SANT GADGE BABA AMRAVATI UNIVERSITY, AMRAVATI

Draft Syllabus for Third Year Undergraduate Degree Programme in

Engineering and Technology

B.E. [INFORMATION TECHNOLOGY]

w.e.f. From the Year: 2026-2207 & onwards

Draft Syllabus of Semester-V&VI B.E. [INFORMATION TECHNOLOGY]

Course Code:5IT220PC	**Course Title: Database Management Systems**	**L:3 T:0 P:0 C:3**
Course Prerequisite:	Discrete Mathematics, Data Structures and Algorithm	
Course Objectives:	1. To understand the fundamental concepts of database management system and its applications. 2. To learn database query languages. 3. To Understand concept of designing database schema and its mapping to relational table. 4. To Apply the concepts of database integrity and Normalization. 5.To Understand the concept of transaction management and its properties. 6. To Understand concept of concurrency control and various types of protocol.	
Course Outcomes (Expected Outcome):	On completion of the course, the students will be able to 1. Understand the database terminology, Design entity relationship and convert entity relationship diagrams into RDBMS. 2. Understand relational algebra, relational databases and formulate SQL queries on the database. 3. Apply constraints and normalization on databases. 4. Apply ACID properties for transaction management and concurrency control. 5. Understand the concept of Concurrency control and study of various database protocols. 6. Understand Database security	
Unit I:	Unit Title: Introduction to DBMS	Hours:7
Database System Applications, Purpose of database systems, View of Data, Database Languages, Database Architecture, Database Users and Administrators, Entity- Relationship Model, Constraints, Removing redundant attributes in Entity sets, E-R diagrams, Reduction to Relational Schemas, E-R design issues, Extended E-R Features		
Unit II:	Unit Title: Relational Data Model and Normalization	Hours:7
Relational Model: Structure of Relational Databases, Database schema, keys, schema diagram, relational query languages, relational operators, The Relational Algebra, Normalization: functional dependencies, Decomposition, Domain & data dependency, types of Normal forms: 1NF, 2NF, 3NF, BCNF, 4NF, 5NF;		
Unit III:	Unit Title: SQL Introduction	Hours:7
Basics of SQL, data types, DDL, DML , create, drop, Alter command, defining constraints – Primary key, unique, not null, check, IN operator ,Query with Select statements, Insert statement, Update statement, Delete statement with use of where, and, or clauses, Use of like and top clause, Distinct Command, Operators:- Relational, Arithmetic, Logical, Set operators, wildcard operators,.Functions - aggregate functions, string functions,Use of group by, having, order by, join and its types View in SQL, Create view, drop view.		
Unit IV:	Unit title :Transaction Management	Hours:7
Transaction Concept, Simple transaction model, Storage structure, Transaction Atomicity and Durability, transaction isolation, Serializability, transaction isolation and atomicity, transaction isolation levels, Implementation of Isolation levels, Transactions as SQL statements		
Unit V:	Unit Title: Concurrency Control	Hours:7
Lock-Based Protocols, Deadlock Handling, Multiple Granularities, Timestamp- Based Protocols, Validation Based Protocols, Multi-version schemes		
Unit VI:	Unit Title: Database Security	Hours:7
Authentication, Authorization and access control, DAC, Mandatory Access Control and Role- Based Access Control models, Intrusion detection, SQL injection.		
Text Book: Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "**Database System Concepts"**, Sixth Edition, Mc Graw Hill.		
Reference Books: 1. Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, McGraw-Hill 2. Shamkant B. Navathe, RamezElmasri, Database Systems, Pearson Higher Education 3. Garcia-Molina, Ullman, Widom: Database System Implementation, Pearson education. 4. S. K. Singh: Database Systems, Concepts, Design and Applications, Pearson Education. 5. G.K. Gupta: Database Management Systems, McGraw Hill. 6. Toledo and Cushman: Database Management Systems, (Schaum's Outlines)		

Course Code: 5IT221PC	**Course Title: Operating Systems**	**L:3 T:0 P:0 C:3**
Course Prerequisite:	Basic knowledge of computer organization, data structures, and programming concepts (C/Java) is required to understand operating system principles.	
Course Objectives:	1. To introduce the fundamental concepts, structure, services, and evolution of operating systems. 2. To explain process management concepts including processes, threads, and CPU scheduling mechanisms. 3. To analyze synchronization issues, interprocess communication, and deadlock handling techniques. 4. To describe memory management techniques including paging, segmentation, and virtual memory. 5. To explain file system organization, implementation techniques, and storage management. 6. To develop the ability to evaluate operating system policies and mechanisms for performance and protection.	
Course Outcomes (Expected Outcome):	On completion of the course, students will be able to: Define and explain the role, components, services, and evolution of operating systems. (L1) 1. Apply process and thread concepts to analyze CPU scheduling algorithms and compute performance metrics. (L3) 2. Analyze synchronization problems and demonstrate solutions using semaphores, monitors, and deadlock handling methods. (L4) 3. Apply memory management techniques such as paging, segmentation, and virtual memory to solve allocation problems. (L3) 4. Explain file system structures, allocation methods, free-space management, and recovery mechanisms. (L2) 5. Evaluate disk scheduling algorithms and system protection mechanisms to improve system performance and security. (L5)	
Unit I:	Unit Title: **Introduction to Operating System**	Hours: 7
Introduction: Operating System definition, Operating System Evolution, Components and Services, Process Concept, Process Scheduling, Operations on Processes, Cooperating Processes, Inter-process Communication, Threads Overview, Multithreading Models, Threading Issues, Java Threads.		
Unit II:	Unit Title: **Process Scheduling**	Hours:7
Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non-pre-emptive, FCFS, SJF, RR, Priority, Multilevel Queue, Multilevel Feedback Queue Scheduling.		
Unit III:	Unit Title: **Process Synchronization**	Hours:7
Process Synchronization Basics: The Critical-Section Problem, Synchronization Hardware, Semaphores, Monitors, Deadlocks: Definition & Characterization, Deadlocks Prevention, Avoidance, Detection and Recovery from Deadlock.		
Unit IV:	Unit Title: **Memory Management**	Hours:7
Memory Management Background, Swapping, Contiguous Memory Allocation Schemes, Paging, Segmentation, Virtual Memory Management: Background, Demand paging scheme, Process Creation, Page Replacement Policies, Allocation of Frames, Thrashing.		
Unit V:	Unit Title: **File System**	Hours:7
File-System Interface; Directory Structure, File-System Mounting, File Sharing &Protection, File System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management. File Recovery.		
Unit VI:	Unit Title: **Storage management**	Hours:7
Mass-Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, System Protection: Goals, Principles, Domain of Protection, Access Matrix, Access Control.		
Text Book: Avi Silberschatz, P. B. Galvin, G. Gagne: "Operating System Concepts" (10/e) John-Wiley & Sons		

Reference Books:

1. A. S. Tanenbaum "Modern Operating Systems" Pearson Education.
2. William Stallings "Operating Systems" Prentice-Hall.
3. D. M. Dhamdhere "Operating Systems" Tata McGraw-Hill.
4. P. Balkrishna Prasad: "Operating Systems" Scitech Publications (I) Pvt

Course Code: 5IT222PC	**Course Title: Theory of Computation**	**L:3 T:0 P:0 C:3**
Course Prerequisite:	Students should have knowledge on: • Discrete Mathematics (sets, relations, functions, logic) • Fundamentals of Computer Programming • Basic Data Structures and Algorithms	
Course Objectives:	1. Introduce fundamental concepts of formal languages, automata, and computational models. 2. Provide a theoretical foundation for understanding finite automata, regular expressions, and grammars. 3. Explain the role of context-free grammars and pushdown automata in language processing. 4. Develop an understanding of Turing machines and their computational power. 5. Analyze decidable and undecidable problems in computation. 6. Introduce recursive function theory and its relation to computability.	
Course Outcomes (Expected Outcome):	1. Define and analyze alphabets, languages, and finite automata, and design deterministic and nondeterministic finite automata for given problems. 2. Construct and manipulate regular expressions and regular grammars, and demonstrate equivalence among regular expressions, finite automata, and regular grammars. 3. Explain context-free grammars and pushdown automata, and perform conversions between CFGs and PDAs for given languages. 4. Design Turing machines for simple computational problems and explain the concept of computability and recursive enumerable languages. 5. Distinguish between decidable and undecidable problems and analyze the computational limits of algorithms using Turing machine models. 6. Describe linear bounded automata, context-sensitive languages, and fundamental concepts of recursive function theory, including primitive recursive and recursive functions.	
Unit I:	Unit Title: Finite State Machines and Finite Automata	Hours: 7
Alphabet, Language, Operations, Finite State Machine, Definitions, Finite Automation Model, Acceptance of Strings and Languages, Non Deterministic Finite Automata, Deterministic Finite Automata, Equivalence between NFA and DFA, Conversion of NFA into DFA, Equivalence between two FSM's, Moore and Mealy Machines, Moore and Mealy Machines Conversion		
Unit II:	Unit Title: Regular Expression and Regular Grammar	Hours: 7
Regular Sets, Regular Expressions, Identity Rules, Manipulation of Regular Expressions, Equivalence between RE and FA, Inter Conversion, Pumping Lemma, Closure Properties of Regular Sets (proofs not required), Chomsky hierarchy of languages, Regular Grammars, Right Linear and Left Linear Grammars, Equivalence between Regular Linear Grammar and FA, Inter conversion between RE and RG		
Unit III:	Unit Title: Context Free Grammer and Pushdown Automata	Hours: 7
Context Free Grammar, Derivation trees, Chomsky Normal Form, Greibach Normal Form, Push down automata, Definition, Model, Acceptance of CFL, Equivalence of CFL and PDA, Inter conversion between CFL and PDA, Enumeration of properties of CFL (proofs omitted).		
Unit IV:	Unit Title: Turing Machines	Hours: 7
Turing Machine, Definition, Model, Design of TM, Computable Functions, TM Design Problems, Recursive enumerable language, Church's hypothesis, Types of TM's		
Unit V:	Unit Title: Linear Bounded Automata and Decidable Undecidable Problems	Hours: 7
Linear bounded automata, Context Sensitive Language, Decidability of problems, Undecidability		
Unit VI:	Unit Title: Recursive Function Theory	Hours: 7
Properties of recursive enumerable languages, Properties of non-recursive enumerable languages, Universal Turing Machine, Post correspondence problem, Introduction to recursive function theory Primitive recursive functions, Recursive functions.		

Text Book: K.L.P. Mishra and N. Chandrasekaran, Theory of Computer Science (Automata, Languages and Computation), PHI Learning
Reference Books: 1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation Pearson Education 2. Michael Sipser, Introduction to the Theory of Computation, Cengage Learning 3. J.C. Martin, Introduction to Languages and the Theory of Computation, McGraw-Hill 4. Peter Linz, An Introduction to Formal Languages and Automata, Narosa / Jones & Bartlett

Course Code: 5IT223PE	**Course Title: Program Elective Course -I** **(i) Information Security Systems**	**L-3, T-0, P-0, C-3**
Course Prerequisite:	Data Communication and Networking, Cyber security	
Course Objectives:	1. To understand the basics of Information Security 2. To know the legal, ethical and professional issues in Information Security 3. To know the aspects of risk management 4. To become aware of various standards in this area 5. To understand & apply the technological aspects of Information Security	
Course Outcomes (Expected Outcome):	1. Study the foundational theory behind information security. 2. Discuss the basic information security. 3. Illustrate the legal, ethical and professional issues. 4. Discuss the aspects of risk management. 5. Summarize various standards for information security. 6. Explain the security techniques.	
Unit I:	**Introduction to Information Security**	**Hours: 7**
History, What is Information Security? Critical Characteristics of Information, NSTISSC Security Model, Components of an Information System, Securing the Components, Balancing Security and Access, The SDLC, The Security SDLC		
Unit II:	**Security Investigation**	**Hours: 7**
Need for Security, Business Needs, Threats, Attacks, Legal, Ethical and Professional Issues.		
Unit III:	**Legal, Ethical, and Professional Issues in Information Security**	**Hours: 7**
Law and Ethics in Information Security, International Laws and Legal Bodies, Ethics and Information Security.		
Unit IV:	**Security Analysis**	**Hours: 7**
An Overview of Risk Management, Risk Identification, Risk Assessment, Risk Control Strategies.		
Unit V:	**Planning for Security**	**Hours: 7**
Information Security Planning and Governance. Information Security Policy, Standards, and Practices, the Information Security Blueprint, Security Education, Training, and Awareness Program. Continuity Strategies .		
Unit VI:	**Physical Security and Implementing Information Security Project**	**Hours: 7**
Introduction, Physical Access Controls, Fire Security and Safety, Failure of Supporting Utilities and Structural Collapse, Interception of Data, Mobile and Portable Systems, Implementing Information Security Project management.		
Text Book: Michael E Whitman and Herbert J Mattord, “Principles of Information Security”, Vikas Publishing House, New Delhi, 2003.		
Reference Books: 1. Micki Krause, Harold F. Tipton, “Handbook of Information Security Management”, Vol 1-3 CRC Press LLC, 2004.111 2. William Stallings, “Cryptography and Network Security: Principles and Practice", 6th Edition, Prentice Hall 3. M. Stamp, “Information Security: Principles and Practice,” 2nd Edition, Wiley, ISBN: 0470626399, 2011. 4. Nina Godbole, “Information Systems Security”, Wiley India Pvt. Ltd, ISBN -978-81-265-1692-6 **5.** Mark Merkow, “Information Security-Principles and Practices”, Pearson Ed. 978-81-317-1288- 7.		

