, Build servers and infrastructure as code, Building by dependency order, Build phases, Alternative build servers, Collating quality measures.

Unit – V	Testing Tools and Deployment
Various types of testing, Automation of testing Pros and cons. Karate testing - Introduction, Karate testing features, Spring Boot Testing Pitfalls, Tips and Tricks. **Deployment of the system**: Deployment systems, Virtualization stacks, Puppet master and agents, Ansible, Deployment tools: Chef, SaltStack and Docker.	

TEXT BOOK:

1. Joakim Verona., Practical DevOps, Packt Publishing, 2016.
2. Benjamin Bischoff, Writing API Tests with Karate: Enhance your API testing for improved security and performance, Kindle Edition.

REFERENCE BOOKS:

1. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint. Wiley publications.
2. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley.
3. Testing Spring Boot Applications Demystified, Philip Riecks.

BH23 B.Tech. CSE Syllabus	B. Tech. II Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS511PE	**Graph Theory** **(Professional Elective – I)**	3	0	0	3
Pre - Requisite	**Discrete Mathematics**				

Course Description: This course deals with some basic concepts and properties in graph theory. This course also covers the concepts of graph connectivity and colouring of graphs.

Course Outcomes:
At the end of the course student will be able to

C511.1	Perform various operations in different digraphs.
C511.2	Apply the algorithms to find the shortest paths in connected graphs
C511.3	Analyze the classification of Trees based on properties and derive spanning trees.
C511.4	Describe the different types of graphs.
C511.5	Formulate and prove theorems about trees and graphs.
C511.6	Solve problems on vertex colourings.

Unit – I	**Introduction**

Introduction: Definitions, Degree of a vertex, Directed walks, paths and cycles, Connectivity in digraphs, Eulerian and Hamilton digraphs, Eulerian digraphs, Hamilton digraphs, Union, Sum, Cartesian Product, Composition.

Unit – II	**Connected graphs and shortest paths**

Walks, trails, paths, cycles, Connected graphs, Cut-vertices and cut-edges, Blocks, Connectivity, Weighted graphs and shortest paths, Menger's theorem, Dijkstra's algorithm, Floyd Algorithm.

Unit – III	**Trees and Graphs**

Definition, Characterization, and Simple Properties, minimum spanning trees, Counting the Number of Spanning Trees, Kruskal's Algorithm, Prim's Algorithm, Cayley's Formula, The Connector Problem.
Special classes of graphs - Bipartite Graphs, Line Graphs, Chordal Graphs, Eulerian Graphs, Chinese Postman problem.

Unit – IV	**Independent sets coverings and matchings**

Introduction: Vertex-Independent sets and Vertex coverings, Matching and factors, Matchings in bipartite graphs, Ramsey's Theorem, Perfect Matchings and the Tutte Matrix.

Unit – V	Vertex Colorings
Basic definitions, Cliques and Chromatic number, Applications of Graph Coloring, Brooks Theorem, B-Colorings, Edge Colorings of graphs, Gupta-Vizing's Theorem. A scheduling problem and equitable edge - coloring.	

TEXT BOOKS:

1. J. A. Bondy and U. S. R. Murty. Graph Theory, volume 244 of Graduate Texts in Mathematics. Springer, 1st edition, 2008.
2. J. A. Bondy and U. S. R. Murty. Graph Theory with Applications.
3. R. Balakrishnan and K. Ranganathan A Textbook of Graph Theory, Second edition.

REFERENCE BOOKS:

1. Lecture Videos: http://nptel.ac.in/courses/111106050/13.
2. Introduction to Graph Theory, Douglas B. West, Pearson.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year ISem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS512PE	**Distributed Databases (Professional Elective – I)**	3	0	0	3
Pre - Requisite	**Database Management Systems**				

Course Description: The course enriches the previous knowledge of database systems and exposes the need for distributed database technology to overcome the deficiencies of the centralized database systems.

Course Outcomes:
At the end of the course, the student will be able to

C512.1	Analyze the architecture and design of distributed database systems.
C512.2	Explore the objectives and algorithms for distributed query processing.
C512.3	Apply the mechanisms for concurrency control and deadlock management.
C512.4	Evaluate the measures for distributed systems reliability and fault tolerance.
C512.5	Choose the appropriate parallel database system architecture for implementation
C512.6	Implement distributed object database management and data management systems.

Unit – I **Distributed DBMS Architecture and Design**

Introduction; Distributed Data Processing, Distributed Database System, Promises of DDBSs, Complications and Design Issues.
Distributed DBMS Architecture: Architectural Models for Distributed DBMS, Autonomy, Distribution, Heterogeneity, Architectural Alternatives, Client/Server Systems, Peer-to-Peer Systems, Multi database System Architecture
Distributed Database Design: Top-Down Design Process, Distribution Design issues, Fragmentation, Allocation.

Unit – II **Query Processing and Optimization**

Query processing and decomposition: Query processing objectives, characterization of query processors, layers of query processing, query decomposition, localization of distributed data.
Distributed query Optimization: Query optimization, centralized query optimization, Join Ordering in Distributed Queries, distributed query optimization algorithms.

Unit – III **Transaction Management and Concurrency Control**

Transaction Management: Definition, properties of transaction, types of transactions.
Distributed concurrency control: serializability, concurrency control mechanisms & algorithms, time - stamped &optimistic concurrency control Algorithms, deadlock Management.

Unit – IV	Reliability and Parallel Database Systems
Distributed DBMS Reliability: Reliability concepts and measures, failures in Distributed DBMS, local & distributed reliability protocols, site failures and network partitioning. **Parallel Database Systems**: Parallel database system architectures, parallel data placement, parallel query processing, load balancing, database clusters.	
Unit – V	**Distributed Object Database Management Systems and Data Management**
Distributed Object Database Management Systems: Fundamental object concepts and models, object distributed design, architectural issues, object management, distributed object storage, object query Processing. **Data Management**: Data Stream Management, Cloud Data Management	

TEXT BOOKS:

1. M. Tamer OZSU and Patuck Valduriez: Principles of Distributed Database Systems, Pearson Edn. Asia, 2001.
2. Stefano Ceri and Giuseppe Pelagatti: Distributed Databases, McGraw Hill.

REFERENCE BOOKS:

1. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom: "Database Systems: The Complete Book", Second Edition, Pearson International Edition.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS513PE	**Data Mining (Professional Elective – I)**	3	0	0	3
Pre - Requisite	**Knowledge of probability and statistics**				

Course Description: This course covers the concepts such as discovering patterns, correlations, and anomalies, trends within large datasets to extract useful information and make informed decisions. This course also introduces the techniques, and algorithms used in data mining.

Course Outcomes:

After the end of the course, the student will be able to

C513.1	Examine data mining tasks, KDD process, challenges and data types of data.
C513.2	Apply Data Pre processing techniques to make data sets ready for mining.
C513.3	Identify the frequent patterns and association rules from transactional datasets.
C513.4	Apply pattern mining to the multilevel and multi-dimensional dataset.
C513.5	Classify the real world data into appropriate classes using various supervised learning techniques and measure its performance.
C513.6	Apply clustering and outlier detection techniques on given data sets and evaluate goodness measures.

Unit – I **Data Mining**

Introduction – Data mining, Kinds of data to be mined, Kinds of patterns to be mined, Technologies used, Applications of data mining, Major Issues in Data Mining.

Know Your Data – Data Objects and Attribute types, Basic statistical descriptions of data, Data visualization, Measuring data similarity and dissimilarity.

Unit – II **Data Preprocessing**

Data Preprocessing – Overview, Data cleaning- Missing values, Noisy data, Data cleaning as a process, Data integration - Entity identification problem, Redundancy and correlation analysis, Tuple duplication, Data value conflict detection and resolution.

Data Reduction & Transformation – Overview of data reduction strategies, Principal Component Analysis, Attribute subset selection, Parametric data reduction, Histograms, Clustering, Sampling, Data cube aggregation, Data transformation and Data discretization.

Unit – III **Association Rule Mining**

Mining Frequent Patterns – Basic concepts, Frequent itemset mining methods, Pattern evaluation methods.

Advanced Pattern Mining – Pattern mining in multilevel, Multidimensional space, Constraint-Based frequent pattern mining, Mining High-dimensional data and Colossal patterns, Pattern exploration and Application.

Unit – IV	Classification
Classification and Prediction – Basic concepts, Decision tree induction, Bayesian classification, Rule based classification, Model evaluation and selection, Lazy Learner.	
Unit – V	**Clustering and Application**
Cluster analysis, Partitioning methods, Hierarchical methods, Density based methods, Grid based methods, Evaluation of clustering. **Outlier Detection** – Outlier Analysis, Outlier detection methods, Statistical approaches, Proximity based approaches.	

