SANT GADGE BABA AMRAVATI UNIVERSITY, AMRAVATI

Draft Syllabus of Semester V & VI of M.C.A. 5 Years Integrated Course

SEMESTER – V

5IMC301 DATA COMMUNICATION & NETWORKING

Course Objectives:

1. To introduce the fundamental concepts of data communication systems and network types.
2. To develop understanding of layered network architecture using OSI and TCP/IP models.
3. To explain core networking protocols used in modern Internet communication.
4. To build basic competency in IP addressing and subnetting.
5. To understand reliable data transmission, flow control and congestion concepts.
6. To familiarize students with wireless communication basics and emerging networking technologies.

Course Outcomes:

After completion of this course Students will be able to

1. Understand the components of data communication systems and various types of networks.
2. Describe the OSI and TCP/IP models and explain the function of each layer.
3. Analyze fundamental networking protocols such as Ethernet, IP, TCP, UDP, HTTP and DNS.
4. Apply IPv4 addressing and perform basic subnetting calculations.
5. Explain concepts of error detection, flow control and congestion control in networks.
6. Acquire introductory knowledge of wireless networking and modern communication technologies.

Unit I : Data Communication Fundamentals (8 Hours)

Components of Data Communication System; Characteristics of effective communication; Analog and Digital Signals; Transmission Modes – Simplex, Half Duplex, Full Duplex; Transmission Media – Guided and Unguided; Basic Modulation concepts; Multiplexing – FDM, TDM; Switching – Circuit and Packet Switching; Network Topologies; Types of Networks – LAN, MAN, WAN; Network Reference Models – ISO-OSI and TCP/IP model.

Unit II: Data Link Layer and LAN Technologies (7 Hours)

Functions of Data Link Layer; Framing; Error Detection (Parity, CRC); Flow Control; Stop-and-Wait ARQ; Multiple Access Protocols – ALOHA, CSMA/CD; Ethernet (IEEE 802.3) basics; MAC Addressing; ARP; Networking Devices – Hub, Bridge, Switch.

Unit III: Network Layer and IP Addressing (7 Hours)

Functions of Network Layer; Internet Protocol – IPv4 Addressing, Classes, Subnetting; Private IP and NAT; Introduction to IPv6; Concept of Routing; Distance Vector and Link State; Overview of RIP, OSPF, BGP; Role of Router; ICMP and DHCP.

Unit IV: Transport Layer (7 Hours)

Transport Layer Services; Port Numbers; Multiplexing and Demultiplexing; UDP – Features and Applications; TCP – Segment Structure, Three-way Handshake; Reliable Data Transfer; Flow Control; Introduction to Congestion Control.

Unit V: Application Layer Protocols (7 Hours)

Client-Server Model; Socket Interface; HTTP – Working, Request/Response, Persistent and Non-persistent Connections; HTTPS; FTP – Service Model; Electronic Mail – SMTP, POP, IMAP; DNS – Services and Working Mechanism.

Unit VI: Wireless Communication (7 Hours)

Advantages and Applications of Wireless Communication; Characteristics of Wireless Signals – Propagation and Interference; Cellular System and Frequency Reuse; Wireless LAN – IEEE 802.11; Bluetooth, Basic architecture of 5G; Role of Edge Computing; Applications in IoT and Smart Systems.

Text Book: "Data Communications and Networking", Behrouz Forouzan, 6th Edition, 2022, McGraw Hill.

Reference Books:

1. "Computer Networking: A Top-Down Approach", James Kurose and Keith Ross, 8th Edition, 2021, Pearson.
2. "Computer Networks", Andrew Tanenbaum, Nick Feamster, and David Wetherall, 6th Edition, 2021, Pearson.
3. "Data and Computer Communications", William Stallings, 10th Edition, 2013, Pearson.

****

5IMC302 CLIENT SERVER COMPUTING

Course Objectives

1. Understand client–server architecture and web development concepts.
2. Learn HTML, CSS, JavaScript, and web technologies.

3. Implement database connectivity using JDBC.
4. Develop dynamic applications using Servlets.
5. Create web applications using JSP.
6. Build and deploy applications using Spring Boot.