<table>
<tr><td>Course Code: 5IT223PE</td><td>Course Title: Program Elective Course -I (ii) Data Science & Statistics</td><td>L:3 T:0 P:0 C:3</td></tr>
<tr><td>Course Prerequisite:</td><td colspan="2">Students should have:
Basic knowledge of programming concepts (variables, loops, functions)
Fundamental understanding of mathematics (algebra and basic probability)
Introductory knowledge of statistics (mean, variance, probability concepts)</td></tr>
<tr><td>Course Objectives:</td><td colspan="2">Introduce Python programming fundamentals and libraries used in Data Science.
Develop the ability to perform statistical analysis using Python.
Enable students to explore, visualize, and summarize data effectively.
Provide a strong foundation in statistical experiments and hypothesis testing.
Familiarize students with regression and classification techniques.
Build practical understanding of machine learning models for prediction and decision-making.</td></tr>
<tr><td>Course Outcomes
(Expected Outcome):</td><td colspan="2">Apply Python programming and statistical concepts to analyze real-world datasets using NumPy and Pandas.
Perform exploratory data analysis (EDA) using visualization tools such as Matplotlib and Seaborn to identify patterns and trends in data.
Conduct statistical experiments and hypothesis testing, including t-tests, ANOVA, and chi-square tests, and interpret statistical significance.
Build and evaluate regression models, including simple and multiple linear regression, while understanding model assumptions and validation techniques.
Develop and assess classification models using logistic regression and performance metrics such as confusion matrix, ROC curve, and AUC.
Implement decision tree and random forest algorithms for prediction tasks and analyze model behavior and performance.</td></tr>
<tr><td>Unit I:</td><td>Unit Title: Python for Data Science</td><td>Hours: 07</td></tr>
<tr><td colspan="3">Mean, Median, Mode, Variance, Standard Deviation Numpy: The Basics of NumPy Arrays, Universal Functions, Aggregators, Broadcasting, Fancy Indexing; Pandas: Introducing Pandas Objects, Operating on Data in Pandas, Handling Missing Data, Hierarchical Indexing, Combining Datasets: Concat, Append, merge and join, aggregation and grouping , pivot Tables</td></tr>
<tr><td>Unit II:</td><td>Unit Title: Exploratory Data Analysis and Statistics</td><td>Hours: 07</td></tr>
<tr><td colspan="3">EDA: Matplotlib and Seaborn: Simple Line Plots, Simple Scatter Plots, Density and Contour Plots, Histograms, Binnings, and Density, Random Sampling, Distributions: Uniform Distribution, Normal Distribution, Poisson’s Distribution, Binomial Distribution</td></tr>
<tr><td>Unit III:</td><td>Unit Title: Statistical Experiments and Significance Testing</td><td>Hours: 07</td></tr>
<tr><td colspan="3">Hypothesis Test: The Null Hypothesis, Alternative Hypothesis, One way, Two way Hypothesis Test; Statistical Significance and P-Values: P-value, alpha, type 1 error , type 2 error; t-Tests, Degrees of Freedom, ANOVA: F statistics, Two-way Anova; Chi-Square Test: A Resampling Approach</td></tr>
<tr><td>Unit IV:</td><td>Unit Title: Regression Techniques</td><td>Hours: 07</td></tr>
<tr><td colspan="3">Introduction to Machine Learning, Hyper parameter and Model Validation, Feature engineering, Assumptions in Regression, Simple Linear Regression, Multiple Linear Regression</td></tr>
<tr><td>Unit V:</td><td>Unit Title: Classification: Logistic regression</td><td>Hours: 07</td></tr>
<tr><td colspan="3">Logistic Response Function and Logit, Predicted Values from Logistic Regression, Interpreting the Coefficients and Odds Ratios; Evaluating Classification Models: Confusion Matrix, Precision, Recall, and Specificity, ROC Curve, AUC</td></tr>
<tr><td>Unit VI:</td><td>Unit Title: Decision Tree and Radom Forest</td><td>Hours: 07</td></tr>
<tr><td colspan="3">A Simple Example, The Recursive Partitioning Algorithm, Measuring Homogeneity or Impurity, Stopping the Tree from Growing, Predicting a Continuous Value; Random Forest</td></tr>
<tr><td colspan="3">Text Book: Practical Statistics for Data Scientists By Peter Bruce, Andrew Bruce, O'Reilly Media.</td></tr>
</table>

Reference Books:

1. Python Data Science Handbook By Jake Vander Plas O'Reilly Media.
2. Introduction to Machine Learning with Python By Andreas C. Müller, Sarah Guido, O'Reilly Media
3. Think Stats By Allen B. Downey O'Reilly Media

Course Code: 5IT223PE	**Course Title: Program Elective Course -I (iii) Internet of Things**	**L:3 T:0 P:0 C:3**
Course Prerequisite:	1. To know basic computer networking. 2. To understand basic electronics and sensors. 3. To have basic programming knowledge. 4. To be familiar with Arduino and Raspberry Pi. 5. To know the basics of cloud computing.	
Course Objectives:	1. To understand the fundamentals of IoT 2. To analyze the transition from M2M to IoT 3. To learn the design principles and reference architectures of IoT 4. To explore various architectural views and technical constraints of IoT 5. To develop practical IoT solutions 6. To explain security, privacy, and trust issues in IoT systems.	
Course Outcomes (Expected Outcome):	1. To explain the basic components and functions of IoT systems. 2. To distinguish between M2M and IoT technologies. 3. To design simple IoT systems using standard architecture. 4. To assess IoT systems based on architecture and constraints. 5. To develop IoT applications using hardware and software tools. 6. To develop secure and reliable IoT applications.	
Unit I:	Unit Title: Introduction to IoT:	Hours:07
Sensing, Actuation, Networking basics, Communication Protocols, Sensor Networks, Machine-to-Machine Communications, IoT Definition, Characteristics. IoT Functional Blocks, Physical design of IoT, Logical design of IoT, Communication models & APIs.		
Unit II:	Unit Title: M2M to IoT	Hours:07
From M2M to IoT, M2M towards IoT-the global context, A use case example, Differing Characteristics. Definitions, M2M Value Chains, IoT Value Chains.		
Unit III:	Unit Title: M2M vs IoT An Architectural Overview	Hours:07
Main design principles and needed capabilities, An IoT architecture outline, standards considerations. Reference Architecture and Reference Model of IoT.		
Unit IV:	Unit Title: IoT Reference Architecture:	Hours:07
Getting Familiar with IoT Architecture, Various architectural views of IoT such as Functional, Information, Operational and Deployment, Constraints affecting design in IoT world- Introduction, Technical design Constraints.		
Unit V:	Unit Title: Developing IoT solutions:	Hours:07
Introduction to Python, Introduction to different IoT tools, Introduction to Raspberry Implementation of IoT with Arduino and Raspberry, Cloud Computing, Fog Computing, Connected Vehicles,		
Unit VI:	Unit Title: Security, Privacy & Trust	Hours:07
IoT security challenge, Spectrum of security considerations, Unique security challenges of IoT devices, Internet of things privacy background, Unique privacy aspects of internet of things, Trust for IoT.		
Text Book: Vijay Madisetti and Arshdeep Bahga, "Internet of Things (A Hands-on-Approach)", First Edition, VPT, 2014		
Reference Books: 1. Peter Waher, "Learning Internet of Things", PACKT publishing, BIRMINGHAM – MUMBAI 2. "Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems", Ovidiu Vermesan, Peter Friess, River Publishers. 3. Bernd ScholzReiter, Florian Michahelles, "Architecting the Internet of Things", ISBN 978-3-642-19156-5 e-ISBN 978-3-642-19157-2, Springer.		

Course Code: 5IT224PC	**Course Title: Database Management Systems Lab**	**L:0 T:0 P:2 C:1**
Course Prerequisite:	Basic concept of programming, Basic concepts of data structures	
Course Objectives:	1. To study the ER model which provides a high level view of the issues in database design, to capture the semantics of realistic applications within the constraints of a data model. 2. To study the primary data model (relational model) for commercial data processing applications. 3. To study the standard structured query language and retrieve the information from the database in various ways. 4. To study the integrity and security constraints of the database by enforcing constraints.	
Course Outcomes(Expected Outcome):	1. To design ER model for any kind of application. 2. To design and develop database. 3. To apply normalization. 4. To query the database. 5. To apply various integrity constraints 6. To build indices, views	

List of Experiments :

1. 10 Experiments based on syllabi, with at least 01 beyond syllabi .
2. One study Practical (Practical No. 18) and a mini project

1. **Practical 1: To Study a Database Modeling Tool.**

Study of Data Modeling Tools:

- Take a description of the enterprise, create its corresponding ER Diagram and build a database model using any modeling tool. The following basic features of the modeling should be covered while building the model:
  - Logical / Physical Modeling
  - Adding an entity / its attributes , relationships (all kinds of relationships viz., parent-child, foreign key references, one to many, many to many etc)
  - Forward / reverse engineering
  - Details of forward engineering / schema generation
  - Steps to generate the schema

2. **Practical 2: To Study and implement DDL Commands**
   Implement the model created in Practical 1, in any of the DBMS like Oracle, MySQL, or Microsoft SQL Server database software.
   - Creating the proper tables
   - Insert the data into it.
   - Study Dropping and Altering the Tables. Study the cascaded deletes.

3. **Practical 3: To Study and implement DML Commands-I**
   - SQL queries : Write and execute different SQL queries
   - Execute Simple queries using SELECT, FROM, WHERE clauses,
   - In Where clause use different predicates involving OR,AND, NOT
   - Rename operation
   - Tuple Variables
   - Write SQL for various String operations (%,_,*)
     - Match beginning with
     - Match ending with
     - Substring
     - Match exactly n characters
     - Match at least n characters
   - Sort the output of the query using **Order by**
   - Write SQL using **Having**

4. **Practical 4** : **To Study and implement DML Commands-II**
   Write SQL queries and perform
   - Set membership operations

- In, not in
- Some
- All
- Exists and not exists, Test for emptyness using exists, not exists
- Test for absence of duplicates.
- Nested queries

5. **Practical 5.** Study and implement aggregation functions.

   Write different queries using following Aggregate functions

   a. Min (minimum 3 SQL queries)
   b. Max (minimum 3 SQL queries)
   c. Avg (minimum 3 SQL queries)
   d. Sum (minimum 3 SQL queries)
   e. Count (minimum 3 SQL queries)

6. **Practical 6: Write SQL to create Views and Indexes**.

7. **Practical 7: Write SQL to perform the modifications to the database**

8. **Practical 8 : PL /SQL**

9. **Practical 9 : Database Access Using Cursors**

   Write a trigger to find the names and cities of customers who have more than xyz in any account.

10. **Practical 10 : Triggers**

    - Write a trigger for dealing with the overdrafts (set the account balance to zero, and creating a loan in the amount of the overdraft. Keep account number as loan number in the loan table)
    - Write a trigger for dealing with blank cities (set the city field to null when it is blank)

11. **Practical 11: Procedures, functions**
    - Write at least 2 functions, and demonstrate its use
    - Write at least 2 procedures, and demonstrate its use

12. **Practical 12 : Web Programming with PL/SQL. (Contents Beyond Syllabus)**
    HTTP, A Simple Example., Printing HTML Tables., Passing Parameters., Processing HTML Forms., Multi-Valued Parameters.

13. **Practical 13: Develop a JDBC Applications, Retrieve the information by connecting to the database using a host language (JAVA, C, C++) (Contents Beyond Syllabus)**

14. **Practical 14: Web Programming with Java Servlets. (Connecting to the database) (Contents Beyond Syllabus)**
    A Simple Servlet., HTTP Servlet API Basics, HTML Form Processing in Servlets.

15. **Practical 15: PHP : Develop a simple application to access the database using PHP (Contents Beyond Syllabus)**

16. Study of Open Source NoSQL Databases

17. **Based on the concepts covered in text create a Mini Project:**

Suggested Topics:

i. Bank database (Given in Korth book)
ii. University Database (Given in Korth book)
iii. Airline Flight Information System.
iv. Library Database Application.
v. University Student Database.
vi. Video Chain Database.
vii. Banking Database.
viii. BiBTeX Database.
ix. Music Store Database.
x. Online Auctions Database.
xi. A Web Survey Management System.

18. **Study Practical:**

To study and document Big Data architecture and technologies covering:

a. Big Data and its characteristics (3 V's)
b. Big Data processing solutions
c. NoSQL databases and data models

d. MongoDB as a NoSQL database
e. CAP theorem and BASE properties
f. Comparative analysis of SQL and NoSQL database

Text Books:

1. Korth, Sudarshan, Silberschatz, Database System Concept, Mc-Graw Hill
2. Mysql Reference Manual (for Mysql database)

Reference Books:

1. Kevin Roebuck, “Storing and Managing Big Data - NoSQL, HADOOP and More”, Emereopty Limited, ISBN: 1743045743, 9781743045749
2. Kristina Chodorow, Michael Dirolf, “MangoDB: The Definitive Guide” ,O‟Reilly Publications, ISBN: 978-1-449-34468-9.
3. Adam Fowler, “NoSQL For Dummies”, John Wiley & Sons, ISBN-1118905628
4. C J Date, “An Introduction to Database Systems”, Addison-Wesley, ISBN: 0201144719

Course Code: 5IT225PC	**Course Title: Operating Systems Lab**	**L:0 T:0 P:2 C:1**
Course Prerequisite:	Basic knowledge of computer organization, data structures, and programming concepts (C/Java) is required to understand operating system principles.	
Course Objectives:	1. Familiarize students with UNIX/Linux operating systems, their architecture, shells, and command-line environment. 2. Provide hands-on experience with Unix shell programming, environment variables, and open-source operating systems (Ubuntu). 3. Enable practical understanding of process creation, CPU scheduling, and process synchronization mechanisms. 4. Develop skills to implement memory management, file system organization, and disk scheduling algorithms. 5. Strengthen analytical and programming skills through simulation of core operating system concepts.	
Course Outcomes (Expected Outcome):	After successful completion of this course, the student will be able to: 1. Explain the fundamentals of UNIX/Linux operating systems, including system organization, shell types, and differences between UNIX and Linux. 2. Use UNIX/Linux shell commands and environment variables effectively to perform system-level operations on Ubuntu. 3. Implement process creation and CPU scheduling algorithms such as FCFS, SJF, and Priority scheduling. 4. Apply process synchronization techniques using semaphores to solve problems like the Producer–Consumer problem. 5. Design and implement memory management and file system techniques, including First Fit, Best Fit, directory organization, and file mounting. 6. Implement and analyze disk scheduling algorithms such as FCFS, SCAN, and LOOK to evaluate system performance.	