TEXT BOOK:

1. Data Mining – Concepts and Techniques – Jiawei Han & Micheline Kamber, 3rd Edition Elsevier.

REFERENCE BOOKS:

1. Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques (Second Edition), Morgan Kaufmann, 2005.
2. Data Mining Introductory and Advanced topics – Margaret H Dunham, PEA.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS514PE	**Optimization Techniques** **(Professional Elective – I)**	**3**	**0**	**0**	**3**
Pre-requisites	**1. Mathematics – I** **2. Mathematics – II.**				

Course Description: The course introduce various optimization techniques like classical, linear programming, transportation problem, simplex algorithm, dynamic programming. Constrained and unconstrained optimization techniques for solving and optimizing electrical and electronic engineering circuits design problems in real world situations. The course also explains the concept of Dynamic programming and its applications to project implementation

Course Outcomes:

After the end of the course, the student will be able to

C514.1	Apply classical optimization techniques, linear programming, simplex algorithm
C514.2	Relate transportation and Assignment problem
C514.3	Illustrate Classical Optimization Techniques, Optimization with equality constraints.
C514.4	Use Multivariable Optimization with inequality constraints and Variable Nonlinear Unconstrained Optimization
C514.5	Analyze unconstrained optimization and constrained non-linear programming and Gradient methods
C514.6	Apply dynamic programming

UNIT – I

Introduction and Classical Optimization Techniques: Statement of an Optimization problem – design vector – design constraints – constraint surface – objective function – objective function surface - classification of Optimization problems.

Linear Programming: Standard form of a linear programming problem – geometry of linear programming problems – definitions and theorems – solution of a system of linear simultaneous equations – pivotal reduction of a general system of equations – motivation to the simplex method – simplex algorithm.

UNIT – II

Transportation Problem: Finding initial basic feasible solution by north – west corner rule, least cost method and Vogel's approximation method – testing for optimality of balanced transportation problems. Degeneracy.

Assignment problem – Formulation – Optimal solution - Variants of Assignment Problem; Traveling Salesman problem.

UNIT - III	
Classical Optimization Techniques: Single variable Optimization – multi variable Optimization without constraints – necessary and sufficient conditions for minimum/maximum – multivariable Optimization with equality constraints: Solution by method of Lagrange multipliers – Multivariable Optimization with inequality constraints: Kuhn – Tucker conditions. Single Variable Nonlinear Unconstrained Optimization: Elimination methods: Uni Model function-its importance, Fibonacci method & Golden section method.	
UNIT – IV	
Multi variable nonlinear unconstrained optimization: Direct search methods – Univariant method, Pattern search methods – Powell's, Hooke - Jeeves, Rosenbrock's search methods. **Gradient methods:** Gradient of function & its importance, Steepest descent method, Conjugate direction methods: Fletcher Reeves method & variable metric method.	
UNIT – V	
Dynamic Programming: Dynamic programming multistage decision processes – types – concept of sub optimization and the principle of optimality – computational procedure in dynamic programming – examples illustrating the calculus method of solution - examples illustrating the tabular method of solution.	

TEXT BOOKS:

1. Engineering Optimization Theory and Practice by Singiresu S. Rao, John Viley & Sons.
2. Optimization for Engineering Design by Kalyanmoy Deb, PHI.

REFERENCE BOOKS:

1. George Bernard Dantzig, Mukund Narain Thapa, "Linear programming", Springer series in Operations Research 3rd Edition, 2003.
2. H. A. Taha, "Operations Research: An Introduction", 8th Edition, Pearson/Prentice Hall, 2007.
3. Optimization Techniques by Belegundu & Chandrupatla, Pearson Asia.
4. Optimization Techniques Theory And Practice by M.C. Joshi, K. M. Moudgalya, Narosa Publications.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS521PE	**Computer Graphics** **(Professional Elective – II)**	3	0	0	3
Pre - Requisites	**1. Engineering Graphics** **2. Programming for Problem Solving**				

Course Description: This course contains the fundamentals of drawing and filling primitives, exploring the attributes associated with output primitives, understanding both two-dimensional and three-dimensional geometric transformations, and delving into the concepts of two-dimensional and three-dimensional viewing.

Course Outcomes:
At the end of the course, the student will be able to

C521.1	Explain application of computer graphics and apply different algorithms for drawing line, circle, polygon and polygon filling.
C521.2	Determine effects of Two-Dimensional geometric transformations on points, lines and planes.
C521.3	Explain window to view-port transformation
C521.4	Apply various clipping algorithms.
C521.5	Elaborate interpolation of line and space curves using Splines and Bezier curves.
C521.6	Determine the effects of Three-Dimensional geometric transformations on Three - Dimensional objects and explain the method of Three-Dimensional viewing and clipping

Unit – I	**Introduction, Output primitives, Polygon Filling**

Introduction: Application areas of Computer Graphics, overview of graphics systems
Output primitives: Points and lines, line drawing algorithms (DDA and Bresenham's Algorithm) circle- generating algorithms and ellipse - generating algorithms
Filled Area primitives: Scan-line polygon fill algorithm, boundary-fill, flood-fill algorithm, Attributes of Output primitives- line attributes and curve attributes.

Unit – II	**Two-Dimensional Geometric Transformations**

Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transformations, transformations between coordinate systems.

Unit – III	**Two-Dimensional Viewing**

The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Clipping operations, point clipping, Line clipping-Cohen Sutherland algorithms, Polygon clipping-Sutherland Hodgeman polygon clipping algorithm.

Unit – IV	Three-Dimensional Object Representation
Polygon surfaces, quadric surfaces, spline representation, Bezier curve, B-Spline curves, Bezier and B-Spline surfaces.	
Unit – V	**Three-Dimensional Geometric and Modelling Transformations**
Translation, rotation, scaling, reflection and shear, composite transformations. **Three-dimensional viewing**: Viewing pipeline, viewing coordinates, projections, view volume, General perspective- Projection Transformation, clipping.	

TEXT BOOK:

1. Computer Graphics C version, Donald Hearn and M. Pauline Baker, Pearson Education

REFERENCE BOOKS:

1. Procedural elements for Computer Graphics, David F Rogers, Tata Mc Graw hill, 2nd edition.
2. Principles of Interactive Computer Graphics", Neuman and Sproul, TMH.
3. Principles of Computer Graphics, Shalini Govil, Pai, 2005, Springer.
4. Computer Graphics Principles & Practice, second edition in C, Foley, VanDam, Feiner and Hughes, Pearson Education.
5. Computer Graphics, Steven Harrington, TMH.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS522PE	**Information Retrieval Systems** **(Professional Elective – II)**	3	0	0	3
Pre - Requisites	**1. Data structures** **2. Database Management Systems**				

Course Description:
The course aims to provide the concepts of information retrieval and their application to locate relevant information in a large corpus of documents.

Course Outcomes
At the end of the course, the student will be able to

C522.1	Understand Information Retrieval Systems' (IRS) principles, capabilities, and functionalities.
C522.2	Choose appropriate data structure, file structure and indexing mechanism for efficient retrieval.
C522.3	Differentiate among various classes of automatic indexing methods and clustering techniques.
C522.4	Select suitable search technique based on the context.
C522.5	Apply visualization techniques for efficient presentation of information.
C522.6	Make use of the algorithms for different media data.

Unit – I

Introduction to Information Retrieval Systems: Definition of Information Retrieval System, Objectives of Information Retrieval Systems, Functional Overview, Relationship to Database Management Systems, Digital Libraries and Data Warehouses
Information Retrieval System Capabilities: Search Capabilities, Browse Capabilities, Miscellaneous Capabilities

Unit – II

Cataloging and Indexing: History and Objectives of Indexing, Indexing Process, Information Extraction
Data Structure: Introduction to Data Structure, Stemming Algorithms, Inverted File Structure, N-Gram Data Structures, PAT Data Structure, Signature File Structure, Hypertext and XML Data Structures, Hidden Markov Models.

Unit – III

Automatic Indexing: Classes of Automatic Indexing, Statistical Indexing, Natural Language, Concept Indexing, Hypertext Linkages
Document and Term Clustering: Introduction to Clustering, Thesaurus Generation, Item Clustering, Hierarchy of Clusters

Unit – IV	
User Search Techniques: Search Statements and Binding, Similarity Measures and Ranking, Relevance Feedback, Weighted Searches of Boolean Systems, Searching the INTERNET and Hypertext **Information Visualization:** Introduction to Information Visualization, Cognition and Perception, Information Visualization Technologies	
Unit – V	
Text Search Algorithms: Introduction to Text Search Techniques, Software Text Search Algorithms, Hardware Text Search Systems **Multimedia Information Retrieval (MIR):** Spoken Language Audio Retrieval, Non-Speech Audio Retrieval, Graph Retrieval, Imagery Retrieval, Video Retrieval	

TEXT BOOK:

1. Information Storage and Retrieval Systems – Theory and Implementation, Second Edition, Gerald J. Kowalski, Mark T. Maybury.

REFERENCE BOOKS:

1. C. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.
2. Frakes, W.B., Ricardo Baeza-Yates: Information Retrieval Data Structures and Algorithms, Prentice Hall, 1992.
3. Information Storage & Retrieval by Robert Korfhage – John Wiley & Sons.
4. Modern Information Retrieval by Yates and Neto Pearson Education.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS523PE	**Data Analytics** **(Professional Elective – II)**	3	0	0	3
Pre - Requisites	**1. Database Management Systems** **2. Knowledge of probability and statistics**				

Course Description

This course explores the fundamental concepts of analytics, its principles and methods using statistical analysis. It contains supervised and unsupervised models with estimation to discover interesting patterns and analyze its impact in business decisions. This also covers the various search methods and visualization techniques to carry out standard data visualization and formal inference procedures.