Course Outcomes

After completion of this course, students will be able to:

1. Explain client–server and web architecture concepts.
2. Develop web pages using HTML, CSS, and JavaScript.
3. Implement database-driven applications using JDBC.
4. Design server-side applications using Servlets.
5. Create dynamic web applications using JSP.
6. Develop and deploy applications using Spring Boot

Unit 1: Introduction to Web Design (6 hours)

History and Evolution of the Web, Basics of Web Design and Development, Web Standards (W3C), Introduction to Client-Side vs. Server-Side Scripting, Overview of Tools and Technologies (Browsers, Editors, Frameworks), Basic Web Project Structure.

Unit II – Introduction to Client–Server Computing (7 Hrs)

Evolution of computing: Centralized, Distributed, and Client–Server models, Two-tier and Three-tier architecture, Web architecture overview, Internet, Intranet, Extranet, Web servers and Application servers, HTTP protocol: Request–Response cycle, Introduction to REST architecture

Unit III – JDBC (7 Hrs)

Introduction to JDBC Architecture, Types of JDBC Drivers, JDBC Concepts, JDBC API, Driver Manager, Connection, Statement, Prepared Statement, Callable Statement and Result Set classes with relevant methods, Types of Result Sets. Handling queries, inserts, deletes and updates to database. Displaying the query results. Stored Procedures. Connecting Java with MySQL/Postgre SQL, JDBC Program

Unit IV – Servlets (8 Hrs)

Introduction to Servlets, Servlet Life Cycle, Servlet API and Interfaces, Handling HTTP Requests (GET & POST),Session Management (Cookies, Http Session, URL Rewriting),Request Dispatcher and Redirect, Form handling using Servlets, MVC architecture using Servlets, JDBC Program Servlets

Unit V – JSP (8 Hrs)

Introduction to JSP; JSP life cycle; JSP directives; JSP scripting elements; implicit objects; Expression Language (EL); JSTL basics; error handling in JSP; and integration of JSP with Servlets. JSP basics; differences between Servlets and JSP; running a simple JSP; understanding the servlet code generated from a JSP; JSP implicit objects; JSP syntax for various elements; and developing JSP beans.

Unit VI – Spring Boot (8 Hrs)

Introduction to Spring Framework, Spring Boot Architecture, Creating Spring Boot Project,
Dependency Injection & Annotations, RESTful Web Services, Connecting Spring Boot with Database (JPA/Hibernate),CRUD Application using Spring Boot, Basic Security Concepts.

Text Books:

1. Java Server Programming (Java EE 7 / J2EE 1.7) Black Book (2014) – Kogent Learning Solutions Inc.
2. Core Servlets and JavaServer Pages: Core Technologies – Marty Hall and Larry Brown, Java 2 Platform Enterprise Edition Series, Prentice Hall.
3. Spring Boot Reference Guide (Version 2.0.0.BUILD-SNAPSHOT)
   Phillip Webb, Dave Syer, Josh Long, Stéphane Nicoll, Rob Winch, Andy Wilkinson, Marcel Overdijk, Christian Dupuis, Sébastien Deleuze, Michael Simons, and Vedran Pavić.

Reference Book: Java EE Cookbook – Elder Moraes, Packt Publishing Limited, 9 April 2018.

****

5IMC303 SOFTWARE ENGINEERING

Course Objectives: After completing this course, students will be able to:

- Explain fundamental concepts and importance of software engineering.
- Describe software development life cycle models and their phases.
- Identify software requirements and prepare SRS documents.
- Apply basic software design principles and UML diagrams.
- Analyze different software testing methods and levels.
- Understand software maintenance and project management practices.

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

- Define and explain software engineering principles and processes.
- Compare and select appropriate SDLC models for given scenarios.
- Develop basic Software Requirement Specification (SRS).
- Design simple software models using UML diagrams.
- Evaluate software quality using testing techniques.
- Apply software maintenance and project planning concepts in real projects.

Unit 1: Introduction to Software Engineering (7 Hrs)
Definition and importance of Software Engineering, Software characteristics, Software Crisis, Software development Process, Role of Software Engineer
Unit 2: Software Development Life Cycle (SDLC) Models (7 Hrs)
Waterfall Model, Incremental Model, Spiral Model, Agile Model (Introduction), Comparison of SDLC Models
Unit 3: Software Requirements Engineering (7 Hrs)
Requirement Engineering Process, Functional vs Non-Functional Requirements, Requirement Gathering Techniques, Software Requirement Specification (SRS), Characteristics of Good SRS
Unit 4: Software Design (7 Hrs)
Design Principles, Modular Design, Coupling and Cohesion, Structured Design, Introduction to UML Diagrams (Use Case Diagram, Class Diagram)
Unit 5: Software Testing (7 Hrs)
Testing Fundamentals, Levels of Testing (Unit Testing, Integration Testing, System Testing), Testing Methods (Black Box Testing, White Box Testing), Debugging
Unit 6: Software Maintenance & Project Management (7 Hrs)
Software Maintenance Types, Software Configuration Management, Risk Management, Project Planning, Cost Estimation (Basic idea)