Suggested List of Experiments:

1) To study ...
   a. UNIX
   b. Difference between LINUX and UNIX
   c. UNIX system organization
   d. Types of shell
2) Study of Unix Shell and Environment Variables
3) To study and execute basic commands related to open-source Ubuntu OS,
4) Write a program for parent process creating child process by using fork system call.
5) To simulate First Come First Serve process scheduling algorithm
6) To simulate Shortest Job First process scheduling algorithm
7) To implement Priority Based Process Scheduling Algorithm
8) To simulate producer-consumer problem using semaphores
9) To implement following memory management techniques: FIRST Fit
10) To implement following memory management techniques: Best Fit
11) Simulate File Mounting and Unmounting
12) To simulate the Single level directory file organization technique
13) To implement FCFS Disk Scheduling algorithm
14) To implement SCAN Disk Scheduling algorithm
15) To implement LOOK Disk Scheduling algorithm

Course Code: 5IT226PE	**Course Title: Program Elective Course -I** **(i) Information Security Systems Lab**	**L-0, T-0, P-2, C-1**
Course Prerequisite:	Programming concepts, Data Communication and Networking, Cyber security, Basic Linux/Windows commands and Operating System	
Course Objectives:	1. To provide hands-on experience with fundamental and advanced information security techniques. 2. To enable students to understand cryptographic algorithms and their real-world implementation.	

	3. To familiarize students with system security, network security, and web security tools. 4. To develop skills to identify vulnerabilities and apply suitable security controls. 5. To enhance practical knowledge of ethical hacking, malware analysis, and secure communication.
Course Outcomes (Expected Outcome):	**CO1:** Implement classical and modern cryptographic algorithms for secure data transmission. **CO2:** Apply authentication, authorization, and access control mechanisms in systems. **CO3:** Analyze system and network vulnerabilities using security tools. **CO4:** Implement security mechanisms for operating systems, databases, and web applications. **CO5:** Demonstrate the use of ethical hacking and penetration testing techniques responsibly. **CO6:** Evaluate security threats and recommend suitable countermeasures.

Minimum eight experiments / programming assignments must be completed based on the syllabus uniformly covering each of the units. Indicative or Sample Experiments list is as follows:

Sample Experiments List:

1. Study of Network Security fundamentals – Etherical Hacking, Social Engineering practices.
2. Study of Information Security concepts, security attacks, services, and mechanisms.
3. System threats attacks – Denial of Services.
4. Digital Signature generation and verification.
5. Implementation of Message Authentication Codes (MAC).
6. Secure file transfer using encryption techniques.
7. Study and configuration of Firewall in Linux/Windows.
8. Study of Intrusion Detection System (IDS) tools.
9. Vulnerability assessment using tools such as Nmap.
10. Study of Web Application Security and common attacks (SQL Injection, XSS).
11. Password cracking and protection techniques.
12. Study of Malware types and basic malware analysis.
13. Implementation of Hash Functions MD5 / SHA-1 / SHA-256
14. Implementation of Asymmetric / **Symmetric Key Cryptography** DES / AES
15. Mini Project / Case Study on Information Security application.

Text Books:

1. William Stallings, “Cryptography and Network Security: Principles and Practice”, Pearson Education.
2. Nina Godbole, “Information Systems Security”, Wiley India.
3. Behrouz A. Forouzan, “Cryptography and Network Security”, McGraw-Hill.

Reference Books:

1. Charles P. Pfleeger, Shari Lawrence Pfleeger, “Security in Computing”, Prentice Hall.
2. Bruce Schneier, “Applied Cryptography”, Wiley.
3. Matt Bishop, “Computer Security: Art and Science”, Addison-Wesley.
4. Eric Cole, “Network Security Bible”, Wiley Publishing.

Course Code:5IT226PE	**Course Title: Program Elective Course -I (ii)Data Science & Statistics Lab**	**L:0 T:0 P:2 C:1**
Minimum eight experiments must be completed based on the syllabus uniformly covering each of the units.		

Course Code:5IT226PE	**Course Title: Program Elective Course -I iii)Internet of Things Lab**	**L:0 T:0 P:2 C:1**
Below is the Tentative list of experiments. Minimum eight experiments must be completed based on the syllabus uniformly covering each of the units.		

LIST OF EXPERIMENTS:

1. To Interface PRI Motion Sensor with Raspberry Pi and write a program to control LED.
2. To Interface Optical Sensor with Raspberry Pi and write a program to control LED.
3. To Interface Rain Drop Sensor with Raspberry Pi and write a program to sound an alarm.
4. To Interface Moisture Sensor with Raspberry Pi and write a program to display value.
5. To Interface Touch Sensor with Raspberry Pi and write a program to detect and record physical touch.

6. To Interface Gas Sensor with Raspberry Pi and write a program to sounds an alarm.
7. To Interface Pressure Sensor with Raspberry Pi and write a program to display value.
8. To Interface Ultrasonic Sensor with Raspberry Pi and write a program to measure the distance between any two objects.

Course Code: 5IT227MD	**Course Title: MDM- III- Computer Networks**	**L-2, T-0, P-0, C-2**
Course Prerequisite:	Data Communication and Operating Systems	
Course Objectives:	1. To understand the fundamentals of computer networks and layered architectures. 2. To analyze data transmission, error control, and network protocols. 3. To learn network layer routing, and transport layer services.	
Course Outcomes (Expected Outcome):	After completion of the course, students will be able to: CO1: Understand basic concepts of computer networks, OSI/TCP-IP models, transmission media, and data link layer functions. CO2: Apply network layer concepts including IPv4 addressing, subnetting, IP, and routing protocols. CO3: Explain transport and application layer protocols along with basic network security concepts.	
Unit I:	**Unit Title: Introduction & Data Link Layer**	**Hours: 9**
Introduction to Computer Networks: Definition, Uses, Network Types (LAN, MAN, WAN), Network Topologies. Network Models: OSI Reference Model and TCP/IP Model – Functions and Comparison, Physical Layer Basics: Transmission Media (Guided and Unguided), Data Transmission Concepts. Data Link Layer: Error Detection and Correction: Parity Check, CRC, Hamming Code. Introduction to Ethernet and MAC Addressing.		
Unit II:	**Unit Title: Network Layer & Routing**	**Hours: 9**
Network Layer Services and Functions, Logical Addressing: IPv4 Addressing, Subnetting, CIDR, Internet Protocol (IP): Datagram Format, Fragmentation, Static and Dynamic Routing, Routing Protocols: RIP, OSPF, BGP (Overview), Basics of Network Devices: Routers, Switches, Gateways.		
Unit III:	**Unit Title: Transport & Application Layer**	**Hours: 9**
Transport Layer: Process-to-Process Communication, UDP and TCP: Services, Features, Comparison, TCP Flow Control and Congestion Control (Conceptual) Application Layer Protocols: DNS, HTTP/HTTPS, FTP, SMTP, Network Security Basics: Firewalls (Types), IDS/IPS (Introduction).		
Text Book: Behrouz A. Forouzan**,** "Data Communications and Networking"**,** 5th Edition, McGraw-Hill Education**.**		
Reference Books: 1. Andrew S. Tanenbaum & David J. Wetherall, *Computer Networks*, 5th Edition, Pearson Education. 2. William Stallings, *Data and Computer Communications*, 10th Edition, Pearson. 3. Larry L. Peterson & Bruce S. Davie, *Computer Networks: A Systems Approach*, Morgan Kaufmann. 4. James F. Kurose & Keith W. Ross, *Computer Networking: A Top-Down Approach*, Pearson. 5. Douglas E. Comer, *Internetworking with TCP/IP – Vol. 1*, Pearson.		

Course Code: 5IT228MD	**Course Title: MDM- IV Object Oriented Programming**	**L:2 T:0 P:0 C:2**
Course Prerequisite:	Principles of Programming Languages	
Course Objectives:	1. To prepare students for learning Object Oriented Programming Methodology. 2. To Study the basic concepts of Java such as operators, classes, objects, inheritance, packages and exception handling. 3. To Study inheritance, packages and exception handling.	

Course Outcomes (Expected Outcome):	1. Apply Object Oriented approach to design software. 2. Implement programs using classes and objects. 3. Specify the forms of inheritance and exception handling and use them in programs.	
Unit I:	Unit Title: **Introduction to Java Programming**	Hours:09
Introduction, Need of OOP, Principles of Object Oriented Languages, Procedural Language Vs OOP, Application of OOP, Java Virtual Machine, Java features, Program Structures. **Java Programming Constructs:** Variables, Primitive data types, Identifier, Literals, Operators, Expressions, Precedence Rules and Associativity, Primitive Type Conversion and Casting, Flow of Control.		
Unit II:	Unit Title**: Classes and Objects**	Hours: 09
Classes, Objects, Creating Objects, Methods, Constructors, Cleaning up Unused Objects, Class Variable and Methods, this keyword, Arrays, Command Line Arguments.		
Unit III:	Unit Title: **Inheritance and Exception Handling**	Hours:09
Inheritance vs. Aggregation, Polymorphism, Method Overloading Method Overriding, super keyword, final keyword, Abstract class. **Interfaces, Packages and Enumeration:** Interface, Packages, java.lang package, Enum type. **Exception Handling:** Introduction, Exception handling Techniques, User-defined exception.		
Text Book: Sachin Malhotra and Saurabh Choudhary: Programming in Java, Oxford University Press 2010.		
Reference Books: 1. Herbert Schildt: Java Complete References (McGraw Hill) 2. E. Balagurusamy: Programming with Java (McGraw Hill) 3. Khalid Mughal: A Programmer's Guide to Java Certification, 3rd Edition (Pearson) 4. Liang: A text Book of Java Programming, (PHI).		

Course Code: 5IT229ML	**Course Title: MDM-Lab-I :** **Object Oriented Programming - Lab**	**L:0 T:0 P:2 C:1**
Course Prerequisite:	Principles of Programming Languages	
Course Objectives:	1.To introduce students to Object Oriented Programming fundamentals. 2. To Enable students to apply object-oriented programming concepts for modeling. 3. To Train students to handle runtime errors and exceptional conditions in programs.	
Course Outcomes (Expected Outcome):	After successful completion of this practical course, students will be able to: 1. Design and execute Java programs using basic programming construct. 2. Implement programs using classes and objects, inheritance, polymorphism. 3. Implement object-oriented features such as abstract classes, interfaces, packages, enumerations and exception handling techniques.	
Practical based on the syllabus of Object Oriented Programming (5IT228MD). Following list is an indicative list and the subject teacher is free to design his/her own list of Eight experiments based on the syllabus of Object Oriented Programming (5IT228MD)		
List of Practicals:		
1. Write a Java program to display *Hello World* and demonstrate the structure of a Java program, variables, and identifiers. 2. Write a Java program to demonstrate primitive data types, operators, expressions, precedence and type conversion/casting. 3. Write Java programs using if–else, switch–case, for, while, and do–while statements. 4. Write a Java program to define a class, create objects, and implement methods. 5. Write a Java program to demonstrate constructors, this keyword, and static variables and methods. 6. Write a Java program using arrays and command line arguments for basic operations. 7. Write a Java program to demonstrate inheritance, method overriding, and use of the super keyword. 8. Write a Java program to demonstrate compile-time and run-time polymorphism using method overloading and overriding. 9. Write a Java program to demonstrate the use of an abstract class or interface, package creation, and enumeration (enum). 10. Write a Java program to demonstrate built-in and user-defined exceptions using try–catch, finally, throw, and throws.		