Course Outcomes:

At the end of the course, the student will be able to

C523.1	Fetch data from various sources and make it ready for analysis.
C523.2	Make use of various tools and technologies for data analysis.
C523.3	Apply regression techniques to data and evaluate performance.
C523.4	Build supervised and unsupervised learning models for objective segmentation.
C523.5	Build models for time series and evaluate its performance.
C523.6	Visualize the data and interpret the insights that exist in the data.

Unit – I	**Data Management**

Design Data Architecture and manage the data for analysis, understand various sources of Data like Sensors/Signals/GPS etc. Data Management, Data Quality - Noise, Outliers, Missing Values, Duplicate data, and Data Processing.

Unit – II	**Data Analytics**

Introduction to Analytics, Introduction to Tools and Environment, Application of Modeling in Business, Databases & Types of Data and Variables, Data Modeling Techniques, Missing Imputations etc. Need for Business Modeling.

Unit – III	**Regression**

Introduction, Correlations and Relationships, Visual Look at relationships, Linear Regression, Blue property assumptions, Least Square Estimation, Non-linear regression, Logistic Regression, Advantages and Disadvantages of Regression Models, Analytics applications to various Business Domains

Unit – IV	Objective Segmentation
Regression Vs Segmentation – Supervised and Unsupervised Learning, Tree Building – Regression, Classification, Overfitting, Pruning and Complexity, Multiple Decision Trees etc. Time Series Methods: Arima, Measures of Forecast Accuracy, ETL approach, Extract features from generated model as Height, Average Energy etc and analyze for prediction.	
Unit – V	**Data Visualization**
Pixel-Oriented Visualization Techniques, Geometric Projection Visualization Techniques, Icon-Based Visualization Techniques, Hierarchical Visualization Techniques, Visualizing Complex Data and Relations. Excellence in Visualization, Types of Charts, Visualization	

TEXT BOOKS:

1. Student's Handbook for Associate Analytics – II, III.
2. Data Mining Concepts and Techniques, Han, Kamber, 3rd Edition, Morgan Kaufmann Publishers.
3. Data Analytics -Anil Maheswaran, McGrawHill.

REFERENCE BOOKS:

1. Introduction to Data Mining, Tan, Steinbach and Kumar, Addision Wisley, 2006.
2. Data Mining Analysis and Concepts, M. Zaki and W. Meira.
3. Mining of Massive Datasets, Jure Leskovec Stanford Univ. Anand Rajaraman Milliway Labs Jeffrey D Ullman Stanford Univ.
4. Beautiful Visualization, Looking at Data Through the Eyes of Experts, Julie Steele and Noah Iliinsky, Oreilly.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS524PE	**High Performance Computing (Professional Elective – II)**	**3**	**0**	**0**	**3**
Pre-requisite	**1. Computer Organization & Architecture** **2. Operating Systems**				

Course Description: The course introduces parallel computing algorithm design, modeling and solving problems using different types of parallel computing architectures. The course elucidates the various paradigms in algorithm design for computationally intensive applications, modern multi-processor and multi-core architectures.

Course Outcomes:
After the end of the course, the student will be able to

C524.1	Understand Modern Processors architectures and optimization techniques.
C524.2	Discuss different parallel computing architectures and networks.
C524.3	Illustrate vector processing, memory bottlenecks, data and thread-level parallelism
C524.4	Design parallel algorithms and measure their performance with OpenMP.
C524.5	Develop the various programming frameworks like MPI.
C524.6	Construct various efficient parallel programs with CUDA.

UNIT – I

Modern Processors: Stored-Program Computer Architecture, General-Purpose cache-based Microprocessor Architecture, Memory Hierarchies, Multicore processors, Multithreaded processors, Vector processors.
Basic optimization techniques for serial code: Scalar profiling, Common sense optimizations, Simple measures, large impact, The role of compilers, Data access optimization.

UNIT – II

Parallel computers: Taxonomy of parallel computing paradigms, Shared-memory computers, Distributed-memory computers, Hierarchical (hybrid) systems, Networks.
Basics of parallelization: Need for Parallelism, Parallel scalability.

UNIT - III

Shared-memory parallel programming with OpenMP: Introduction to OpenMP, Profiling OpenMP programs, Performance pitfalls, Case study: OpenMP-parallel Jacobi algorithm.

UNIT – IV

Distributed-memory parallel programming with MPI: Message passing, Introduction to MPI, MPI performance tools, Communication parameters, Synchronization, serialization, contention, Reducing communication overhead, Case study: Parallel sparse matrix-vector multiply.

UNIT – V	
CUDA: Understanding the CUDA computing model and the API using nvcc compiler, Introduction to modern supercomputing architectures featuring NVIDIA processors	

TEXT BOOK:

1. Georg Hager, Gerhard Wellein, Introduction to High Performance Computing for Scientists and Engineers, Chapman & Hall / CRC Computational Science series, 2011.
2. CUDA Programming A Developer's Guide to Parallel Computing with GPUs by Shane Cook, Morgan Kaufman Publishers

REFERENCE BOOKS:

1. Introduction to Parallel Computing, Second Edition, Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta, Addison-Wesley, 2003
2. Parallel Computing – Theory and Practice, Second Edition, Michaek J. Quinn, Tata McGraw-Hill Edition.
3. Parallel Computers – Architectures and Programming, V. Rajaraman, C. Siva Ram Murthy, PHI.
4. Parallel Programming in C with MPI and OpenMP by Michael Quinn, McGraw-Hill Publisher.
5. Computer Architecture A Quantitative Approach by John Hennessey and David Patterson, Morgan Kaufman Publishers.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS504PC	**DevOps Lab**	0	0	2	1

Course Description: This lab helps to develop a sustainable base for applications and ensure high scalability. DevOps aims to shorten the software development lifecycle to provide continuous delivery with high-quality.

Course Outcomes:
After completion of this course, the student will be able to:

C504.1	Practice Source code management using GIT
C504.2	Build the environment for software application development using Jenkins.
C504.3	Apply different project management, integration and development tools
C504.4	Use Selenium tool for automated testing of application

LIST OF PROGRAMS

CYCLE 1:

1. Write code for a simple user registration form for an event using micro services frameworks.
2. Explore Git and GitHub commands.
3. Practice Source code management on GitHub. Experiment with the source code in exercise 1.
4. Jenkins installation and setup, explore the environment.
5. Demonstrate continuous integration and development using Jenkins.
6. Explore Docker commands for content management.
7. Develop a simple containerized application using Docker.

CYCLE 2:

8. Integrate Kubernetes and Docker
9. Automate the process of running containerized applications for exercise 7 using Kubernetes.
10. Install and Explore Karate and Spring boot testing for automated testing.
11. Write a simple program in JavaScript and perform testing using Karate testing.
12. Develop test cases for the above containerized application using Spring boot testing.

REFERENCE BOOKS:

1. Joakim Verona., Practical DevOps, Packt Publishing, 2016.
2. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint. Wiley publications.
3. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley.
4. Benjamin Bischoff, Writing API Tests with Karate: Enhance your API testing for improved security and performance, Kindle Edition.
5. Testing Spring Boot Applications Demystified, Philip Riecks.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS506PC	**UI design- Flutter** **(Common to CSE,CSE(AIML) & IT)**	0	0	2	1

Course Description: The Flutter UI design lab provides hands-on experience in creating visually appealing and responsive user interfaces using the Flutter framework. Participants learn to build cross-platform applications with a single codebase, utilizing Flutter's widget-based architecture to craft beautiful designs for both iOS and Android platforms. The lab emphasizes practical skills in UI development, enhancing proficiency in Flutter's rich set of tools and components.

Course Outcomes:

After completion of this course, the student will be able to:

C506.1	Apply the basics of the Dart programming language, Flutter Widgets.
C506.2	Create responsive UI Widgets using navigator in Flutter Applications
C506.3	Implement a form with various input fields and animations, along with validation and error handling.
C506.4	Demonstrate Flutter Application using REST API and Flutter debugging tools.

LIST OF PROGRAMS

CYCLE 1:

1. a) Install Flutter and Dart SDK.

   b) Write a simple Dart program to understand the language basics.
2. a) Explore various Flutter widgets (Text, Image, Container, etc.).

   b) Implement different layout structures using Row, Column, and Stack widgets.
3. a) Design a responsive UI that adapts to different screen sizes.

   b) Implement media queries and breakpoints for responsiveness.
4. a) Set up navigation between different screens using Navigator.

   b) Implement navigation with named routes.
5. a) Design stateful and stateless widgets.

   b) Implement state management using set State and Provider.
6. a) Create custom widgets for UI elements.

   b) Apply styling using themes and custom styles.