Text Books:
1. **Roger S. Pressman & Bruce R. Maxim -** *Software Engineering: A Practitioner's Approach* (McGraw-Hill)
   Ian Sommerville- *Software Engineering* (Pearson)

Reference Books:
1. **Pankaj Jalote-** *An Integrated Approach to Software Engineering* (Springer)
   K.K. Aggarwal & Yogesh Singh- *Software Engineering* (New Age International)
2. **Rajib Mall-** *Fundamentals of Software Engineering* (PHI Learning)
3. **Shari Lawrence Pfleeger & Joanne Atlee-** *Software Engineering: Theory and Practice* (Pearson)
4. **James F. Peters & Witold Pedrycz-** *Software Engineering: An Engineering Approach* (Wiley)
5. **Karl Wiegers & Joy Beatty-** *Software Requirements* (Microsoft Press)

****

5IMC304 Programme Elective Course – I (i) Systems Biology

Course Objectives:
1. Introduce students to systems-level understanding of biological processes
2. Explain biological networks and pathways in simple computational terms
3. Provide basic mathematical and computational tools used in systems biology
4. Introduce omics data and its analysis
5. Prepare students for advanced study in Machine Learning in Biomedical and Health Informatics

Course Outcomes:
1. Understand biological systems from a systems perspective
2. Represent biological interactions using networks and pathways
3. Apply simple mathematical and computational models to biological problems
4. Perform basic preprocessing of biomedical and omics data
5. Connect systems biology concepts with machine learning applications in healthcare

Unit I: Introduction to Systems Biology

What is Systems Biology? Scope and applications, Difference between reductionist and systems approach, Levels of biological organization (cell, tissue, organ, system), Central Dogma (DNA–RNA–Protein) from systems view, Types of biological data: genomic, proteomic, metabolomic, clinical.

Unit II: Biological Networks and Pathways

Gene regulatory networks, Protein–protein interaction networks, Metabolic and signaling pathways, Basics of graph theory (nodes, edges, degree, centrality), Pathway databases: KEGG, Reactome, BioCyc, Network visualization and interpretation, Introduction to network inference.

Unit III: Modeling of Biological Systems
Dynamic behavior of biological systems, Continuous and discrete models, Introduction to Ordinary Differential Equations (ODEs), Feedback loops and regulatory circuits, Steady-state and stability concepts, Parameter estimation (basic idea), Simple case studies.

Unit IV: Omics Data and Integration

Genomics, Transcriptomics, Proteomics, Metabolomics, Characteristics of omics data (high dimension, noise), Data preprocessing and normalization, Feature representation of biological data, Introduction to multi-omics integration, Challenges in systems biology data analysis

Unit V: Computational Tools and Simulation

Computational tools overview, Modeling using Python , Simulation of simple biological systems, Model validation and interpretation, Mechanistic vs data-driven models, Reproducible workflows.

Unit VI: Systems Biology and Machine Learning
Limitations of traditional systems biology, Need for machine learning in biology, Applications in health informatics: Disease prediction, Drug response modeling, Biomarker discovery, Connecting systems biology with ML: Networks → Graph-based learning, Time-series data → Sequential models, Omics data → Feature-based learning.

Text Books:

1. Hiroaki Kitano, *Systems Biology: A Brief Overview*, Science, 2002
2. Uri Alon, *An Introduction to Systems Biology*, Chapman & Hall/CRC
3. Brian Ingalls, *Mathematical Modeling in Systems Biology*, MIT Press
4. *Systems Biology: A Textbook*, Wiley-VCH
5. *PLOS Computational Biology* articles on Machine Learning and Computational Biology

Reference Books:

1. Eberhard O. Voit, *A First Course in Systems Biology*, Garland Science
2. *Complex Systems Science in Biomedicine*, Springer
3. *Genomic Signal Processing*
4. *Pattern Recognition and Machine Learning*, Springer
5. *The Elements of Statistical Learning*, Springer.