Course Code: 5IT230OE1	**Course Title: Open Elective-III** **(i) Fundamentals of Cyber Security**	**L:2 T:0 P:0 C:2**

<table>
<tr><td>Course Prerequisite:</td><td colspan="2">Students should have:
Basic Computer Knowledge: Familiarity with computer hardware, operating systems, and general software usage.
Networking Fundamentals: Understanding of computer networks, Internet protocols (TCP/IP), and basic network operations.
Programming Fundamentals: Basic knowledge of at least one programming language (e.g., Python, C, or Java) for understanding scripts, algorithms, and security tools.
Mathematical Concepts: Basic understanding of discrete mathematics, logic, and probability for cryptography and security analysis.</td></tr>
<tr><td>Course Objectives:</td><td colspan="2">Introduce the fundamental concepts of cyber security, including threats, vulnerabilities, and security principles used to protect information systems.
Explain cryptographic techniques and secure communication mechanisms used for data protection and authentication.
Develop understanding of network, operating system, and application security mechanisms to defend against cyber attacks.
Familiarize learners with cyber crimes, digital forensics processes, and incident response methodologies.
Create awareness of cyber laws, ethical issues, and professional responsibilities in cyberspace.
Expose students to emerging cyber security trends, security management practices, and real-world attack scenarios.</td></tr>
<tr><td>Course Outcomes (Expected Outcome):</td><td colspan="2">Explain basic cyber security concepts, security goals, threats, vulnerabilities, and attack models in cyber space.
Apply fundamental cryptographic techniques for secure data transmission, authentication, and integrity verification.
Analyze network and system-level security issues and recommend appropriate security mechanisms.
Identify common malware, web application attacks, and operating system security weaknesses.
Understand cybercrimes, digital forensics procedures, and incident response processes in cyber investigations.
Evaluate emerging cyber security challenges and apply best practices, standards, and policies to real-world scenarios.</td></tr>
<tr><td>Unit I:</td><td>Unit Title: Introduction to Cyber Security</td><td>Hours: 9</td></tr>
<tr><td colspan="3">Introduction to Cyber Space and Information Security, CIA Triad: Confidentiality, Integrity, Availability,
Cyber Security vs Information Security, Security Goals, Threats, Vulnerabilities, and Attacks, Types of Cyber Attacks: Passive and Active, Cyber Security Challenges and Trends, Legal, Ethical, and Professional Issues in Cyber Security</td></tr>
<tr><td>Unit II:</td><td>Unit Title: Operating System and Application Security</td><td>Hours: 9</td></tr>
<tr><td colspan="3">Operating System Security Principles, Authentication and Authorization Mechanisms, Access Control Models: DAC, MAC, RBAC, Malware Types: Viruses, Worms, Trojans, Ransomware, Secure Software Development Life Cycle (SSDLC), Web Application Security Basics, Common Web Attacks: SQL Injection, XSS, CSRF</td></tr>
<tr><td>Unit III:</td><td>Unit Title: Cyber Crimes, Digital Forensics and Incident Response</td><td>Hours: 9</td></tr>
<tr><td colspan="3">Introduction to Cyber Crimes and Cyber Laws, Types of Cyber Crimes: Identity Theft, Phishing, Cyber Stalking, Cyber Crime Investigation Process, Digital Evidence and Forensic Principles, Phases of Digital Forensics, Incident Response Life Cycle, Basics of Risk Management and Business Continuity</td></tr>
<tr><td colspan="3">Text Book: “Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives” Nina Godbole & Sunit Belapure, Wiley India</td></tr>
<tr><td colspan="3">Reference Books:
1. “Principles of Information Security” – Michael E. Whitman & Herbert J. Mattord, Cengage
2. “Computer Security: Principles and Practice” – William Stallings & Lawrie Brown, Pearson
3. “Network Security Essentials: Applications and Standards” – William Stallings, Pearson.
4. “Cybersecurity and Cyberwar: What Everyone Needs to Know” – P.W. Singer & Allan Friedman, Oxford University Press</td></tr>
</table>

Course Code: 5IT230OE2	**Course Title: Open Elective-III** **(ii) Design Thinking and Innovation**	**L:2 T:0 P:0 C:2**
Course Pre-requisite:	The primary prerequisite for Design Thinking & Innovation is an open, creative, and problem-solving mindset rather than technical knowledge.	
Course Objectives:	1. To introduce students to the fundamental principles and stages of design thinking, including empathy, ideation, and prototyping. 2. To enhance students' ability to approach complex problems creatively and collaboratively, using design thinking methodologies. 3. Show students how to use design thinking techniques in everyday situations to come up with innovative solutions.	
Course Outcomes (Expected Outcome):	After completion of the course, the students will be able to: 1. Know the steps of design thinking and how to use them to create new solutions for problems. 2. Better at understanding what people need, which helps in making designs that work well for users. 3. Make simple models and improve them by testing and getting feedback to find the best solutions.	
Unit I:	Unit Title: An Overview of Design Thinking	Hours:09
Introduction to Design Thinking, Creativity and Innovation- The Drivers of Design Thinking, Attributes of Design Thinking, Critical Success Factors, Principles of Design Thinking, Evolution of Design Thinking, 3Ps of Design Thinking, Design Thinking as a Business Strategy.		
Unit II:	Unit Title: Features and applications of Design Thinking	Hours:09
Features of Design Thinking, Solving wicked problems, Design thinking aids project management; Applications of Design Thinking, Design Thinking Vs. Traditional Thinking. Design Thinking Tools and Techniques that are popularly used in different phases of the design thinking process.		
Unit III:	Unit Title: Double Diamond Design Thinking	Hours:09
Double Diamond Model, The 4 D's of Double Diamond; Designing for Business Growth: Growth Design Vs. Product Design, Designing for growth process; Ethics in Design Thinking: Five ways of Ethical Thinking. Design Thinking for Sustainability: Introduction, Pillars of Sustainability.		
Text Book: 1. E. Balagurusamy, Bindu Vijaykumar, "Design Thinking: A Beginner's Perspective", McGrawHill Publications. 2. E. Balagurusamy, Developing Thinking Skills.		
Reference Books: 1. Anuja Agarwal, "Design Thinking: A Framework for Applying Design Thinking in Problem Solving", Cengage Publications. 2. Pavan Soni, "Design Your Thinking.		

Course Code: 5IT230OE3	**Course Title: Open Elective-III** **Principals of Management**	**L:2 T:0 P:0 C:2**
Course Prerequisite:	1. Basic understanding of organizational structure and functions 2. Fundamental knowledge of communication skills 3. Exposure to engineering project work and teamwork	
Course Objectives:	1. Understand the basic concepts, principles, and evolution of management. 2. Learn planning and organizing techniques used in engineering organizations. 3. Develop leadership and communication skills for effective management.	
Course Outcomes (Expected Outcome):	1. Explain the basic concepts and functions of management. 2. Apply planning and organizing techniques in technical projects. 3. Demonstrate leadership and effective communication in teams	
Unit I:	Unit Title: Introduction to Management	Hours:09
Meaning, Nature and Scope of Management, Functions of Management: Planning, Organizing, Directing, Coordinating, Controlling. Levels of Management, Evolution of Management Thought: Scientific Management (Taylor), Administrative Theory (Fayol), Behavioral Approach, Role of Management in Engineering and IT Industries		
Unit II:	Unit Title: Planning and Decision Making	Hours:09
Meaning and Importance of Planning, Steps in Planning Process, Types of Plans: Strategic, Tactical, Operational, Forecasting Techniques, Decision Making: Types of Decisions, Decision Making Process, Application of Planning in Software and IT Projects.		

Unit III:	Unit Title: Organizing and Staffing	Hours:09
Meaning and Principles of Organizing, Organization Structure: Line, Functional, Matrix , Authority, Responsibility and Accountability, Delegation of Authority, Staffing: Recruitment, Selection and Training, Role of HR in IT and Engineering Organizations.		
Text Book: P. C. Tripathi & P. N. Reddy – Principles of Management, Tata McGraw Hill.		
Reference Books: 1. V. S. P. Rao & V. Hari Krishna – Management: Text and Cases, Excel Books. 2. Harold Koontz & Heinz Weihrich – Essentials of Management, McGraw Hill. 3. Robbins & Coulter – Management, Pearson Education. 4. Ramaswamy – Principles of Management, Himalaya Publishing House.		

Course Code: 5IT247DH1	**Course Title: Big Data and Data Lakes**	**L:4 T:0 P:0 C:4**
Course Prerequisite:	• Basic Database Management Systems • Programming fundamentals (Python/Java) • Operating Systems concepts • Computer Networks basics	
Course Objectives:	1. To understand Big Data characteristics and distributed processing models. 2. To learn Hadoop ecosystem and parallel processing techniques. 3. To design and implement Data Lake architectures. 4. To perform ingestion, storage, and governance of large-scale data. 5. To apply analytics and machine learning on Big Data platforms..	
Course Outcomes(Expected Outcome):	After completion of the course, students will be able to: 1. Explain Big Data processing frameworks and architectures 2. Work with Hadoop and Spark ecosystem 3. Design and manage Data Lakes 4. Implement ingestion pipelines and governance policies 5. Perform analytics and build scalable data applications.	
Unit I:	Unit Title: Fundamentals of Big Data	Hours: 8
Introduction to Data Explosion and Digital Data Growth, Definition and Characteristics of Big Data (5V's: Volume, Velocity, Variety, Veracity, Value), Traditional Data Processing vs Big Data Processing, Structured, Semi-structured and Unstructured Data, Big Data Lifecycle, Big Data Architecture Overview, Distributed Computing Basics, CAP Theorem and BASE Properties, Batch Processing vs Stream Processing, Applications of Big Data (Healthcare, Finance, IoT, Social Media, Smart Cities)		
Unit II:	Unit Title: Hadoop Ecosystem	Hours: 8
Introduction to Apache Hadoop, Hadoop Architecture and Components, Hadoop Distributed File System (HDFS) Architecture, Name Node, Data Node, Secondary Name Node, Blocks, Replication, Rack Awareness, MapReduce Programming Model, Mapper, Reducer, Combiner, Partitioner, Job Execution Workflow, YARN Resource Management, Hadoop Cluster Setup Concepts (Pseudo vs Fully Distributed), Hadoop Commands and File Operations, Limitations of MapReduce		
Unit III:	Unit Title: Data Processing Frameworks	Hours: 8
Need for Advanced Processing Engines, Apache Spark Architecture, RDD, Data Frame, Dataset, Lazy Evaluation, DAG Execution Engine, Spark Core Operations (Transformations & Actions), Spark SQL and Structured Processing, Spark Streaming Basics, Comparison: MapReduce vs Spark, Introduction to NoSQL Databases, HBase Architecture, Cassandra Overview, Query Engines: Hive and Pig, HiveQL Basics, Pig Latin Basics		
Unit IV:	Unit Title: Data Lakes Concepts and Architecture	Hours: 8
Evolution: Database → Data Warehouse → Data Lake, Data Lake Definition and Characteristics, Data Lake vs Data Warehouse vs Data Mart, Data Lake Architecture Layers, Ingestion Layer, Storage Layer, Processing Layer, Access Layer, Schema-on-Read vs Schema-on-Write, Metadata Management and Catalogs, Data Lake Zones (Raw, Cleaned, Curated, Sandbox), Data Lake Governance Principles, Modern Lakehouse Architecture Concept		
Unit V:	Unit Title: Data Ingestion, Storage and Governance	Hours: 8
Data Ingestion Techniques, Batch Ingestion, Real-time Streaming Ingestion, Tools Overview, Sqoop (RDBMS to HDFS), Flume (Log ingestion), Kafka Basics, Data Formats, CSV, JSON, XML, Avro, Parquet, ORC, Data Partitioning and Optimization, Data Quality Management, Data Cataloging and Metadata Management, Security in Big Data, Authentication and Authorization, Kerberos Basics, Encryption (At Rest & In Transit), Data Privacy and Compliance Concepts.		
Unit VI:	Unit Title: Analytics and Applications on Data Lakes	Hours: 8
Data Exploration and Querying, Analytical Processing, OLAP vs OLTP in Big Data, Machine Learning on Big Data, ML Pipelines Concept, Spark MLlib Overview, Visualization and BI Tools Integration, Real-time Analytics		

Use Cases, Industry Applications, Fraud Detection, Recommendation Systems, Predictive Maintenance, Performance Optimization Techniques, Future Trends, Lakehouse, Data Fabric, AI-Driven Data Platforms
Text Book: Tom White — *Hadoop: The Definitive Guide* O'Reilly Media.
Reference Books: 1. Big Data Fundamentals — Thomas Erl, Wajid Khattak, Paul Buhler 2. Spark: The Definitive Guide — Bill Chambers & Matei Zaharia 3. The Data Warehouse Toolkit — Ralph Kimball & Margy Ross 4. Fundamentals of Data Engineering — Joe Reis & Matt Housley

Course Code: 5IT247DH2	**Course Title: Big Data Analytics**	**L:4 T:0 P:0 C:4**
Course Prerequisite:	• Database Management Systems • Statistics fundamentals • Basic Programming (Python/Java)	
Course Objectives:	1. To understand Big Data ecosystem and distributed storage models. 2. To learn Hadoop and Spark for data processing. 3. To apply data mining and analytics algorithms on large datasets. 4. To work with NoSQL databases and big data storage models. 5. To analyze and visualize data for decision making.	
Course Outcomes(Expected Outcome):	After completion of the course, students will be able to: 1. Explain Big Data architecture and analytics workflow 2. Process large datasets using Hadoop/Spark 3. Apply classification and clustering techniques 4. Use NoSQL databases for big data storage 5. Perform analytics and visualization for real-world problems	
Unit I:	Unit Title: Introduction to Big Data and Analytics	Hours: 8
Evolution of Data and Need for Big Data Analytics, Characteristics of Big Data (5V Model), Types of Data (Structured, Semi-Structured, Unstructured), Big Data Analytics Lifecycle, Traditional Analytics vs Big Data Analytics, Big Data Architecture Overview, Distributed Computing Concepts, Applications of Big Data Analytics (Healthcare, Retail, Finance, IoT, Social Media)		
Unit II:	Unit Title: Hadoop Framework and Distributed Storage	Hours: 8
Introduction to Hadoop Ecosystem, Hadoop Architecture, Hadoop Distributed File System (HDFS), Blocks and Replication, Name Node & Data Node, Rack Awareness, MapReduce Programming Model, Mapper, Reducer, Combiner, Execution Flow, YARN Resource Management, Limitations of MapReduce		
Unit III:	Unit Title: Big Data Processing with Spark	Hours: 8
Need for In-Memory Processing, Apache Spark Architecture, Resilient Distributed Dataset (RDD), Transformations and Actions, Spark SQL and Data Frames, Spark Streaming Concepts, Comparison: Hadoop MapReduce vs Spark, Performance Advantages of Spark		
Unit IV:	Unit Title: Data Mining and Analytical Methods	Hours: 8
Introduction to Data Mining, Classification Techniques, Decision Trees, Naïve Bayes, Clustering Techniques, K-Means, Hierarchical Clustering, Association Rule Mining (Apriori Algorithm), Regression Analysis, Recommendation Systems Basics, Text Analytics Fundamentals		
Unit V:	Unit Title: NoSQL and Data Management	Hours: 8
Limitations of Relational Databases in Big Data, Introduction to NoSQL Databases, Types of NoSQL Databases, Key-Value Stores, Column-Oriented Databases, Document Databases, Graph Databases, CAP Theorem and BASE Properties, HBase Architecture Overview, Data Warehousing vs Big Data Storage, Data Quality and Governance Concepts		
Unit VI:	Unit Title: Visualization, Applications and Case Studies	Hours: 8
Data Visualization Principles, Visual Analytics Tools Concepts, Real-time Analytics, Predictive Analytics, Sentiment Analysis, Fraud Detection, Recommendation Engines, Big Data in AI and Machine Learning, Ethical Issues and Privacy, Future Trends in Big Data Analytics		
Text Book: Big Data Analytics, Dr. Radha Shankarmani and Dr. M. Vijayalakshmi Publisher: Wiley India.		
Reference Books: 1. Big Data and Analytics — Seema Acharya & Subhashini Chellappan (Wiley India) 2. Big Data Analytics — Raj Kamal & Preeti Saxena (McGraw-Hill Education India)		