CYCLE 2:

7. a) Design a form with various input fields.

   b) Implement form validation and error handling.

8. a) Add animations to UI elements using Flutter's animation framework.

   b) Experiment with different types of animations (fade, slide, etc.).

9. a) Fetch data from a REST API.

   b) Display the fetched data in a meaningful way in the UI.

10. a) Write unit tests for UI components.

    b) Use Flutter's debugging tools to identify and fix issues.

REFERENCE BOOKS:

1. Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development.
2. “Flutter in Action” by Eric Windmill.
3. “Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2” by Alessandro Biessek.
4. “Flutter Cookbook” by Paul Deitel and Harvey Deitel.
5. “Google Flutter Mobile Development Quick Start Guide: Get up and running with iOS and Android development” by Prajyot Mainkar.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year I Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS507PC	**Computer Networks Lab (Common to CSE & CSE(AIML))**	0	0	2	1
Pre - Requisite	**Programming for problem solving**				

Course Description:

This lab introduces, implementation of various framing methods, error detection and correction mechanisms. Design and implementation of various routing protocols and congestion control. It helps in providing & monitoring security using tools like nmap, Wireshark and NS2.

Course Outcomes:

After completion of this course, the student will be able to:

C507.1	Implement various Framing methods, Error Control methods and Sliding window protocols.
C507.2	Analyze various protocols, operating system detection using appropriate monitoring tools.
C507.3	Evaluate various routing protocols and congestion control mechanisms.
C507.4	Evaluate the performance of routing protocols and IEEE 802.x standards using NS2 simulator

LIST OF PROGRAMS

CYCLE 1:

1. Implement the data link layer framing methods such as character count, character stuffing and bit stuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC-CCIP.
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4. Wireshark
    i. Packet Capture Using Wire shark
    ii. Starting Wire shark
    iii. Viewing Captured Traffic
    iv. Analysis and Statistics & Filters.
5. Nmap
    i. How to run Nmap scan
    ii. Operating System Detection using Nmap

CYCLE 2:

6. Implement Dijsktra's algorithm to compute the shortest path through a network
7. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
8. Implement distance vector routing algorithm for obtaining routing tables at each node.
9. Write a program for congestion control using Leaky bucket algorithm.
10. Do the following using NS2 Simulator
    i. NS2 Simulator-Introduction
    ii. Simulate to Find the Number of Packets Dropped
    iii. Simulate to Find the Number of Packets Dropped by TCP/UDP
    iv. Simulate to Find the Number of Packets Dropped due to Congestion
    v. Simulate to Compare Data Rate & Throughput.
    vi. Simulate to Plot Congestion for Different Source/Destination
    vii. Simulate to Determine the Performance with respect to Transmission of Packets

TEXT BOOKS:

1. Computer Networks, Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI.

REFERENCE BOOKS:

1. An Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson Education.
2. Data Communications and Networking – Behrouz A. Forouzan. 3rd Edition, TMH.

WEB REFERENCES:

1. https://nmap.org/
2. https://www.wireshark.org/
3. https://ns2simulator.com/ns2-download/

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS601PC	**Machine Learning**	3	0	0	3

Course Description: The course introduces the basic concepts and techniques of Machine Learning with thorough understanding of the Supervised and Unsupervised learning techniques and its differences. The course elucidates the principles of evolutionary computing algorithms, ensembling techniques for increased prediction accuracy, and enlightens the principles of Reinforcement learning.

Course Outcomes:
At the end of the course, the student will be able to

C601.1	Understand the basic concepts of Machine Learning Techniques.
C601.2	Apply the neural network concepts with Perceptron and Back Propagation.
C601.3	Evaluate various supervised, unsupervised learning algorithms with ensemble techniques.
C601.4	Make use of Dimensionality Reduction concepts for model building.
C601.5	Apply evolutionary computing algorithms approach for search and optimization.
C601.6	Analyze the concepts of Reinforcement Learning for building autonomous Systems**.**

Unit – I **Introduction to Learning**

Learning – Types of Machine Learning – Supervised Learning – The Brain and the Neuron – Design a Learning System – Perspectives and Issues in Machine Learning – Concept Learning Task – Concept earning as Search – Finding a Maximally Specific Hypothesis – Version Spaces and the Candidate Elimination Algorithm – Linear Discriminants: – Perceptron – Linear Separability – Linear Regression.

Unit – II **Multi-layer Perceptron– Going Forwards – Going Backwards**

Multi-layer Perceptron – Going Forwards – Going Backwards: Back Propagation Error – Multi-layer Perceptron in Practice – Examples of using the MLP – Overview – Deriving Back - Propagation – Radial Basis Functions and Splines – Concepts – RBF Network – Curse of Dimensionality – Interpolations and Basis Functions – Support Vector Machines

Unit – III **Learning with Trees**

Learning with Trees – Decision Trees – Constructing Decision Trees – Classification and Regression Trees – Ensemble Learning – Boosting – Bagging – Different ways to Combine Classifiers – Basic Statistics – Gaussian Mixture Models – Nearest Neighbour Methods – Unsupervised Learning – K means Algorithms.

Unit – IV	Dimensionality Reduction and Evolutionary Learning
Dimensionality Reduction – Linear Discriminant Analysis – Principal Component Analysis – Factor Analysis – Independent Component Analysis – Locally Linear Embedding – Isomap – Least Squares Optimization **Evolutionary Learning** – Genetic algorithms – Genetic Offspring: - Genetic Operators – Using Genetic Algorithms	
Unit – V	**Reinforcement Learning**
Reinforcement Learning – Overview – Getting Lost Example Markov Chain Monte Carlo Methods – Sampling – Proposal Distribution – Markov Chain Monte Carlo – Graphical Models – Bayesian Networks – Markov Random Fields – Hidden Markov Models – Tracking Methods	

TEXTBOOK:

1. Stephen Marsland **—** Machine Learning – An Algorithmic Perspective, Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series.

REFERENCE BOOKS:

1. Tom M Mitchell **—** Machine Learning, First Edition, McGraw Hill Education, 2017.
2. Marco Gori, Alessandro Betti, Stefano Melacci, Machine Learning A Constraint-Based Approach, 2023.
3. Peter Flach **—** Machine Learning: The Art and Science of Algorithms that Make Sense of Data, First Edition, Cambridge University Press, 2012.
4. Jason Bell **—** Machine learning – Hands on for Developers and Technical Professionals, First Edition, Wiley, 2014.
5. Ethem Alpaydin **—** Introduction to Machine Learning 3e (Adaptive Computation and Machine Learning Series), Third Edition, MIT Press, 2014.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS602PC	**Formal Languages and Automata Theory**	3	0	0	3
Pre - Requisite	**1. Discrete Mathematics** **2. Design and Analysis of Algorithms**				

Course Description: This course introduces a set of abstract machines, serve as models for computation - finite automata, pushdown automata and turing machines. This course also deals with undecidability and the relationship between these automata and formal languages.

Course Outcomes:

At the end of the course, the student will be able to

C602.1	Design Finite Automata (FA) machines, minimization, achieve conversions among them.
C602.2	Construct Regular expressions and Test for regular languages
C602.3	Analyze Left Most Derivation (LMD), Right Most Derivation (RMD) and normal forms for context free grammars.
C602.4	Design Pushdown Automata for Languages, grammars and conversions.
C602.5	Design appropriate Turing Machine for a given problem.
C602.6	Distinguish between decidability and undecidability.

Unit – I **Introduction to Finite Automata**

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems.

Nondeterministic Finite Automata (NFA): Formal Definition, an application, Text Search, NFA with ε-Transitions.

Deterministic Finite Automata (DFA): Definition of DFA, Process Strings, The language of DFA, Conversion of NFA with ε-transitions to NFA without ε-transitions. Conversion of NFA to DFA, Moore and Melay machines, Equivalence of Automata and Minimization of Automata.

Unit – II **Regular Expressions**

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions and Regular Expression to Finite Automata.

Pumping Lemma for Regular Languages, Statement of the pumping lemma, Applications of the Pumping Lemma, closure properties of Regular languages, Decision Properties of Regular Languages.

Unit – III **Context-Free Grammars**

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Trees, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.

Normal Forms for Context- Free Grammars: Eliminating useless symbols, Eliminating €-Productions, Unit productions. Chomsky Normal form and Greibach Normal form.	
Unit – IV	**Push Down Automata**
Push Down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state, Acceptance by empty stack, Deterministic Pushdown Automata and Non- Deterministic Pushdown Automata, Conversion of CFG to PDA, PDA to CFG. **Pumping Lemma for Context-Free Languages:** Statement of pumping lemma, Applications, closure properties of CFL's, decision Properties of CFL's	
Unit – V	**Turing Machines**
Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, the language of a Turing machine, types of Turing machine. Turing machines and halting. **Undecidability:** Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines, Recursive languages, Properties of recursive languages. Post's Correspondence Problem, Modified Post Correspondence problem, Other Undecidable Problems, Counter machines.	

TEXT BOOKS:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Pearson Education.
2. Mishra and Chandra Shekaran, Theory of Computer Science – Automata languages and computation, 2nd edition, PHI.
3. K.V.N Sunitha, N. Kalyani, Formal Languages and Automata Theory, Pearson.