****

5IMC304 – Program Elective Course- I (ii) Android Application Development

Course Objectives:

- Understand the Android ecosystem, its architecture, and the core components of an application.
- Master UI design principles using both traditional Layouts and modern Jetpack Compose.
- Learn to handle device hardware, including sensors, camera, and location-based services.
- Implement data persistence using SQLite, Session Management, and remote PHP/MySQL integration.
- Explore advanced interactivity features like animations, gestures, and multimedia.
- Prepare and publish professional-grade applications to the Google Play Store.

Course Outcomes:

After completion of this course, students will be able to:
1. Design robust Android applications utilizing Activities, Fragments, and Services.
2. Build interactive User Interfaces with event handling, custom components, and themes.
3. Integrate multimedia (audio/video) and hardware features like Bluetooth and Sensors.
4. Establish network connectivity and synchronize local data with remote databases.
5. Master the use of Android Studio, SDK Manager, and debugging tools.
6. Successfully package and deploy applications for public distribution.

Unit 1: Introduction to Android and Core Components

What is Android? Features and Environment Setup. Android Architecture and Application Structure. Organizing and Accessing Resources. Core Components: Activities and their Lifecycle, Services, Broadcast Receivers, and Content Providers. Understanding Fragments, Intents, and Intent Filters. Creating a "Hello World" example.

Unit 2: UI Design, Layouts, and Event Handling

UI Layouts (Linear, Relative, Frame, and Constraint) and UI Controls. Event Handling and Listeners. Styles and Themes. Creating Custom Components. Working with Alert Dialogs, Progress Circles, Progress Bars, and Loading Spinners. Implementing Drag & Drop functionality and Notifications.

Unit 3: Interactivity, Multimedia, and Hardware

Animations and Image Effects. Multimedia: Audio Capture, Audio Manager, Media Player, and JetPlayer. Image Switcher and Autocomplete. Hardware Integration: Bluetooth, Camera, and Sensors. Working with Gestures, Multitouch, and the Clipboard. Custom Fonts and Localization.

Unit 4: Data Storage and Networking

Internal Storage and Data Backup. Session Management. Local Databases: SQLite Database and CRUD operations. Networking: Network Connection handling and Webview. PHP/MySQL integration for remote data synchronization. Push Notifications and Facebook Integration.

Unit 5: Advanced Services and UI Patterns

Location-Based Services and Google Maps integration. Communication: Sending E-mail, Sending SMS, and handling Phone Calls. UI Patterns and modern UI Design using the Support Library. Login Screen implementation and Navigation patterns.

Unit 6: Tools and Google Play Deployment

Developer Tools and SDK Manager. Working with the Emulator. ADB (Android Debug Bridge) and Logcat. Application Security and Signing. App Store Optimization (ASO). Step-by-step process to Publish Android applications on the Google Play Store.

Text Books:

1. Android Programming: The Big Nerd Ranch Guide - Author: Bill Phillips and Chris Stewart
2. Professional Android - Author: Reto Meier and Ian Lake
3. Android Application Development (Black Book) - Author: Pradeep Kothari

Reference Books:

1. Head First Android Development - Author: Dawn Griffiths and David Griffiths
2. Android Cookbook: Problems and Solutions for Android Developers - Author: Ian F. Darwin
3. Google Android SDK Development - Author: Sayed Hashimi
4. Learning Android - Author: Marko Gargenta
5. Android Programming for Beginners - Author: John Horton.

****

5IMC304 – Programme Elective Course – (iii) 3D Printing & Design

Course Objectives

- To introduce the fundamentals of 3D printing technologies and additive manufacturing processes.
- To develop skills in 3D modeling and design using CAD tools.
- To understand materials, processes, and constraints involved in 3D printing.
- To enable students to prepare, slice, and fabricate models using 3D printers.
- To explore applications of 3D printing in engineering, healthcare, and product development.

Course Outcomes

After successful completion of the course, students will be able to:

1. Explain different 3D printing technologies, processes, and materials.
2. Design 3D models using CAD tools suitable for additive manufacturing.
3. Prepare and optimize models for printing using slicing software.
4. Operate 3D printers and fabricate physical prototypes.
5. Apply 3D printing concepts in real-world applications and evaluate design feasibility.

Syllabus (6 Units)

Unit I: Introduction to 3D Printing & Additive Manufacturing

Definition and history of 3D printing , Additive vs Subtractive manufacturing , Advantages and limitations of 3D printing , Overview of 3D printing workflow , Applications: Engineering, healthcare, architecture, education.