Course Code: 6IT231PC	Course Title: Compiler Design	L-3 T-0 P-0 C-3
Course Prerequisite:	Basic knowledge of Programming in C, Discrete Mathematics, Operating Systems and Theory of Computation.	
Course Objectives:	1. To explain various phases of compiler and their working. 2. To describe the various top-down and Bottom-Up parsers. 3. To illustrate the various optimization techniques for designing various optimizing compilers. 4. Explore the use of storage organization, data structures and optimization techniques in the design and construction of efficient compilers. 5. To recognize the challenges involved in code generation and the basics of code generation algorithms.	
Course Outcomes (Expected Outcome):	1. Describe the fundamentals of compiler and various phases of compilers. 2. Design and implement to test LL parsers. 3. Solve the various parsing techniques of LR Parser like SLR, CLR & LALR. 4. Examine the concept of Syntax-Directed Definition and translation. 5. Assess the concept of Intermediate-Code Generation and run-time environment. 6. Explain the concept of code generation and code optimization.	
Unit I:	**Introduction to Compilers**	**Hours:** 07
Introduction to Compilers, Phases of compiler design, Relating Compilation Phases with Formal Systems. Lexical Analysis, tokens, pattern and lexemes, Design of Lexical analyzer, Regular Expression, recognition of tokens.		
Unit II:	**Syntax Analysis**	**Hours:** 07
The role of the parser, Review of context free grammar for syntax analysis, Ambiguity in Grammar, Elimination of left recursion and left factoring. Top-down parsing: recursive descent parsing, predictive parsers, FIRST and FOLLOW, LL (1) Grammars, Construction of predictive parsing tables, non-recursive predictive parsing.		
Unit III:	**Bottom-Up Parsing**	**Hours:** 07
Handle pruning, Stack implementation of Shift Reduce Parsing, conflicts during shift reduce parsing Introduction to LR parsing: Simple LR, Items and the LR(0) Automation, The LR-Parsing algorithm, Construction of SLR parsing table, More powerful LR Parsers: canonical LR(1) Items, Constructing LR(1) sets of items and canonical LR(1) parsing tables, Constructing LALR parsing tables.		
Unit IV:	**Syntax Directed Translation**	**Hours:** 07
Syntax directed definitions, Inherited and synthesized attributes, Evaluation orders of SDD's: Dependency Graphs, S-attributed definitions, L-attributed definition. Construction of syntax trees. Syntax-directed Translation Schemes.		
Unit V:	**Intermediate-Code Generation**	**Hours:** 07
Variants of Syntax Trees: Directed Acyclic Graphs (DAG), Three Address Code. Run Time Environments: Storage Organization, Static versus Dynamic Storage Organization, Stack Allocation of Space: Activation trees, Activation Records, Calling Sequences. Introduction to Garbage Collection: Design Goals for Garbage Collectors.		
Unit VI:	**Code Generation and Optimization**	**Hours:** 07
Issues in Design of a Code generator, The Target Language, Address in the target code, Basic blocks and flow graphs. Optimization of Basic Blocks, Peephole Optimization and The Principal sources of optimization: global common subexpression, copy propagation, Dead-code elimination.		
Text Book: 1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman Compilers: "Principles, Techniques and Tools", Pearson Education Second Edition.		
Reference Books: 1. D. M. Dhamdhere, Compiler Construction—Principles and Practice, (2/e), Macmillan India. 2. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman Compilers: "Principles, Techniques and Tools", Pearson Education (Low Price Edition). 3. V. Raghavan; Principles of Compiler Design, McGraw Hill Education (India), 2010. 4. Andrew Appel, Modern Compiler Implementation in C, Cambridge University press. 5. K C. Louden "Compiler Construction—Principles and Practice" India Edition, CENGAGE. 6. J. P. Bennett, "Introduction to Compiling Techniques", 2/e (TMH).		

Course Code: 6IT232PC	**Course Title: Design and Analysis of Algorithm**	**L-3 T-0 P-0 C-3**
Course Prerequisite:	Discrete Structure & Graph Theory, Data Structure	
Course Objectives:	1. To understand the fundamental concepts of algorithm design and analysis. 2. To learn various algorithm design techniques such as divide and conquer, greedy, dynamic programming, and backtracking. 3. Analyze the time and space complexity of algorithms and evaluate the trade-offs among different algorithm design techniques. 4. Apply algorithm design techniques to solve real-world problems and evaluate the efficiency and correctness of the resulting algorithms.	
Course Outcomes (Expected Outcome):	1. Analyse worst-case running times of algorithms using asymptotic analysis. 2. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. 3. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. 4. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. 5. Understand the concept of Backtracking and Combinational Search. 6. Identify worst and average case behavior of Polynomial Time; Non Polynomial Time Algorithms.	
Unit I:	**Top-Down Design:**	**Hours:7**
Structured Programming, Control Constructs, Procedures & Functions, Recursion. Iterative Algorithm Design Issue: Introduction, Use of Loops, Efficiency of Algorithms, Estimating & Specifying Execution Times, Order Notations, Algorithm Strategies, Design using Recursion.		
Unit II:	**Divide and Conquer:**	**Hours:7**
Multiplication Algorithm and its analysis, Application to Graphics Algorithms: Introduction to Triangulation, Covex Hulls.		
Unit III:	**Greedy Methods:**	**Hours:7**
Introduction, Knapsack Problem, Job sequencing with deadlines, Minimum Spanning Trees, Prim's Algorithms, Kruskal's Algorithm, Dijkstras Shortest Path Algorithm.		
Unit IV:	**Dynamic Programming:**	**Hours:7**
Introduction, Multistage Graphs, Traveling Salesman, Matrix multiplication, Longest Common Sub-Sequences, Optimal Polygon Triangulation. Single Source Shortest Paths		
Unit V:	**Backtracking:**	**Hours:7**
Combinational Search, Search & Traversal, Backtracking Strategy, Backtracking Framework-8-Queen's problem, graph coloring, Some Typical State Spaces, Branch-and-Bound Algorithms.		
Unit VI:	**Polynomial Time & Non Polynomial Time Algorithms:**	**Hours:7**
Worst and Average case Behavior, Time Analysis of Algorithm, Efficiency of Recursion, Complexity, Examples of Complexity Calculation for Various Sorting algorithms.		
Text Book: Dave and Dave: "Design and Analysis of Algorithms" Pearson Education. **Reference Books:** 1. Aho,Hopcroft & Ullman "The Design & Analysis of Computer Algorithms", Addison Wesley 2. G. Brassard, P.Bratley: "Fundamentals of Algorithmics" , PHI 3. Horowitz & Sahani: "Fundamental Algorithms" , Galgotia. 4. Cormen, T.H, Lierson & Rivest: "Introduction to Algorithms", Mc Graw-Hill		

Course Code: 6IT233PC	Course Title: Artificial Intelligence	L-3 T-0 P-0 C-3
Course Prerequisite:	Basic Knowledge of Programming, Introduction to Python, Discrete Mathematics and Data Structure	
Course Objectives:	1. To present an overview of Artificial Intelligence (AI) principles and approaches. 2. To understand the historical evolution of Artificial Intelligence. 3. To learn various searching techniques and identify to address a particular problem. 4. Learn knowledge representation technique, reasoning technique. 5. To understand Predicate Logic and Adversarial Search & Games.	
Course Outcomes (Expected Outcome):	1. Describe concepts of Artificial Intelligence and different types of intelligent agents and their architecture. 2. Explain different types of uninformed search strategies. 3. Illustrate different types of heuristic search strategies. 4. Investigate various knowledge representation issues. 5. Use predicate logic for knowledge representation. 6. Inspect adversarial search and game theory.	
Unit I:	**Introduction to AI**	**Hours:** 07
Introduction: What Is AI? The Foundations of Artificial Intelligence, The History of Artificial Intelligence, The State of the Art, Risks and Benefits of AI, Intelligent Agents: Agents and Environments, Good Behavior: The Concept of Rationality, The Nature of Environments, The Structure of Agents		
Unit II:	**Uninformed Search Strategies**	**Hours:** 07
Uninformed Search Strategies: Problem-Solving Agents, Example Problems, Search Algorithms, Breadth-First Search, Uniform Cost Search, Depth First Search, Bidirectional Search, Depth Limited Search, Iterative Deepening Depth-First Search.		
Unit III:	**Informed Search Strategies**	**Hours:** 07
Informed Search Strategies: Basic Concept of Heuristic Search and Knowledge, Designing of Heuristic Function, Heuristic Search Strategies: Generate-And-Test, Best-First Search, Problem Reduction, Hill Climbing, Constraint Satisfaction, Means-Ends-Analysis.		
Unit IV:	**Knowledge Representation and Issues**	**Hours:** 07
Knowledge Representation and Issues: Representation and Mapping. Approaches to Representation, Issues in Representation: Important attribute, relationship among attributes, choosing the granularity of representation, representing sets of Objects, finding the right structures as needed.		
Unit V:	**Knowledge Representation using Predicate Logic**	**Hours:** 07
Knowledge Representation using Predicate Logic: Representing Simple Facts in Logic, Representing Instance and ISA relationship, Computable Functions and Predicates, Resolution.		
Unit VI:	**Adversarial Search & Games**	**Hours:** 07
Adversarial Search & Games: Game Theory, Optimal Decisions in Games, MiniMax Search, Alpha Beta Pruning, Additional Refinements, Monte Carlo Tree Search, Stochastic Games, Partially Observable Games, Limitations of Game Search Algorithms		
Text Books: 1. Artificial Intelligence: A Modern Approach by Stuart Russell & Peter Norvig (Pearson - 4th Ed.) 2. Artificial Intelligence by Elaine Rich and Kevin Knight (Tata McGraw Hill - 3rd Ed.)		
Reference Books: 1. Artificial Intelligence by Ela Kumar (IK International Publishing House Pvt. Ltd.) 2. A First Course in Artificial Intelligence by Deepak Khemani (Tata McGraw Hill - 1st Ed.) 3. Artificial Intelligence and Expert Systems by Patterson (PHI) 4. Introduction to Artificial Intelligence by Rajendra Akerkar (PHI Learning Pvt. Ltd.)		

Course Code: 6IT234PE	Course Title: Program Elective Course-II (i)Cryptography	L:3 T:0 P:0 C:3
Course Prerequisite:	Computer Networks	
Course Objectives:	1. Understand fundamental concepts like confidentiality, integrity, authentication, and non-repudiation. 2. Understand modular arithmetic, number theory, and algebraic structures used in cryptography 3. Learn cryptographic techniques 4. Study modern algorithms 5. Learn about secure protocols and authentication mechanisms 6. Identify vulnerabilities, threats, and attacks in communication systems.	
Course Outcomes (Expected Outcome):	1. Understand the principles and practices of cryptographic techniques. 2. Understand information security goals for designing secure systems. 3. Implement symmetric, asymmetric and hashing algorithms to secure data. 4. Apply security algorithms in solving real-life security problems in communicating systems. 5. Apply security to information over the network and World Wide Web. 6. Develop secure communication programs using cryptographic utilities.	
Unit I:	**Unit Title: Basics of Cryptography**	**Hours: 7**
Information Security understanding, Security goals, Security attacks, Security services, security mechanisms.		
Unit II:	**Unit Title: Cryptographic Mathematics and Classical Ciphers**	**Hours: 7**
Modular arithmetic, linear congruence, Algebraic structure quadratic congruence, Symmetric cipher model, substitution ciphers, transposition ciphers, steganography.		
Unit III:	**Unit Title: Modern symmetric key ciphers**	**Hours: 7**
Modern block ciphers, modern stream ciphers, Data Encryption standard, advanced encryption standard, Electronic code book mode, CBC, cipher feedback mode, output feedback mode.		
Unit IV:	**Unit Title: Public key cryptography**	**Hours: 7**
RSA, RSA proof, RSA attacks, Rabin cryptosystem, Key management: Diffie Hellman Key Exchange Algorithm.		
Unit V:	**Unit Title: Message Authentication and Hash functions**	**Hours: 7**
Authentication requirements, functions, Message authentication codes (MAC), Hash functions, security of Hash functions, SHA- 512, Basics, digital signature standards.		
Unit VI:	**Unit Title: Network and System Security**	**Hours: 7**
Understanding of Worms, Virus, Trojan Horse, Malwares, IP and Network Security ,Web security Email Security, System Security, tools		
Text Book: William Stallings, "Cryptography and Network Security: Principles and Practice".		
Reference Books: 1. Bruce Scheneir: “Applied Cryptography”, John Wiley. 2. Behrouz Forouzan: “Cryptography & Network Security”, TMH.		