REFERENCE BOOKS:

1. John C Martin, Introduction to Languages and The Theory of Computation, TMH.
2. Daniel I.A. Cohen, John Wiley, Introduction to Computer Theory.
3. P. K. Srimani, Nasir S. F. B, A Textbook on Automata Theory, Cambridge University Press.
4. Michael Sipser, Introduction to the Theory of Computation, 3rd edition, Cengage Learning.
5. Kamala Krithivasan, Rama R, Introduction to Formal languages Automata Theory and Computation Pearson.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS603PC	**Artificial Intelligence**	3	0	0	3
Pre - Requisite	**1. Programming for problem solving** **2. Data Structures**				

Course Description: This course explains the theory and development of computer systems capable of performing tasks that historically required human intelligence, such as recognizing speech, making decisions, and identifying patterns.

Course Outcomes:

At the end of the course, the student will be able to

C603.1	Understand search strategies and intelligent agents
C603.2	Understand different adversarial search techniques
C603.3	Interpret search algorithms for any AI problem
C603.4	Apply propositional logic, predicate logic for knowledge representation
C603.5	Apply AI techniques to solve problems of game playing, and machine learning
C603.6	Design Bayesian networks to model complex relationships and understand the semantics encoded in these graphical structures.

Unit – I	**Introduction to Artificial Intelligence**

Introduction to AI: Intelligent Agents, problem-Solving Agents, Searching for Solutions, Uninformed Search Strategies: Breadth-first search, Uniform cost search, Depth-first search, Iterative deepening Depth-first search, Bidirectional search, Informed (Heuristic) Search Strategies: Greedy best-first search, A* search, Heuristic Functions, Beyond Classical Search: Hill-climbing search, Simulated annealing search, Local Search in Continuous Spaces

Unit – II	**Problem Solving by Search-II and Propositional Logic**

Adversarial Search: Games, Optimal Decisions in Games, Alpha–Beta Pruning, Imperfect Real - Time Decisions. Constraint Satisfaction Problems: Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Local Search for CSPs, The Structure of Problems.

Propositional Logic: Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses, Forward and backward chaining, Effective Propositional Model Checking, Agents Based on Propositional Logic.

Unit – III	**Logic and Knowledge Representation**

First-Order Logic: Representation, Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic.

Inference in First-Order Logic: Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution

Unit – IV	Knowledge Representation
Knowledge Representation: Ontological Engineering, Categories and Objects, Events. Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information. **Classical Planning:** Definition of Classical Planning, Algorithms for Planning with State-Space Search, Planning Graphs, other Classical Planning Approaches, Analysis of Planning approaches.	
Unit – V	**Uncertain knowledge and Learning Uncertainty**
Uncertain knowledge and Learning Uncertainty: Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and Its Use **Probabilistic Reasoning:** Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First-Order Probability, Other Approaches to Uncertain Reasoning; Dempster-Shafer theory	

TEXT BOOK:

1. Artificial Intelligence: A Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCE BOOKS:

1. Artificial Intelligence, 3rd Edn, E. Rich and K. Knight (TMH)
2. Artificial Intelligence, 3rd Edn., Patrick Henry Winston, Pearson Education.
3. Artificial Intelligence, Shivani Goel, Pearson Education.
4. Artificial Intelligence and Expert systems – Patterson, Pearson Education

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS611PE	**Full Stack Development (Professional Elective – III)**	3	0	0	3
Pre - Requisite	**NODE JS/REACT JS/ DJANGO**				

Course Description: This course contains full-stack components such as Node.js, MongoDB, Express, React, and Angular JS to develop web page.

Course Outcomes:
At the end of the course, the student will be able to

C611.1	Understand the Full-stack components for developing web applications.
C611.2	Apply packages of NodeJS to work with Data, Files, HTTP Requests and Responses.
C611.3	Use MongoDB database for storing and processing huge data.
C611.4	Explore MongoDB database connection with NodeJSapplication.
C611.5	Design faster and more effective single-page applications using Express and Angular.
C611.6	Create interactive user interfaces with react components.

Unit – I **Introduction to Full Stack Development**

Basic Web Development Framework- User, Browser, Webserver, BackendServices
Full Stack Components - Node.js, MongoDB, Express, React, Angular.

Unit – II **Node.js**

NodeJS: Understanding Node.js, Working with Node Packages, creating a Node.js Application, Understanding the Node.js Event Model, Adding Work to the EventQueue, Implementing Callbacks, Working with JSON.
Accessing the File System from Node.js: Opening, Closing, Writing, Reading Files, Processing Query strings and Form parameters, Understanding Request, Response, and Server Objects, Implementing HTTP service in Node.js

Unit – III **MongoDB**

Understanding NoSQL and MongoDB, Building the MongoDB Environment, MongoDB CRUD operations, Accessing and Manipulating MongoDB Collections
Adding the MongoDB Driver to Node.js, Connecting to MongoDB from Node.js, and objects Used in the MongoDB Node.js Driver.

Unit – IV **Express and Angular**

Getting Started with Express: Configuring Routes, Using Requests and Response Objects.
Angular: importance of Angular, Understanding Angular, creating a Basic Angular Application, Angular Components, Expressions, Data Binding, Built-in Directives, Custom Directives, Implementing Angular Services in Web Applications.

Unit – V	React
Need of React, Simple React Structure, The Virtual DOM, React Components, Introducing React Components, Creating Components in React, Data and Data Flow in React. Rendering and Life Cycle, Methods in React, working with forms in React, integrating third-party libraries, Routing in React.	

TEXT BOOK:

1. Brad Dayley, Brendan Dayley, Caleb Dayley, Node.js, MongoDB and Angular WebDevelopment, 2nd Edition, Addison-Wesley, 2019.
2. Mark Tielens Thomas, React in Action, 1st Edition, Manning Publications.

REFERENCE BOOKS:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, Apress, 2019.
2. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer, 1st edition, Apress, 2018.
3. Kirupa Chinnathambi, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, 2nd edition, Addison-Wesley Professional, 2018.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS612PE	**Scripting Languages (Professional Elective – III)**	3	0	0	3
Pre - Requisite	**Programming for Problem Solving**				

Course Description: This course contains script programming paradigm and introduces languages namely Ruby, Perl, TCL and Tk components.

Course Outcomes:
At the end of the course, the student will be able to

C612.1	Differentiate among typical scripting languages, typical system and application programming languages.
C612.2	Apply the basic concepts of Perl. Design programs in Advanced Perl and Interfacing Different operating Systems Security Issues.
C612.3	Develop Ruby web applications along with SOAP and web services. Incorporate a Ruby interpreter with Embed Ruby.
C612.4	Design the scripts using TCL and Tk along with Perl.
C612.5	Implement the appropriate utilities to manage the Linux environment.
C612.6	Implement shell scripts to automate the administrative process

Unit – I	**Introduction to Scripting PERL, Advanced PERL**

Scripts and Programs, Origin of Scripting, Scripting Today, Characteristics of Scripting Languages, Uses for Scripting Languages, Web Scripting, and the universe of Scripting Languages.

PERL- Names and Values, Variables, Scalar Expressions, Control Structures, Arraysand Lists, Hashes, Strings, Patterns and Regular Expressions, Subroutines.
Advanced PERL - Finer points of looping, Pack and unpack, filesystem, eval, Data Structures, Packages, Modules, Objects, Interfacing to the operating system, Creating Internet-aware applications, Dirty Hands Internet Programming, Security Issues.

Unit – II	**Introduction to Ruby, Extending Ruby**

The structure and Execution of Ruby Programs- Lexical Structure, Syntactic Structure, File Structure, Program Encoding, Program Execution. Ruby and web: Writing CGI scripts, Cookies, Choice of Webservers, SOAP and Web services

Ruby Tk – Simple Tk Application, Widgets, Binding events, Canvas, Scrolling

Extending Ruby: Ruby Objects in C, the Jukebox Extension, Memory allocation, Ruby Type System, Embedding Ruby to Other Languages, Embedding a Ruby Interpreter

Unit – III	TCL and Tk
TCLTCL Structure, syntax, Variables and Data in TCL, Control Flow, Data Structures, Input/Output, Procedures, Strings, Patterns, Files, Advance TCL- eval, source, exec and uplevel commands, Name spaces, Trapping errors, Event driven programs, Making applications Internet-aware, Nuts and Bolts Internet Programming, Security Issues-Running Untrusted Code, C Interface. **Tk** Tk-Visual ToolKits, Fundamental Concepts of Tk, Tk by example, Events and Binding, Perl-Tk	
Unit – IV	**Linux Shell Scripts**
Linux Utilities-File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities. Sed-Scripts, Operation, Addresses, Commands, awk-Execution, Fields and Records, Scripts, Operation, Patterns, Actions, Associative Arrays, String and Mathematical functions, System commands in awk, Applications	
Unit – V	**Shell Programming**
Shell programming with Bourne again shell(bash)- Introduction, Shell responsibilities, Pipes and Redirection, Running a Shell Script, The Shell as a Programming Language, Shell Meta Characters, File Name Substitution, Shell Variables, Command substitution, Shell commands, The environment, Quoting, Test Command, Control Structures, Arithmetic in Shell, Shell Script examples, Interrupt Processing, Functions, Debugging Shell Scripts.	