Unit II: 3D Printing Technologies

FDM (Fused Deposition Modeling) , SLA (Stereolithography) ,SLS (Selective Laser Sintering) , DLP and Binder Jetting (overview) , Comparison of technologies (cost, accuracy, materials, use cases).

Unit III: 3D Modeling and Design

Basics of 3D geometry (points, edges, faces, meshes), CAD modeling concepts (sketching, extrusion, revolution) , Design for Additive Manufacturing (DfAM) , File formats: STL, OBJ, AMF Introduction to tools (Tinkercad / Fusion 360 / Blender).

Unit IV: Materials and Process Parameters: Types of materials: PLA, ABS, PETG, Resin, Metal powders (overview), Material properties and selection criteria , Printing parameters: layer height, infill, speed, temperature , Support structures and orientation, Common printing defects and troubleshooting .

Unit V: Slicing, Printing & Post-Processing

Slicing software (Cura / PrusaSlicer) , G-code basics , Printer components and setup , Printing workflow (bed leveling, calibration) , Post-processing techniques: sanding, curing, polishing ,

Unit VI: Applications and Future Trends

Rapid prototyping and product development, 3D printing in healthcare (prosthetics, implants), Construction and food printing (overview) , Sustainability and challenges , Future trends: bioprinting, 4D printing, mass customization.

Text Books

1. Ian Gibson, David Rosen, Brent Stucker – *Additive Manufacturing Technologies*, Springer
2. Ben Redwood, Filemon Schöffer, Brian Garret – *The 3D Printing Handbook*, 3D Hubs

Reference Books

1. Andreas Gebhardt – *Understanding Additive Manufacturing*, Hanser Publications
2. Chee Kai Chua, Kah Fai Leong – *3D Printing and Additive Manufacturing*, World Scientific
3. Hod Lipson, Melba Kurman – *Fabricated: The New World of 3D Printing*, Wiley
4. Sandeep Kumar, et al. – *Additive Manufacturing: Principles, Technologies and Applications*
5. Richard Horne – *3D Printing for Dummies*, Wiley
6. Mastering Autodesk Fusion 360 (Latest Edition).

******

5IMC309 Programme Elective Course – I
System Biology – Lab. (Practical)

Course Objectives:

- To familiarize students with biological data sources, databases, and different types of omics data through hands-on exploration.
- To develop the ability to construct, analyze, and visualize biological networks using computational tools.
- To enable students to model and simulate simple biological systems using mathematical and programming approaches.
- To impart skills in preprocessing, normalization, and feature extraction of high-dimensional biomedical datasets.
- To introduce students to the application of computational tools and basic machine learning techniques in healthcare and systems biology problems.

Course Outcomes:

After successful completion of the course, students will be able to:

1. Explore and interpret biological data from standard databases and classify different types of omics datasets.
2. Construct and analyze biological networks and pathways using graph-based methods and visualization tools.
3. Implement and simulate basic mathematical models (e.g., ODEs, feedback systems) to study biological processes.
4. Perform preprocessing, normalization, and feature engineering on omics datasets for computational analysis.
5. Apply basic machine learning techniques to biological datasets for tasks such as disease prediction and biomarker identification.

List of 20 Practicals (Aligned to Syllabus)

(Perform any 08)

1. Exploration of Biological Databases

- Access and explore NCBI / UniProt / Ensembl

- Retrieve gene/protein information

2. Central Dogma Simulation (DNA → RNA → Protein)
- Simple Python script to simulate transcription & translation

3. Biological Data Types Identification
- Classify datasets into genomic, proteomic, metabolomic, clinical
- Basic dataset exploration using Python (CSV files)