Course Code: 6IT234PE	Course Title: Program Elective Course-II (ii)Network Security	L:3 T:0 P:0 C:3
Course Prerequisite:	• Basic Computer Knowledge: Understanding of computer hardware, operating systems, and general software usage. • Networking Fundamentals: Knowledge of computer networks, TCP/IP protocols, routing, switching, and Internet basics. • Programming Fundamentals: Familiarity with at least one programming language (e.g., Python, C, or Java) for scripting and network tools. • Mathematical Background: Basic understanding of discrete mathematics, number theory, and logic (for cryptography and security analysis).	
Course Objectives:	1. To introduce core concepts of network security, including threats, vulnerabilities, and security mechanisms. 2. To explain cryptographic principles and techniques used to secure network communication. 3. To develop understanding of network protocols and secure communication standards. 4. To familiarize learners with network attack types, malware, intrusion detection, and prevention systems. 5. To expose students to authentication, access control, and security management practices.	

<table>
<tr><td></td><td colspan="2">6. To create awareness of emerging network security trends, cloud/IoT security, and real-world case studies.</td></tr>
<tr><td>Course Outcomes:</td><td colspan="2">On completion of the course students are expected to -
1. Explain fundamental concepts, goals, and threats in network security.
2. Apply cryptographic methods for secure communication and data protection.
3. Analyze network protocols and secure their implementation against potential attacks.
4. Identify network attacks, malware, and vulnerabilities, and implement preventive measures.
5. Implement authentication and access control mechanisms and understand security management practices.
6. Evaluate emerging network security challenges and apply best practices to real-world network scenarios.</td></tr>
<tr><td>Unit I:</td><td>Unit Title: Introduction to Network Security</td><td>Hours: 7</td></tr>
<tr><td colspan="3">Overview of Network Security Concepts and Goals, Security Attacks: Passive and Active, Security Services: Confidentiality, Integrity, Authentication, Non-repudiation, Security Mechanisms and Model: ISO Security Architecture, Network Security Design Principles, Security Policies and Practices, Case Study: Real-world network security incidents</td></tr>
<tr><td>Unit II:</td><td>Unit Title: Cryptography and Encryption Techniques</td><td>Hours: 7</td></tr>
<tr><td colspan="3">Introduction to Cryptography and Key Concepts, Symmetric Key Cryptography: DES, AES, Block and Stream Ciphers, Asymmetric Key Cryptography: RSA, Diffie–Hellman, Hash Functions: SHA, MD5, HMAC, Digital Signatures and Certificates, Public Key Infrastructure (PKI), Applications of Cryptography in Network Security</td></tr>
<tr><td>Unit III:</td><td>Unit Title: Network Security Protocols</td><td>Hours: 7</td></tr>
<tr><td colspan="3">Secure Socket Layer (SSL) / Transport Layer Security (TLS), Secure Email Protocols: PGP, S/MIME, IP Security (IPSec): Authentication Header (AH), Encapsulating Security Payload (ESP), Virtual Private Networks (VPNs), Wireless Network Security: WPA, WPA2, WPA3, Secure Routing Protocols, Practical Example: HTTPS and VPN configurations</td></tr>
<tr><td>Unit IV:</td><td>Unit Title: Network Threats and Attack Prevention</td><td>Hours: 7</td></tr>
<tr><td colspan="3">Common Network Attacks: Spoofing, Sniffing, DoS, DDoS, Man-in-the-Middle, Malware and Worms affecting networks, Firewalls: Packet Filtering, Stateful Inspection, Proxy Firewalls, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), Network Monitoring Tools and Techniques, Honeypots and Deception Techniques, Case Studies of Network Breaches</td></tr>
<tr><td>Unit V:</td><td>Title: Authentication, Access Control, and Security Management</td><td>Hours: 7</td></tr>
<tr><td colspan="3">Authentication Methods: Passwords, Biometrics, Tokens, Access Control Models: DAC, MAC, RBAC, AAA Framework: Authentication, Authorization, Accounting, Security Management Practices and Policies, Security Auditing and Compliance Standards, Role of Security Operations Centers (SOC), Incident Response and Risk Management</td></tr>
<tr><td>Unit VI:</td><td>Unit Title: Emerging Trends and Network Security Case Studies</td><td>Hours: 7</td></tr>
<tr><td colspan="3">Cloud Network Security Challenges, IoT and Mobile Network Security, Ethical Hacking and Penetration Testing in Networks, Security in Software Defined Networks (SDN), Network Forensics and Logging, Real-world Network Security Case Studies, Best Practices and Future Trends in Network Security</td></tr>
<tr><td colspan="3">Text Book: “Network Security Essentials: Applications and Standards” – William Stallings, 6th Edition, Pearson</td></tr>
<tr><td colspan="3">Reference Books:
1. “Principles of Network and System Administration” – Mark Burgess, Wiley
2. “Computer Networking: A Top-Down Approach” – Kurose & Ross, Pearson
3. “Cryptography and Network Security” – William Stallings, 8th Edition, Pearson
4. “Network Security: Private Communication in a Public World” – Charlie Kaufman, Radia Perlman, Mike Speciner, 2nd Edition, Pearson</td></tr>
</table>

<table>
<tr><td>Course Code: 6IT234PE</td><td>Course Title: Program Elective Course- II (iii)Big Data Analytics</td><td>L:3 T:0 P:0 C:3</td></tr>
<tr><td>Course Prerequisite:</td><td colspan="2">Students should have:
Basic knowledge of Programming Concepts (preferably Java / Python)
Understanding of Data Structures such as arrays, lists, trees, and graphs
Fundamental knowledge of Database Management Systems (DBMS)
Basic understanding of Operating Systems and Computer Networks</td></tr>
<tr><td>Course Objectives:</td><td colspan="2">Introduce students to the fundamental concepts and characteristics of Big Data and its role in modern data-driven applications.
Provide a clear understanding of Big Data architectures and frameworks such as Hadoop, Spark, and NoSQL databases.
Explain the MapReduce programming paradigm and its application in processing large-scale datasets.
Familiarize students with stream data processing techniques and algorithms used for handling continuous data.
Introduce Big Data mining algorithms for pattern discovery, clustering, and classification.
Enable students to understand real-world Big Data analytics applications such as web analytics, social network analysis, and recommendation systems.</td></tr>
<tr><td>Course Outcomes
(Expected Outcome):</td><td colspan="2">Explain the core concepts, characteristics, and challenges of Big Data, and differentiate between traditional data processing systems and Big Data approaches.
Describe the architecture and working of Big Data frameworks such as Hadoop, Spark, and NoSQL databases.
Apply the MapReduce paradigm to solve basic large-scale data processing problems such as aggregation, joins, and matrix operations.
Analyze streaming data models and apply sampling, filtering, and approximate counting techniques for real-time data streams.
Identify and explain Big Data mining algorithms for frequent pattern mining, clustering, and classification in large datasets.
Demonstrate understanding of Big Data analytics applications, including link analysis, social network mining, and recommendation systems.</td></tr>
<tr><td>Unit I:</td><td>Unit Title: Introduction to Big Data</td><td>Hours: 7</td></tr>
<tr><td colspan="3">Introduction to Big Data, Characteristics and Types of Big Data, Traditional Data Processing vs Big Data Approach, Big Data Use Cases and One Case Study of Big Data Solutions</td></tr>
<tr><td>Unit II:</td><td>Unit Title: Big Data Frameworks: Hadoop, Spark, and NoSQL</td><td>Hours: 7</td></tr>
<tr><td colspan="3">Introduction to Hadoop, Hadoop Architecture and Core Components (HDFS, YARN, MapReduce – overview), Hadoop Ecosystem: Hive, Pig, HBase, Sqoop (conceptual overview), Introduction to Apache Spark (comparison with Hadoop MapReduce), Introduction to NoSQL Databases, NoSQL Data Models: Key-Value, Column-Family, Document Stores, Overview of MongoDB</td></tr>
<tr><td>Unit III:</td><td>Unit Title: MapReduce Paradigm</td><td>Hours: 7</td></tr>
<tr><td colspan="3">MapReduce Programming Model, Map Phase, Reduce Phase, and Data Flow, Role of Combiners and Fault Tolerance, MapReduce Execution Workflow, Basic Algorithms Using MapReduce: Selection and Projection, Aggregation and Grouping, Natural Join (conceptual), Matrix Multiplication (single MapReduce overview)</td></tr>
<tr><td>Unit IV:</td><td>Unit Title: Big Data Stream Processing</td><td>Hours: 7</td></tr>
<tr><td colspan="3">Stream Data Model and Stream Sources, Stream Processing Challenges, Sampling Techniques for Data Streams, Bloom Filters for Stream Filtering, Counting Distinct Elements: Count-Distinct Problem, Flajolet-Martin Algorithm (overview), Sliding Window Counting: DGIM Algorithm (conceptual understanding)</td></tr>
<tr><td>Unit V:</td><td>Unit Title: Big Data Mining Algorithms</td><td>Hours: 7</td></tr>
<tr><td colspan="3">Frequent Pattern Mining: Challenges with Large Datasets, SON Algorithm (conceptual with MapReduce), Clustering Algorithms: Canopy Clustering, CURE Algorithm (overview) Classification Algorithms: Decision Trees for Big Data (parallel concept), Overview of SVM and KNN for Big Data</td></tr>
<tr><td>Unit VI:</td><td>Unit Title: Big Data Analytics Applications</td><td>Hours: 7</td></tr>
<tr><td colspan="3">Link Analysis: Web as a Graph, PageRank Concept and Applications, PageRank using MapReduce (high-level) Social Network Analysis: Social Networks as Graphs, Community Detection (conceptual), Recommendation Systems: Content-Based Recommendation, Collaborative Filtering</td></tr>
</table>

Text Book: Seema Acharya & Subhashini Chellappan — Big Data and Analytics (Wiley India)

Reference Books:

1. Radha Shankarmani & M. Vijayalakshmi — Big Data Analytics (Wiley)
2. Tom White — Hadoop: The Definitive Guide (O'Reilly)
3. "Mining of Massive Datasets" — Anand Rajaraman & Jeff Ullman (Cambridge Univ. Press)
4. Mongo DB: The Definitive Guide — Kristina Chodorow & Michael Dirolf (O'Reilly)

Course Code: 6IT235PE	**Course Title: Program Elective Course-III** **(i) Cyber Security**	**L-3 T-0 P-0 C-3**
Course Prerequisite:	Basic knowledge of computer networks, operating systems, and fundamentals of information technology.	
Course Objectives:	1. Describe fundamental concepts, principles, and importance of cyber security. 2. Explain various cyber threats, vulnerabilities, and types of cyber crimes 3. Apply basic cyber and digital forensic concepts for cyber crime investigation. 4. Analyze social engineering techniques and human factors exploited in cyber attacks. 5. Examine cyber laws, ethics, and regulatory frameworks governing cyberspace. 6. Propose suitable preventive measures and best practices to enhance cyber security.	
Course Outcomes (Expected Outcome):	On completion of the course, students will be able to– **CO1:** To develop basic understanding of cyber security. **CO2:** Differentiate among different types of cyber threats and cyber-crimes. **CO3:** Illustrate cyber forensic techniques to identify the criminal activities. **CO4:** Apply forensic analysis tools to recover important evidence for identifying computer crime **CO5:** Distinguish and classify the forms of cybercriminal activity and the technological and social engineering' methods used to undertake such crimes **CO6:** Evaluate the effectiveness of cyber-security, cyber-laws and other countermeasures against cybercrime	
Unit I:	Unit Title: **Introduction To Cyber Security**	Hours: 7
Introduction: Introduction to Cyber Security, Need, Importance and challenges in Cyber Security, Cyberspace, Cyber threats, Cyber-warfare, CIA Triad, Cyber Terrorism, Cyber Security of Critical Infrastructure, Cyber security - Organizational Implications..		
Unit II:	Unit Title: **Cyber Crimes And Hacking**	Hours:7
Overview of Cyber-Attacks and Vulnerabilities, **Types of Threats** – Malware, spyware, Sniffing, Gaining Access, Escalating Privileges, Executing Applications, Hiding Files, Covering Tracks, Worms, Trojans, Viruses, Backdoors. **Types of Cyber Crime -** cyber stalking, forgery, software piracy, cyber terrorism, phishing, computer vandalism, computer hacking, creating and distributing viruses over internet, spamming, cross site scripting, online auction fraud, cyber-squatting, logic bombs, web jacking, internet time thefts, DoS attack, salami attack, data diddling, email spoofing. **Types of Hacker** Hacking and Cracking, Hacking: Ethical issues, Ethical Hacking		
Unit III:	Unit Title: **: Cyber Forensics**	Hours:7
Introduction to Cyber Forensics: What are cyber forensics, cyber forensics investigation process, digital evidence, challenges in cyber forensics; **Web Attack Forensics:** Intrusion forensics, database forensics, preventive forensics; Anti- forensics practices, Anti-forensics detection techniques, Network forensics analysis tools; **Malware Forensics:** Malware types, Malware Analysis, Tools for analysis; **Email Forensics:** e-mail Protocols, e-mail crimes, email forensics; Bitcoin **Forensics:** crypto currency, crimes related to bitcoin; **Case Study:** A detailed case study on cyber forensics and its Investigation Reports		
Unit IV:	Unit Title: **Digital Forensics**	Hours:7
Introduction to Digital Forensics, Cyber Forensics vs Digital Forensics, the role of digital forensics and its environment, Forensic Software and Hardware, properties of digital evidence, recovering and preserving digital evidence, Advanced forensic Tools, selecting and analyzing digital evidence, validating the evidence, Forensic Technology and Practices, Forensic Ballistics and Photography, Face, Iris and Fingerprint Recognition, Audio Video Analysis. **Case Study:** A detailed case study on Digital Forensics.		
Unit V:	Unit Title: **SOCIAL ENGINEERING**	Hours:7

Introduction of social engineering and cyber security, social engineering conceptual evolution, defining social engineering-categories, Phases, attack spiral model, Attack Vendors-social approach, socio-technical approach. Advanced social engineering attack, Phishing Attack, Insider Attack, Identity Theft, Preventing Insider Threats, Social Engineering Targets and Defense Strategies.
Case Study: Phishing and Identity Theft Online Scams.