TEXT BOOKS:

1. The World of Scripting Languages, David Barron, Wiley Publications.
2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O'Reilly.
3. "Programming Ruby" The Pramatic Progammers guide by Dabve Thomas 2nd edition.
4. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH.
5. Unix System Programming using C++, T.Chan, PHI

REFERENCE BOOKS:

1. Open Source Web Development with LAMP Using Linux Apache, MySQL, Perl and PHP, James Lee and Bent Ware, Addison Wesley, Pearson Education.
2. PERL by Example, Ellie Quigley, Pearson Education.
3. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O'Reilly, SPD.
4. Tcl and the Tk Tool kit, Ousterhout, Pearson Education.
5. Perl Power, J. P. Flynt, Cengage Learning.
6. Beginning Linux Programming, 4th Edition, Neil Matthew, Richard Stones, Wrox.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS613PE	**Internet of Things (Professional Elective – III)**	3	0	0	3
Pre - Requisite	**Computer Networks**				

Course Description: The course introduces Physical Design of IoT, Logical Design of IoT, M2M, IoT System Management with NETCONF-YANG, Arduino board. The course elucidates IoT Systems – Logical design using Python programme, IoT Physical Devices and Endpoints using Raspberry Pi board, and IoT and Case studies on Agriculture, Environment.

Course Outcomes:

At the end of the course, the student will be able to

C613.1	Understand basic components in various IoT architectures
C613.2	Analyze different system management skills to address the challenges in implementation
C613.3	Apply Arduino programming skills for integration with the board.
C613.4	Make use of Python concepts to create solutions for diverse applications.
C613.5	Interface different components using Raspberry Pi board.
C613.6	Develop suitable solutions for the problems occurred in Industry.

Unit – I	**Introduction, Physical Design of IoT, Logical Design of IoT**

Introduction: Definition and Characteristics of IoT, Sensing, Actuation

Physical Design of IoT: Things in IoT, IoT protocols

Logical Design of IoT: IoT Functional blocks, IoT communication Models, IoT communication APIs

Unit – II	**IoT and M2M, IoT System Management with NETCONF-YANG**

IoT and M2M - Introduction, M2M, Difference between IoT and M2M

IoT System Management with NETCONF-YANG - Need for IoT system Management, Simple Network management protocol, and Network operator requirements.

Unit – III	**Arduino, IoT Systems – Logical design using Python**

Arduino: Introduction to Arduino Programming, Integration of Sensors and Actuators with Arduino Python Data types & Datastructures, Control flow, Functions, Modules, Packaging, File handling, Data/Time operations, Classes

Unit – IV	**IoT Physical Devices and Endpoints**

Basic building blocks of an IoT Device, Raspberry Pi, About the Board, Linux on Raspberry Pi **Raspberry Pi Interfaces**-Serial, SPI, I2C

Programming Raspberry PI with Python- Controlling LED with Raspberry Pi, Interfacing LED and Switch with Raspberry Pi, Interfacing a Light Sensor(LDR) with Raspberry Pi

Unit – V	Case studies
Home Automation, Environment-weather monitoring-weather reporting- air pollution monitoring, Agriculture, Health & Lifestyle.	

TEXT BOOK:

1. Internet of Things: A Hands-on Approach, by Arshdeep Bahga and Vijay Madisetti
2. The Internet of Things: Enabling Technologies, Platforms, and Use Cases, by Pethuru Raj and Anupama C. Raman (CRC Press)
3. Make sensors: Terokarvinen, kemo, karvinen and villey valtokari, 1st edition, maker media, 2014

REFERENCE BOOKS:

1. Waltenegus Dargie, Christian Poellabauer, Fundamentals of Wireless Sensor Networks: Theory and Practice.
2. Beginning Sensor networks with Arduino and Raspberry Pi – Charles Bell, Apress, 2013

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS614PE	**Software Testing Methodologies** **(Professional Elective – III)**	3	0	0	3
Pre - Requisite	**Software Engineering**				

Course Description This Course provides a comprehensive overview of various software testing methodologies, techniques and best practices to ensure the delivery of high-quality software products. This course also deals with the principles and processes involved in testing software applications from unit testing to system testing.

Course Outcomes:
At the end of the course, the student will be able to

C614.1	Understand the software testing foundations and bugs
C614.2	Apply suitable testing techniques for representation of flow graphs
C614.3	Apply data flow and interface testing techniques for identifying nice and ugly domains
C614.4	Choose appropriate path expression, KV charts for logic based testing strategies
C614.5	Examine the state graphs for state testing and the testability tips
C614.6	Explore graph matrices, matrix properties and node reduction algorithms.

Unit – I	**Software Testing fundamentals**

Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of Bugs.
Flow graphs and Path testing: Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

Unit – II	**Testing types- Transaction flow testing and Domain testing**

Transaction Flow Testing: transaction flows, transaction flow testing techniques.
Data Flow testing: Basics of data flow testing, strategies in data flow testing, application of data flow testing.
Domain Testing: Domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domains and testability

Unit – III	**Path testing and Logic based Testing**

Paths, Path products and Regular expressions: Path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.
Logic Based Testing: overview, decision tables, path expressions, kv charts, specifications.

Unit – IV	State testing and Transition Testing
State, State Graphs and Transition testing: State graphs, good & bad state graphs, state testing, Testability tips.	
Unit – V	**Applications of Graph matrices**
Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm.	

TEXT BOOK:

1. Software Testing techniques - BarisBeizer, Dreamtech, second edition.
2. Software Testing Techniques – SPD(Oreille)

REFERENCE BOOKS:

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS604PC	**Machine Learning Lab**	**0**	**0**	**2**	**1**
Pre - Requisite	**Python Programming Laboratory**				

Course Description: This Course implements fundamental statistical concepts using Python libraries. Various classification and regression models are built, which helps to gain practical experience in machine learning model selection and evaluation.

Course Outcomes:
After completion of this course, the student will be able to

C604.1	Implement statistical concepts required for data analysis
C604.2	Analyze data, model, model complexity and predict the trends.
C604.3	Correlate various machine learning algorithms along with their strengths and Weaknesses.
C604.4	Build predictive models from data and analyze the model performance.

LIST OF EXPERIMENTS:

CYCLE 1:

1. Write a python program to compute Central Tendency Measures: Mean, Median, Mode, and Measure of Dispersion: Variance and Standard Deviation.

2. Write a program to perform Arithmetic Array Operations (Addition, Subtraction, Multiplication, Division, Exponentiation, and Modulus) using Libraries such as Statistics, Math, Numpy and Scipy.

3. Implement and demonstrate the following by importing the dataset from load_ dataset.
   a. Pandas library function for data analysis & manipulation
   b. Matplotlib library functions for data visualization

4. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis for a given dataset.
   ['Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same', 'Yes']
   ['Sunny', 'Warm', 'High', 'Strong', 'Warm', 'Same', 'Yes']
   ['Rainy', 'Cold', 'High', 'Strong', 'Warm', 'Change', 'No']
   ['Sunny', 'Warm', 'High', 'Strong', 'Cool', 'Change', 'Yes']

5. Write a Python program to find out the correlation between Salary (dependent variable) and Experience (independent variable) using Simple Linear Regression.

CYCLE 2:

6. Write a Python program to find out the correlation between Salary (dependent variable) and Experience (independent variable) using Simple Linear Regression.
7. Using scikit-learn, perform House Price Prediction with Multiple Linear Regression (minihomeprices.csv)
8. Write a program to Predict Humidity using Decision Tree Algorithm.(daily_weather.csv)
9. Write a program to implement k-Nearest Neighbour classification algorithm using iris dataset.
10. Write a program to predict rainfall using Logistic Regression. (weatherAUS.csv)
11. Write a program to implement K-Means algorithm for clustering Mallcustomers (Mallcustomers.csv)

TEXT BOOK:

1. Tom M Mitchell ― Machine Learning, First Edition, McGraw Hill Education, 2017.

REFERENCE BOOK:

1. Machine Learning: An Algorithmic Perspective, Stephen Marshland, Taylor & Francis, Second Edition.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS605PC	**Artificial Intelligence Laboratory**	**0**	**0**	**2**	**1**

Course Description. This provides hands-on experience in implementing, experimenting with, and understanding various AI techniques and algorithms. Students engage in practical exercises.

Course Outcomes:
After the completion of this course, the student will be able to

C605.1	Demonstrate a deep understanding of fundamental search algorithms.
C605.2	Apply algorithmic techniques to implement games.
C605.3	Exhibit proficiency in solving complex problems through heuristic search algorithms
C605.4	Apply evaluation skills, to assess and select appropriate optimization techniques.