Unit VI:	Unit Title: **Cyber Ethics And Laws**	Hours:7

Introduction to Cyber Laws, E-Commerce and E-Governance, Certifying Authority and Controller, Offences under IT Act, Computer Offences and its penalty under ISO 27001, IT Act 2000, Positive Aspects and weak areas of ITA 2000, **Digital signatures and the Indian ITA act**, ITA 2008, and International Standards maintained for Cyber Security, Security Audit, Investigation by Investing Agency, Intellectual Property Rights in Cyberspace

Text Books:

1. Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Nina Godbole and Sunil Belapure, Wiley INDIA. ISBN 978-81-265-2179-1

2. Practical Cyber Forensics an Incident-Based Approach to Forensic Investigations, Niranjan Reddy, Apress, ISBN-13: 978-1-4842-4459-3

3. Practical Digital forensics – Richard Boddingtion, PACKT Publishing ISBN 978-1-78588-710-9

Reference Books:

1. William Stallings, Computer Security: Principles and Practices, Pearson 6th Ed, ISBN: 978-0- 13- 335469-0
2. Bernard Menezes, Network Security and Cryptography, Cengage Learning, ISBN-978-81- 315-1349- 1
3. Dr. V.K. Pachghare, Cryptography and Information security, PHI, Second edition, ISBN- 978- 81- 203-5082-3

Course Code: 6IT235PE	**Course Title: Program Elective Course –III (ii) Cognitive Technology**	**L-3, T-0, P-0, C-3**
Course Prerequisite:	Basic knowledge of Artificial Intelligence, Programming and Data Structures.	
Course Objectives:	1. To introduce concept of cognitive technologies and important approaches of cognitive technologies. 2. To learn and analyze key concept of cognitive technologies. 3. To understanding of innovation concepts, terminology, current and future trends in cognitive technologies. 4. To introduces students to IBM Watson platform, an artificially intelligent computer system capable of answering questions posed in natural language, 5. To developed in IBM's Deep QA project.	
Course Outcomes (Expected Outcome):	1. Describe the Cognitive computing and principles of cognitive systems. 2. Identify role of Natural Language Processing in cognitive system. 3. Outline application of advanced analytics in cognitive computing. 4. Justify role of Cloud and Distributed Computing in Cognitive Computing. 5. Assess the process of building a Cognitive Application. 6. Identify the Emerging Areas and Future Applications of Cognitive Computing.	
Unit I:	**Foundation of Cognitive Computing & Design: Principle of Cognitive Systems**	Hours: 7
The Foundation of Cognitive Computing: Cognitive Computing as a New Generation, The Uses of Cognitive Systems, What Makes a System Cognitive, Gaining Insights from Data, Domains Where Cognitive Computing Is Well Suited, Artificial Intelligence as the Foundation of Cognitive Computing, Understanding Cognition, Two Systems of Judgment and Choice, Understanding Complex Relationships Between Systems, The Elements of a Cognitive System, Infrastructure and Deployment Modalities. **Design Principles for Cognitive Systems:** Components of a Cognitive System, Building the Corpus, Bringing Data into the Cognitive System, Machine Learning, Hypotheses Generation and Scoring, Presentation and Visualization Services.		
Unit II:	**NLP and Big Data in Cognitive System**	Hours: 7
Natural Language Processing in Support of a Cognitive System: The Role of NLP in a Cognitive System, Semantic Web, Applying Natural Language Technologies to Business Problems. **The Relationship Between Big Data and Cognitive Computing:** Dealing with Human-Generated Data, Defining Big Data, The Architectural Foundation for Big Data, Analytical Data Warehouses, Hadoop, Data in Motion and Streaming Data, Integration of Big Data with Traditional Data.		
Unit III:	**Knowledge Representation and Advance Analytics in Cognitive Computing**	Hours: 7

Representing Knowledge in Taxonomies and Ontologies: Representing Knowledge, developing a Cognitive System, Defining Taxonomies and Ontologies, Explaining How to Represent Knowledge, Models for Knowledge Representation. **Applying Advanced Analytics to Cognitive Computing:** Advanced Analytics Is on a Path to Cognitive Computing, Key Capabilities in Advanced Analytics, Using Advanced Analytics to Create Value, Impact of Open-Source Tools on Advanced Analytics.

Unit IV:	**Role of Cloud and Distributed Computing in Cognitive Computing**	Hours: 7

The Role of Cloud and Distributed Computing in Cognitive Computing: Leveraging Distributed Computing for Shared Resources, Why Cloud Services Are Fundamental to Cognitive Computing Systems, Characteristics of Cloud Computing, Cloud Computing Models, Delivery Models of the Cloud, Managing Workloads, Security and Governance, Data Integration and Management in the Cloud.

The Business Implications of Cognitive Computing: Preparing for Change, Advantages of New Disruptive Models, What Does Knowledge Mean to the Business, The Difference with a Cognitive Systems Approach, Meshing Data Together Differently, Using Business Knowledge to Plan for the Future, Answering Business Questions in New Ways, Building Business Specific Solutions, Making Cognitive Computing a Reality, How a Cognitive Application Can Change a Market.

Unit V:	**IBM Watson and Process of Building a Cognitive Application**	**Hours: 7**

IBM's Watson as a Cognitive System: Watson Defined, Advancing Research with a "Grand Challenge", Preparing Watson for Jeopardy, Preparing Watson for Commercial Applications, The Components of Deep QA Architecture.

The Process of Building a Cognitive Application: The Emerging Cognitive Platform, Defining the Objective, Defining the Domain, Understanding the Intended Users and Defining their Attributes, Defining Questions and Exploring Insights, Creating and Refining the Corpora, Training and Testing.

Building a Cognitive Healthcare Application: Foundations of Cognitive Computing for Healthcare, Constituents in the Healthcare Ecosystem, Learning from Patterns in Healthcare Data, Building on a Foundation of Big Data Analytics, Cognitive Applications across the Healthcare Ecosystem, Starting with a Cognitive Application for Healthcare, Using Cognitive Applications to Improve Health and Wellness, Enhance the Electronic Medical Record and to Improve Clinical Teaching.

Unit VI:	**Emerging Areas and Future Application**	**Hours: 7**

Smarter Cities: Cognitive Computing in Government: How Cities Have Operated, The Characteristics of a Smart City, The Rise of the Open Data Movement Will Fuel Cognitive Cities, The Internet of Everything and Smarter Cities, Understanding the Ownership and Value of Data, Smarter Approaches to Preventative Healthcare, Building a Smarter Transportation Infrastructure, Using Analytics to Close the Workforce Skills Gap, Creating a Cognitive Community Infrastructure, The Next Phase of Cognitive Cities.

Emerging Cognitive Computing Areas: Characteristics of Ideal Markets for Cognitive, Computing Vertical Markets and Industries.

Future Applications for Cognitive Computing: Requirements for the Next Generation, Technical Advancements That Will Change the Future of Cognitive Computing, What the Future Will Look Like, Emerging Innovations

Text Book: Judith Hurwitz, Marcia Kaufman and Adrian Bowles, "Cognitive Computing and Big Data Analytics", publication John Wiley & Sons, Inc, 2015.

Reference Books:

1. José Luis Bermúdez, Cognitive Science: An Introduction to the Science of the Mind, publication Cambridge University Press, New York, Second Edition.
2. Jay Friedenberg and Gordon Silverman, Cognitive Science: An Introduction to the Study of Mind, Sage Publications, Inc. London, 2014.
3. Huimin Lu (Editor), Cognitive Internet of Things: Frameworks, Tools and Applications, Springer Nature Switzerland AG 2020.
4. Danish Contractor and Aaditya Telang (Editors), Applications of Cognitive Computing Systems and IBM Watson, 8th IBM Collaborative Academia Research Exchange, publication Springer Nature Singapore Pte Ltd., 2017.
5. S. Bird, E. Klein, E. Loper (2009), Natural Language Processing with Python, O' Reilly Media.

Course Code: 6IT235PE	**Course Title: Program Elective Course –III (iii) Software Testing**	**L:3 T:0 P:0 C:3**
Course Prerequisite:	● Basic knowledge of Software Engineering concepts ● Understanding of Software Development Life Cycle (SDLC) ● Familiarity with at least one programming language (such as Java, C++, or Python) ● Basic knowledge of Object-Oriented Programming concepts ● Understanding of databases and system fundamentals ● Basic analytical and logical reasoning skills	
Course Objectives:	1. To understand fundamental concepts, principles, and life cycle of software testing.	

	2. To apply various testing techniques and strategies for conventional and object-oriented software. 3. To design and document effective test plans, test cases, and traceability matrices. 4. To execute testing processes systematically, including regression and defect management. 5. To evaluate software quality using testing metrics and debugging techniques.	
Course Outcomes (Expected Outcome):	Students successfully completing this course will be able to: 1. Understand Various Software Process models. 2. Evaluate various software metrics. 3. Analyze Software Configuration and risk management Technique. 4. Classify and analyze various software testing techniques.	
Unit I:	Unit Title: **Introduction to Software Testing**	Hours: 7
Testing Principles, Best Practices in Testing, Skills for Testing, Various Task Involved In Testing, Difference between Verification & Validation, Difference between QA &QC, Test Strategies for Conventional and Object Oriented Software, Unit Testing, Integration Testing, Validation Testing, Alpha and Beta Testing, System Testing, Recovery Testing, Security Testing, Stress Testing, Performance Testing, Metrics for Source Code, Metrics for Testing, Debugging Process, Debugging Strategies.		
Unit II:	Unit Title: **Testing Techniques**	Hours: 7
Software Testing Fundamentals, Black Box and White Box Testing, Basis Path Testing, Flow Graph Notation, Independent Program Paths, Graph Matrices, Control Structure Testing, Condition Testing, Data Flow Testing, Loop Testing, Graph Based Testing Methods, Equivalence Partitioning, Boundary Value Analysis, Object Oriented Testing Methods: Applicability of Conventional Test Case Design Methods, Fault-Based Testing, Scenario-Based Testing, Random Testing and Partition Testing for Classes, Inter Class Test Case Design.		
Unit III:	Unit Title: **Types of Non-functional Testing**	Hours: 7
Security Testing, Recovery Testing, Configuration testing, Compatibility Testing, Installation Testing, Performance Testing, Parallel Testing, Volume Testing, Internationalization Testing, Localization Testing		
Unit IV:	Unit Title: **Process of Testing**	Hours: 7
Test Planning, Test Analysis, Test Design Construction, and verification, Testing Cycles, Final Testing and Implementation, Post Implementation		
Unit V:	Unit Title: **Test Design for Functional Testing**	Hours: 7
Introduction To Test Design, Inputs For Test Design Activity, Test scenarios, Test Cases, Test Case Management, Best Practices Of Test Cases, Test Data, Black Box Test Technique, Requirement Traceability Matrix(RTM)		
Unit VI:	Unit Title: **Test Execution and Defect Management**	Hours: 7
Test Execution Cycle, Entry Criteria For Test Execution, Smoke /Sanity Testing, Test Execution, Retesting And Regression Testing, Ending The Test Execution Activity, Test Closure Activities. Defect /Bug, Causes For Defect, Defect Reporting, Defect Attributes, Severity And Priority, Defect Life Cycle, Advantages Of Defect Tracking		
Text Books: 1. Software Engineering by Rajib Mall, PHI 2. Software Testing: A Craftsman's Approach, by Paul C. Jorgensen, Third Edition		
Reference Books: 1. An Integrated Approach to Software Engineering, Second Edition Pankaj Jalote Narosa 2. Software Engineering: Theory and Practice, Second Edition S.L. Pfleeger, J.M. Atlee Pearson Education 3. Software Engineering, Roger Pressman, Tata Mcgraw Hill		

Course Code: 6IT236PC	**Course Title: Compiler Design Lab**	**L:0 T:0 P:2 C:1**
Course Prerequisite:	Basic knowledge of Programming in C, Discrete Mathematics, Operating Systems and Theory of Computation.	
Course Objectives:	1. To know the basic components of a Compiler. 2. To implement Lexical Analyzer and Syntax Analyzer using various compiler Tool. 3. To implement various parsing methods and code optimization techniques.	
Course Outcomes (Expected Outcome):	1. Implement lexical analyzer in C language for validating tokens from input string and construction of regular expression for given regular language. 2. Demonstrate the complier tools like lex and Yacc for designing lexical analyzer and parser. 3. Design and implement SLR, CLR and LALR parser and three address code.	

Minimum 8 to 10 practical's are to be conducted based on theory subject ***6IT231PC Compiler Design*** Syllabus.

Course Code: 6IT237PC	**Course Title: Design and Analysis of Algorithm Lab**	**L-0 T-0 P-2 C-1**
Course Prerequisite:	Discrete Structure & Graph Theory, Data Structure	
Course Objectives:	1 To understand the fundamentals of structured programming, control constructs, procedures, functions, and recursion. 2 To analyze the efficiency of algorithms using order notations and execution time estimation techniques. 3 To design and implement algorithms using various strategies such as iterative methods, recursion, greedy method, dynamic programming, backtracking, and branch-and-bound. 4 To apply algorithmic techniques to solve classical problems such as Knapsack, Traveling Salesman, Minimum Spanning Trees, Shortest Path, Matrix Multiplication, LCS, and Graph Coloring. 5 To evaluate worst-case and average-case behavior of algorithms and compute time complexity for sorting and other computational problems.	
Course Outcomes (Expected Outcome):	1. Apply structured programming concepts and develop modular programs using functions and recursion. 2. Analyze algorithms using asymptotic notations (Big-O, Big-Ω, Big-Θ) and estimate execution time. 3. Design and implement efficient solutions using greedy, dynamic programming, backtracking, and branch-and-bound techniques. 4. Solve graph and optimization problems such as MST (Prim's & Kruskal's), Dijkstra's algorithm, Knapsack, TSP, Multistage Graphs, and LCS. 5. Compare algorithms based on time and space complexity and evaluate worst and average case performance for various sorting and computational problems. 6. Develop the ability to model real-world computational problems, choose appropriate algorithmic strategies, and justify the selected approach based on complexity analysis and performance evaluation.	

Suggested List of Experiments covering entire syllabus:

1. Structured Programming using Control Constructs (if–else, switch, loops).
2. Procedures and Functions – Parameter Passing Techniques.
3. Recursive Algorithms – Factorial, Fibonacci, Tower of Hanoi.
4. Iterative Algorithm Design and Efficiency Analysis (Linear & Binary Search).
5. Order Notations and Execution Time Estimation of Algorithms.
6. Multiplication Algorithms and their Complexity Analysis.
7. Fractional Knapsack Problem (Greedy Method).
8. Job Sequencing with Deadlines (Greedy Algorithm).
9. Minimum Spanning Tree – Prim's Algorithm.
10. Minimum Spanning Tree – Kruskal's Algorithm.
11. Dijkstra's Single Source Shortest Path Algorithm.
12. Multistage Graph – Shortest Path (Dynamic Programming).
13. Traveling Salesman Problem (TSP).
14. Matrix Chain Multiplication (Dynamic Programming).
15. Longest Common Subsequence (LCS).
16. Optimal Polygon Triangulation.
17. Convex Hull Algorithms (Graham Scan / Jarvis March).
18. 8-Queens Problem using Backtracking.
19. Graph Coloring using Backtracking.
20. Branch and Bound – 0/1 Knapsack or TSP.
21. Sorting Algorithms – Complexity Analysis (Bubble, Selection, Insertion, Merge, Quick).

Course Code: 6IT238PC	**Course Title: Artificial Intelligence Lab**	**L-0 T-0 P-2 C-1**
Course Prerequisite:	Basic Knowledge of Programming, Introduction to Python, Discrete Mathematics and Data Structure	
Course Objectives:	1. To Formulate Problems and Evaluation of Informed and Uniformed Search Strategies 2. To understand the various searching techniques, constraint satisfaction problem and example problems - game playing techniques. 3. Learn knowledge representation technique, reasoning technique. 4. To understand Predicate Logic and Adversarial Search & Games.	
Course Outcomes (Expected Outcome):	1. Design and implement search strategies 2. Develop logical reasoning systems. 3. Investigate various knowledge representation issues. 4. Use predicate logic for knowledge representation. 5. Design adversarial search and game theory.	
Minimum 8 to 10 practical's are to be conducted based on theory subject ***6IT233PC Artificial Intelligence*** Syllabus.		

Course Code: 6IT239MD	**Course Title: MDM- V: Software Project Management**	**L-2 T-0 P--0 C-2**
Course Prerequisite:	Software Engineering, Programming Fundamentals	
Course Objectives:	1. To outline the need for Software Project Management. 2. To highlight different techniques for software cost estimation and activity planning. 3. To outline appropriate software process models and agile methods for effective software project development and management.	
Course Outcomes (Expected Outcome):	1. Understand the knowledge of stepwise project planning. 2. Analyze Effort and Cost estimation techniques 3. Apply network planning models to create and manage realistic project schedules. 4. Apply Earned Value Analysis	
Unit I:	**Project Evaluation and Planning**	**Hours: 9**
Importance of Software Project Management – Activities – Methodologies – Categorization of Software Projects – Setting objectives – Management Principles – Management Control – Project Portfolio Management – Cost-benefit evaluation technology – Risk evaluation – Strategic Program Management – Stepwise Project Planning.		
Unit II:	**Project Life Cycle and Effort Estimation**	**Hours: 9**
Software Process and Process Models – Choice of Process Models – Rapid Application Development – Agile Methods – Dynamic System Development Method – Extreme Programming – Managing Interactive Processes – Basics of Software Estimation – Effort and Cost Estimation Techniques.		
Unit III:	**Unit Title: Activity Planning, Risk Management and Project Management & Control**	**Hours: 9**
Objectives of Activity Planning – Project Schedules – Activities – Sequencing and Scheduling – Network Planning Models. Risk Identification – Assessment – Risk Planning – Risk Management. Framework for Management and Control – Collection of Data – Visualizing Progress – Cost Monitoring – Earned Value Analysis – Prioritizing Monitoring – Project Tracking – Change Control – Software Configuration Management		
Text Book: 1. Bob Hughes, Mike Cotterell and Rajib Mall, *Software Project Management*, Fifth Edition, McGraw Hill, New Delhi, 2012.		
Reference Books: 1. Robert K. Wysocki, *Effective Software Project Management*, Wiley Publication, 2011. 2. Walker Royce, *Software Project Management*, Addison-Wesley, 1998. 3. Gopalaswamy Ramesh, *Managing Global Software Projects*, McGraw Hill Education (India), Fourteenth Reprint, 2013.		

Course Code: 6IT240VS	**Course Title: Computer Skills - II**	**L:1 T:0 P:2 C:2**
Course Prerequisite:	Basic understanding of HTML, CSS, and JavaScript, OOPs concept.	
Course Objectives:	1. To Explain the fundamentals of Angular framework and TypeScript for modern web application development. 2. To Apply Angular project structure concepts including components, templates, services, directives, and TypeScript compilation. 3. To Develop dynamic and reusable UI components using data binding and event handling techniques. 4. To Implement template-driven and model-driven forms with controlled HTML rendering in Angular applications. 5. To Utilize Observables and integrate Angular applications with backend APIs for asynchronous data communication. 6. To Design and build a complete Single Page Application (SPA) using routing and implement CRUD operations with Firebase integration. 7.	
Course Outcomes (Expected Outcome):	After completion of the course, students will be able to: 1. Programming and Application Development using Angular and TypeScript 2. Handling Project Files and TypeScript Compilation, Components Templates Services Directives 3. Data Rendering, Events Handling, Building Reusable Components 4. Controlling Rendering of HTML, Creating Template Driven Forms and Model Driven Forms 5. Using Observables and Communicating with Server using APIs 6. Building Single Page Application with Routing and Developing applications integrating CRUD with Firebase	

Minimum eight experiments/programming assignments covering the following aspects:

1. Introduction to Angular
2. Angular Environment Set up
3. Creating Angular Project and basic introduction about project structure / directory.
4. Understanding Components and how to create components in Angular
5. Understanding of data binding in Angular component and view files.
6 Understanding and use of different types of Angular directives
7. Understanding of modules and routing in angular.
8. Understanding of services and component ‘s life cycle method
9. Understanding of package. json file in Angular Project.
10. Understanding of how to fetch data from the API using services.

List of Experiments:

1. Introduction to Angular and TypeScript
2. Working with Project Files and TypeScript Compilation in Angular.
3. Demonstration of Components, Templates, Services, and Directives
4. Data rendering and events handling in angular
5. Building reusable components
6. Controlling and rendering of HTML
7. Designing and working with Template Driven Forms
8. Designing and working with Model Driven Forms
9. Working with Observables
10. Communicating with Server using APIs
11. Building Single Page Application with Routing
12. Performing CRUD with Firebase in Angular

Text Book: Beginning Angular with Typescript by Greg Lim

Course Code: 6IT248DH1	**Course Title: Databases with Java**	**L:4 T:0 P:0 C:4**
Course Prerequisite:	• Programming in Java • Basic knowledge of DBMS • Basic HTML	
Course Objectives:	1. To understand relational database concepts and SQL. 2. To develop Java programs that connect to databases using JDBC. 3. To build dynamic web applications integrated with databases. 4. To apply transaction management and security techniques. 5. To develop enterprise-level database-driven Java applications.	

Course Outcomes (Expected Outcome):	After completion of the course, students will be able to: 1. Design and query relational databases using SQL 2. Develop JDBC-based database applications 3. Implement transaction management 4. Build servlet/JSP based database web applications 5. Apply MVC and DAO design patterns	
Unit I:	Unit Title: Fundamentals of Databases and Java	Hours: 8
Introduction to Database Systems, DBMS vs RDBMS, Data Models and ER Concepts, Relational Model Concepts, SQL Basics (DDL, DML, DCL, TCL), Introduction to Java for Database Programming, Java Architecture Overview (JVM, JRE, JDK), Introduction to JDBC Architecture, Types of JDBC Drivers, Setting up Java–Database Environment		
Unit II:	Unit Title: JDBC Programming	Hours: 8
JDBC API Overview, Steps to Connect Java with Database, Loading Driver, Establishing Connection, Creating Statements, Executing Queries, Processing Result Set, Closing Resources, Statement vs Prepared Statement vs Callable Statement, Handling SQL Exceptions, Transaction Management in JDBC, Batch Processing, Metadata (Database Meta Data & Result Set Meta Data)		
Unit III:	Unit Title: Advanced JDBC and Database Operations	Hours: 8
CRUD Operations using JDBC, Scrollable and Updatable Result Sets, Stored Procedures in Java, Connection Pooling Concepts, Performance Optimization Techniques, Handling Large Objects (BLOB & CLOB), Security in JDBC Applications, Preventing SQL Injection		
Unit IV:	Unit Title: Servlets and Database Integration	Hours: 8
Introduction to Servlets, Servlet Life Cycle, Handling Client Requests (GET & POST), Form Processing and Validation, Integrating Servlets with JDBC, MVC Architecture in Web Applications, Session Management (Cookies, URL Rewriting, HttpSession), Error Handling in Web Applications		
Unit V:	Unit Title: JSP and Enterprise Database Applications	Hours: 8
Introduction to JSP, JSP Life Cycle, JSP Elements (Directives, Scriptlets, Expressions), Implicit Objects, JSTL Basics, Custom Tags Overview, JSP with JDBC Integration, Building Database-Driven Web Applications		
Unit VI:	Unit Title: Modern Database Connectivity and Best Practices	Hours: 8
Introduction to DAO Pattern, MVC with Java Web Applications, Introduction to ORM Concepts, Overview of JPA and Hibernate, Basic Spring JDBC Concept, Connection Pooling using Data Source, Security and Authentication in DB Applications, Deployment Concepts (WAR file, Web Server basics)		
Text Book: Head First Servlets & JSP — Bryan Basham, Kathy Sierra & Bert Bates, O'Reilly Media		
Reference Books: 1. JDBC API Tutorial and Reference — Maydene Fisher, Jon Ellis, Jonathan Bruce 2. Core Servlets and JavaServer Pages — Marty Hall & Larry Brown		

Course Code: 6IT248DH2	**Course Title: Intelligent Systems**	**L:4 T:0 P:0 C:4**
Course Prerequisite:	• Basic Programming (e.g., Python/Java/C++) • Discrete Mathematics • Probability and Statistics fundamentals	
Course Objectives:	1. To understand fundamental concepts of intelligent systems and agent models. 2. To learn major search and reasoning techniques. 3. To study knowledge representation strategies. 4. To introduce machine learning and neural models. 5. To explore real-world intelligent applications..	
Course Outcomes (Expected Outcome):	After completion of the course, students will be able to: 1. Describe the architecture and components of intelligent systems 2. Implement basic search and reasoning techniques 3. Apply knowledge representation methods 4. Use learning algorithms for classification & pattern recognition 5. Understand neural networks and hybrid intelligent models	
Unit I:	Unit Title: Introduction to Intelligent Systems	Hours: 8
What Is Intelligence? Natural vs Artificial, Evolution of Intelligent Systems, Intelligent Agents and Environments, Types of Agents (Reactive, Deliberative, Hybrid), Intelligent System Architecture, Problem Solving and Search in AI, Intelligent System Life Cycle, Applications of Intelligent Systems		
Unit II:	Unit Title: Search and Problem Solving Techniques	Hours: 8

State Space Search Concepts, Uninformed Search: BFS, DFS, Informed (Heuristic) Search: Best-First, A*, Local Search Techniques (Hill Climbing, Simulated Annealing), Constraint Satisfaction Problems (CSP), Problem Formulation for Intelligent Systems		
Unit III:	Unit Title: Knowledge Representation and Reasoning	Hours: 8
Knowledge Representation Techniques, Propositional Logic, First-Order Logic, Semantic Networks, Frames and Inheritance, Rule-Based Systems and Production Rules, Inference Mechanisms, Expert Systems Overview		
Unit IV:	Unit Title: Machine Learning Fundamentals	Hours: 8
Introduction to Machine Learning, Learning Paradigms: Supervised, Unsupervised, Reinforcement, Decision Trees, Bayesian Learning, Instance-Based Learning (k-NN), Neural Networks Basics, Evaluating Learning Models (Accuracy, Precision, Recall)		
Unit V:	Unit Title: Neural Networks and Intelligent Models	Hours: 8
Single-Layer and Multi-Layer Perceptron, Backpropagation Algorithm, Radial Basis Function Networks Fuzzy Logic and Fuzzy Systems, Genetic Algorithms & Evolutionary Computation, Hybrid Intelligent Systems (Neuro-Fuzzy, Genetic-Neural)		
Unit VI:	Unit Title: Applications of Intelligent Systems	Hours: 8
Intelligent Robotics, Natural Language Processing (NLP) Basics, Computer Vision Overview, Intelligent Decision Support Systems, Data Mining and Pattern Recognition, Intelligent Agents in Real-World Systems, Emerging Trends in Intelligent Systems (AI in IoT, Smart Cities)		
Text Book: 1. Intelligent Systems: Principles, Paradigms and Pragmatic Approaches, Authors: Vinod Vasudevan, Ashwin Ramanujam, Anil Kumar Publisher: McGraw-Hill Education		
Reference Books: 1. Artificial Intelligence: A Modern Approach — Stuart Russell & Peter Norvig, Pearson. 2. Artificial Intelligence — Elaine Rich, Kevin Knight & Shivashankar B. Nair, McGraw-Hill. 3. Machine Learning — Tom M. Mitchell, McGraw-Hill		

****