LIST OF PROGRAMS

CYCLE 1:

1. Create a program to implement linear regression.
2. Implement a decision tree classifier for a basic classification problem.
3. Implement the Breadth-First Traversal algorithm on a given graph.
4. Write a program for Depth-First Traversal in a graph.
5. Develop a two-player Tic-Tac-Toe game. Allow users to make moves on a 3x3 grid, and implement the logic to determine the winner or declare a draw.
6. Implement the search algorithm to solve the 8-Puzzle problem.

CYCLE 2:

7. Solve the classic Water-Jug Problem using a search algorithm. Given two jugs with different capacities, determine the sequence of actions needed to measure a specific quantity of water
8. Design a program to solve the Travelling Salesman Problem using a suitable algorithm.
9. Write a program to solve the Tower of Hanoi problem.
10. Solve the Monkey Banana Problem using a search or planning algorithm.

11. Implement the Alpha-Beta Pruning algorithm for optimizing the search in a minimax game tree.
12. Solve the 8-Queens Problem using a search algorithm. Implement a program to find a placement of eight queens on a chessboard such that no two queens attack each other

TEXT BOOK:

1. Artificial Intelligence a Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCE BOOKS:

1. Artificial Intelligence, 3rd Edn, E. Rich and K. Knight (TMH)
2. Artificial Intelligence, 3rd Edn., Patrick Henny Winston, Pearson Education.
3. Artificial Intelligence, Shivani Goel, Pearson Education.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS621PE	**Full Stack Development Lab (Professional Elective – III)**	**0**	**0**	**2**	**1**

Course Description This Lab introduces front and backend tools, to create web-based applications, and understands effective database access. This introduces fast, efficient, interactive, and scalable web applications using a run-time environment provided by the full-stack components.

Course Outcomes:
After completion of this course, the student will be able to

C621.1	Design flexible and responsive Web applications using Node JS
C621.2	Perform CRUD operations with MongoDB on a huge amount of data.
C621.3	Use various full-stack modules to handle HTTP requests and responses.
C621.4	Develop real time applications using react and Angular components.

LIST OF PROGRAMS

CYCLE 1:

1. Create an application to setup node JS environment and display "Hello World".
2. Create a Node JS application for user login system.
3. Write a Node JS program to perform read, write, and other operations on a file.
4. Create a food delivery website where users can order food from a particular restaurant listed on the website for handling http requests and responses using NodeJS.
5. Implement a program with basic commands on databases and collections using MongoDB.
6. Implement CRUD operations on the given dataset using MongoDB.
7. Perform Count, Limit, Sort, and Skip operations on the given collections using MongoDB.

CYCLE 2:

8. Develop an angular JS form to apply CSS and Events.
9. Develop a Job Registration form and validate it using angular JS.

10. Write an angular JS application to access JSON file data of an employee from a server using $http service.
11. Develop a web application to manage student information using Express and Angular JS Write a program to create a simple calculator Application using React JS.
12. Write a program to create a voting application using React JS
13. Develop a leave management system for an organization where users can apply different types of leaves such as casual leave and medical leave. They also can view the available number of days using react application.
14. Build a music store application using react components and provide routing among the web pages.

REFERENCE BOOKS:

1. Brad Dayley, Brendan Dayley, Caleb Dayley, Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley, 2019.
2. Mark Tielens Thomas., React in Action, 1st Edition, Manning Publications.

Course Code	Course Title	L	T	P	Credits
CS622PE	**Scripting Languages Lab (Professional Elective – III)**	**0**	**0**	**2**	**1**

Course Description This course contains the programs using Ruby, Perl, TCL scripting languages to solve the problems from different domains.

Course Outcomes:
At the end of the course, the student will be able to

C622.1	Script using the features of Perl Script
C622.2	Solve the problems writing the appropriate Ruby Script
C622.3	Apply the constructs of TCL using Tk to write the scripts.
C622.4	Make use of the features of Shell scripts.

CYCLE 1:

1. a) Perl script that determines the largest number among three given numbers.
2. b) Perl script that utilize subroutines to print multiplication tables from 1-10.
3. Implement the following list of manipulating functions using Perl program.
4. a) Shift b) Unshift c) Push
5. a) Develop a Perl script that replaces one word with another in a given string.
6. b) Create a Perl script to check the validity of both IP addresses and email addresses.
7. Develop a Perl script that prints the contents of a file in reverse order, utilizing command line arguments.
8. Write a Ruby script which accept the radius of a circle from the user and compute the parameter and area.
9. Develop a Ruby script that prompts the user for their first and last names, then prints them in reverse order with a space between them.
10. Create a Ruby script that prompts the user to input a filename and then prints out the file extension.
11. Craft a Ruby program to obtain the total marks from a hash containing subject names and their respective marks for a student.

CYCLE 2:

12. Compare and contrast different approaches to calculating factorials in TCL.
13. Compose a TCL script demonstrating comprehension by calculating the product of numbers ranging from 1 to 10.
14. Construct a TCL script illustrating application skills by sorting a list using a custom - defined comparison function.
15. Define TCL script to (i) create a list (ii) append elements to the list (iii) Traverse the list (iv) Concatenate the list.
16. AWK script to find the number of characters, words and lines in a file.
17. AWK script to count the number of lines in a file that do not Contain vowels.
18. Shell script that receives any number of file names as arguments, Check if every argument supplied is a file or directory & report accordingly. Whenever the arguments are supplied as a file the number of lines on it is also reported.
19. Using shell script that displays a list of all files in the current directory to which the user has read, write and execute permissions.

TEXT BOOKS:

1. The World of Scripting Languages, David Barron,Wiley Publications.
2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O'Reilly
3. "Programming Ruby" The Pramatic Progammers guide by Dabve Thomas Second edition

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS623PE	**Internet of Things Lab (Professional Elective – III)**	**0**	**0**	**2**	**1**

Course Description This Lab introduces the programming using Arduino, Raspberry Pi, Node MCU for the data collected from different sensors.

Course Outcomes:
After the end of the course, the student will be able to

C623.1	Inference the impact and challenges posed by IoT networks leading to new architectural models.
C623.2	Compare and contrast the deployment of smart objects and the technologies to connect them to the network.
C623.3	Appraise the role of IoT protocols for efficient network communication.
C623.4	Elaborate Python programming with various interfacing devices using Raspberry PI.
C623.5	Illustrate different sensor technologies for sensing real-world entities and identify the applications of IoT in Industry.
C623.6	Construct a restful web API.

CYCLE 1:

1. Installing OS on Raspberry Pi
   a) Installation using PiImager
   b) Installation using image file
      Downloading an Image
      Writing the image to an SD card
         using Linux
         using Windows
      Booting up Follow the instructions given in the URL
      https://www.raspberrypi.com/documentation/computers/getting-started.html
2. Accessing GPIO pins using Python
   a) Installing GPIO Zero library.
      First, update your repositories list:
      sudo apt update
      Then install the package for Python 3:
      sudo apt install python3-gpiozero
   b) Blinking an LED connected to one of the GPIO pin
   c) Adjusting the brightness of an LED Adjust the brightness of an LED (0 to 100, where 100 means maximum brightness) using the in-built PWM wave length.

3. Displaying the RSS news feed headlines on a LCD display connected to a device using Raspberry pi. (Additional Program)

CYCLE 2:

4. Using raspberry pi
   a) Calculate the distance using a distance sensor.
   b) Basic LED functionality.
5. Using Arduino
   a) Calculate the distance using a distance sensor.
   b) Basic LED functionality.
   c) Calculate temperature using a temperature sensor.
6. Using Arduino

   Using the light sensors, monitor the surrounding light intensity &automatically turn ON/OFF the high intensity LED's by taking some pre-defined threshold light intensity value. (Additional Program)

CYCLE 3:

7. Using Node MCU
   a) Calculate the distance using a distance sensor.
   b) Basic LED functionality.
   c) Calculate temperature using a temperature sensor.
8. Collecting Sensor Data
   a) DHT Sensor interface
      - Connect the terminals of DHT GPIO pins of Raspberry Pi.
      - Import the DHT library using import Adafruit_DHT Read sensor data and display it on screen
9. Collecting Sensor Data in cloud
   a) DHT Sensor interface
      - Connect the terminals of DHT GPIO pins of Raspberry Pi.
      - Import the DHT library using import Adafruit_DHT Read sensor data and display it on screen and cloud (Thing Speak). (Additional Program)

REFERENCE BOOKS:

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015.
2. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014.

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS624PE	**Software Testing Methodologies Lab (Professional Elective – III)**	**0**	**0**	**2**	**1**

Course Description To provide knowledge of software testing methods. To develop skills in automation of software testing and software test automation management using the latest tools.

Course Outcomes:
After completion of this course, the student will be able to

C624.1	Design and develop the best test strategies in accordance with the development model.
C624.2	Design and develop GUI, Bitmap and database checkpoints.
C624.3	Develop database checkpoints for different checks
C624.4	Perform batch testing with and without parameter passing

CYCLE 1:

1. Recording in context sensitive mode and analog mode.
2. GUI checkpoint for single property.
3. GUI checkpoint for single object/window.
4. GUI checkpoint for multiple objects.
5. a. Bitmap checkpoint for object/window.

   b. Bitmap checkpoint for screen area.
6. Database checkpoint for Default check.
7. Database checkpoint for custom check.

CYCLE 2:

8. Database checkpoint for runtime record check.
9. a. Data driven test for dynamic test data submission.

   b. Data driven test through flat files.

   c. Data driven test through front grids.

   d. Data driven test through excel test.

10. Batch testing without parameter passing.

11. Batch testing with parameter passing.

12. Data driven batch.

13. Silent mode test execution without any interruption.

14. Test case for calculator in windows application.

15. Use Protractor tool for end-to-end testing of an Angular application.

TEXT BOOKS:

1. Software Testing techniques, Baris Beizer, 2nd Edition, Dreamtech.
2. Software Testing Techniques – SPD(Oreille)

REFERENCE BOOKS:

1. The craft of software testing, Brian Marick, Pearson Education.
2. Software Testing Tools, Dr. K.V.K.K.Prasad, Dreamtech.
3. Software Testing in the Real World, Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing, Meyers, John Wiley.

BH23 B.Tech. CSE Syllabus	Open Elective	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS600OE	**Problem solving using Data Structures (Open Elective – I)**	3	0	0	3
Pre - Requisite	**Programming for Problem Solving**				

Course Description: This course covers linear and non-linear data structures. It also introduces various sorting techniques and pattern matching algorithms.

Course Outcomes:
At the end of the course, the student will be able to

C600.1	Implement various operations on linear data structures to solve real world problems.
C600.2	Design solutions using Dictionaries and Hash Tables.
C600.3	Implement various kinds operations on trees.
C600.4	Implement Traversing techniques in Graphs.
C600.5	Choose appropriate sorting algorithm.
C600.6	Examine Pattern matching algorithms and Tries.

Unit – I **Introduction to Data Structures**

Introduction to Data Structures, abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks- Operations, array and linked representations of stacks, stack applications, Queues- operations, array and linked representations, Time complexity.

Unit – II **Dictionaries**

Dictionaries: linear list representation, skip list representation, operations - insertion, deletion and searching.
Hash Table Representation: hash functions, collision resolution-separate chaining, open addressing, linear probing, and quadratic probing, double hashing, rehashing, extendible hashing.

Unit – III **Trees**

Trees: Terminology, Representation, Binary Tree - Implementation, Operations- Searching, Insertion and Deletion, Traversals.
Binary Search Trees: Definition, Implementation, Operations- Searching, Insertion and Deletion.

Unit – IV **Graphs and Sorting**

Graphs: Terminology**,** Implementation Methods. Traversal Methods.
Sorting: Quick Sort, Heap Sort, External Sorting- Model for external sorting, Merge Sort.

Unit – V	Pattern Matching and Tries
Pattern Matching and Tries: Pattern matching algorithms-Brute force, Boyer – Moore algorithm, Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, and Suffix tries.	

TEXT BOOKS:

1. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson Freed, Universities Press.
2. Data Structures using C – A. Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson Education.

REFERENCE BOOK:

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B.A. Forouzan, Cengage Learning.

BH23 B.Tech. CSE Syllabus	Open Elective	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS601OE	**Introduction to Java Programming (Open Elective – III)**	3	0	0	3
Pre - Requisite	**-**				

Course Description: This course covers Object oriented thinking, basics of java. Exceptions, multi threading, exploring various utilities and handling data base & file management**.**

Course Outcomes:
At the end of the course, the student will be able to:

C601.1	Understand Object Oriented Programming concepts.
C601.2	Apply the basic constructs, Inheritance, packages and Interfaces.
C601.3	Make use of concepts of Exceptions to handle run time errors.
C601.4	Implement the concepts of multi threading.
C601.5	Utilize collection framework to implement various data structures
C601.6	Develop standalone applications using JDBC & file management.

Unit – I **Object oriented thinking and Java Basics**

Object oriented Thinking and Java Basics - Class Hierarchies - Inheritance, Summary of OOP concepts.
History and Evolution of Java, Java buzzwords, data types, variables and arrays, operators, expressions, control statements, type conversion and casting, introducing classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, method binding, inheritance, overriding and exceptions, argument passing, recursion, nested and inner classes. String handling.

Unit – II **Inheritance, Packages and Interfaces**

Inheritance, Packages and Interfaces – Inheritance basics, using super, muliti level hierarchy, using final with inheritance, polymorphism- method overriding, abstract classes, the Object class.
Defining, Creating and Accessing a Package, Understanding CLASS PATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces.

Unit – III **Exception handling and Multi threading**

Exception handling and Multi threading-- exception handling fundamentals, exception types, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception subclasses. Differences between multi threading and multitasking, thread life cycle, creating threads, thread priorities, synchronization, inter thread communication, thread groups, daemon threads. Enumerations, auto boxing, annotations, generics.

Unit – IV	Collections Framework
The Collections Framework (java.util) - Collection Interfaces, List, Set, Queue, Deque, collection classes, Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Accessing a Collection via an Iterator, Using an Iterator, The For-Each alternative, Map Interfaces and Classes, Comparators, Collection algorithms, Arrays, Dictionary, Hashtable, Properties. More Utility classes, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter, Scanner.	
Unit – V	**JDBC & File Handling**
Connecting to Database-JDBC drivers, connecting to a Data base, preparing SQL statements, Querying a data base and processing the results, updating data with JDBC. **Exploring java.io:** streams- byte streams, character streams, Console class and sterilization.	

TEXT BOOKS:

1. Java the complete reference, 7th edition, Herbert Schildt, TMH.
2. Understanding OOP with Java, updated edition, T. Budd, Pearson education.

REFERENCE BOOK:

1. An Introduction to programming and OO design using Java, J.Nino and F.A. Hosch, John wiley & sons.
2. An Introduction to OOP, third edition, T. Budd, Pearson education.
3. Introduction to Java programming, Y. Daniel Liang, Pearson education.
4. An introduction to Java programming and object-oriented application development, R.A. Johnson- Thomson.
5. Object Oriented Programming with Java, R.Buyya, S.T.Selvi, X.Chu, TMH.
6. Core Java (an Integrated approach), R.Nageswara Rao , DreamTech Press

BH23 B.Tech. CSE Syllabus	B. Tech. III Year II Sem	BVRITHCEW			
Course Code	**Course Title**	**L**	**T**	**P**	**Credits**
CS602OE	**Fundamentals of AI (Open Elective – III)**	3	0	0	3
Pre - Requisite	**Data Structures**				

Course Description: This course contains diverse branches of AI through a discussion of its theoretical foundations. At the end of the course the students shall have in-depth understanding of different knowledge representation formalisms and various techniques used for reasoning and learning.

Course Outcomes:

At the end of the course, the student will be able to

C602.1	Identify suitable search agents for problem solving.
C602.2	Apply search techniques to solve problems in various domains
C602.3	Make use of mathematical logic to represent and reason to logical conclusions.
C602.4	Demonstrate/ Illustrate knowledge schemes for various domains and reason under uncertainty.
C602.5	Apply learning concepts and techniques to real-world problem-solving scenarios
C602.6	Acquire knowledge acquisition techniques and methodologies for extracting domain knowledge from expert systems.

Unit – I

Introduction: AI History, Overview of AI application areas, AI problems, Agents and Environments, Structure of Agents, Problem Solving Agents.

Basic Search Strategies: Problem Spaces, Uninformed Search (Breadth-First, Depth-First Search, Depth-first with Iterative Deepening), Heuristic Search (Hill Climbing, Generic Best-First, A*), Constraint Satisfaction (Backtracking, Local Search).

Unit – II

Advanced Search: Constructing Search Trees, Stochastic Search, Minimax Search, Alpha-Beta Pruning.

Basic Knowledge Representation: Propositional Logic, First-Order Logic, Forward Chaining and Backward Chaining.

Unit – III

Advanced Knowledge Representation and Reasoning: Knowledge Representation Issues, Non-monotonic Reasoning, Other Knowledge Representation Schemes.

Reasoning Under Uncertainty: Basic probability**,** Introduction to Probabilistic Reasoning, Bayes Theorem, Acting Under Uncertainty, Representing Knowledge in an Uncertain Domain, Bayesian Networks.

Unit – IV	
Learning: What Is Learning? Rote Learning, Learning by Taking Advice, Learning in Problem Solving, Learning from Examples, Winston's Learning Program, Decision Trees.	
Unit – V	
Expert Systems: Representing and Using Domain Knowledge, Shell, Explanation, Knowledge Acquisition.	

TEXT BOOKS:

1. Artificial Intelligence: A Modern Approach, Fourth Edition, Stuart Russell and Peter Norvig, Pearson Education. 2021
2. Artificial Intelligence and Expert systems – Patterson, Pearson Education

REFERENCE BOOKS:

1. Artificial Intelligence, Elaine Rich, Kevin Knight, Shivasankar B. Nair, The McGraw Hill publications, Third Edition, 2009.
2. George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education, 6th ed., 2009.
3. D. Poole and A. Mackworth. Artificial Intelligence: Foundations of Computational Agents, Cambridge University Press, 2010.