4. Construction of Simple Biological Network
- Create gene/protein interaction network using Python (NetworkX)

5. Graph Theory Metrics on Biological Networks
- Degree, centrality, clustering coefficient calculation

6. Pathway Analysis using KEGG / Reactome
- Identify pathways for a given gene set

7. Network Visualization
- Visualize biological networks using tools like Cytoscape / Python libraries

8. Simulation of Population Growth Model (ODE based)
- Implement exponential/logistic growth using Python

9. Modeling Gene Regulation using ODEs
- Simple gene expression model

10. Feedback Loop Simulation
- Positive/negative feedback system modeling

11. Steady-State Analysis
- Analyze equilibrium points in simple biological models

12. Preprocessing of Omics Dataset
- Handling missing values, normalization

13. Visualization of High-Dimensional Data
- PCA / basic dimensionality reduction

14. Feature Extraction from Omics Data
- Convert biological data into feature vectors

15. Introduction to Multi-Omics Integration
- Combine two datasets (e.g., gene expression + clinical data)

16. Introduction to Python for Systems Biology
- NumPy, Pandas, Matplotlib basics

17. Simulation of Biological System using Python
- End-to-end modeling + plotting results

18. Model Validation and Interpretation
- Compare simulated vs expected results

19. Disease Prediction using Basic ML Model
- Use classification (Logistic Regression / Decision Tree)

20. Biomarker Discovery / Feature Importance Analysis
- Identify important features from dataset

******

5IMC309 Programme Elective Course-I Lab –
II Android Application Development Lab

Course Objectives:

1. Set - up and configure the Android development environment to build basic mobile applications.
2. Design interactive graphical user interfaces using traditional layouts and modern UI components.
3. Implement device hardware features, multimedia, and handle sensor data in applications.
4. Develop data-driven apps using local SQLite databases and remote server integration (PHP/MySQL).
5. Understand the process of testing, signing, and publishing professional Android applications.

Course Outcomes:

1. To configure Android Studio and manage Android project lifecycles and core components.
2. To design responsive UIs using various layouts, event listeners, and custom themes.
3. To integrate multimedia playback, camera access, and gesture detection into mobile apps.
4. To perform CRUD operations locally and synchronise data over the network.
5. To successfully package, sign, and prepare an Android application for Google Play Store deployment.

Tentative List of Practicals: (Perform any 10)

1. **Environment Setup & "Hello World":** Install Android Studio, configure the SDK/AVD Manager, and create a basic "Hello World" application to understand the project structure.
2. **Activity Lifecycle Observation:** Write a program with Logcat statements to demonstrate and monitor the different states of the Android Activity Lifecycle (onCreate, onStart, onResume, etc.).
3. **Intents and Navigation:** Create an application using Implicit and Explicit Intents to pass data and navigate between multiple Activities.
4. **Working with Fragments:** Develop a multi-pane application using Fragments to create a dynamic and flexible user interface.
5. **Designing User Interfaces:** Design a user registration screen utilising combinations of ConstraintLayout, LinearLayout, and RelativeLayout.
6. **Event Handling and UI Controls:** Create an application to handle button clicks, read input from EditText fields, and display output using Toasts and TextViews.
7. **Alert Dialogs and Progress Indicators:** Implement a form submission interface that triggers an Alert Dialog for confirmation and displays a Progress Spinner during the mock submission process.
8. **Custom Components and Notifications:** Build an app demonstrating Drag & Drop functionality and trigger a local system Notification upon task completion.
9. **Multimedia Playback:** Develop an application using MediaPlayer to play, pause, and stop a local audio file, and display dynamic images using an Image Switcher.
10. **Camera Integration:** Create an application that requests hardware permissions, accesses the device camera, captures a photo, and displays it on the screen.
11. **Sensor Data Reading:** Write a program to access device sensors (e.g., Accelerometer or Proximity sensor) and display real-time physical values on the UI.
12. **Gestures and Localisation:** Implement multi-touch gesture detection (swipe, pinch) and apply custom fonts along with multiple language support (localisation) via strings.xml.
13. **Session Management with Shared Preferences:** Create a login application that uses Shared Preferences to remember user credentials and manage active sessions.
14. **SQLite Database (CRUD Operations):** Develop a simple Contact Manager application that performs Create, Read, Update, and Delete operations using a local SQLite database.
15. **WebView Integration:** Create an app containing a WebView component that loads a specific URL and handles basic web navigation (back/forward).
16. **Remote Database Synchronisation:** Write an application that fetches and sends JSON data to a remote PHP/MySQL server using HTTP networking libraries (e.g., Volley or Retrofit).
17. **Location-Based Services (Google Maps):** Integrate the Google Maps API to display the user's current GPS location and place a custom interactive marker on the map.
18. **Device Communication Features:** Develop an application that uses implicit intents to trigger device communication: sending an SMS, drafting an e-mail, and initiating a phone call.
19. **Modern UI Navigation Patterns:** Implement modern app navigation using the Android Support Library, featuring a Navigation Drawer or Bottom Navigation View.
20. **Debugging with ADB and Logcat:** Use the Android Debug Bridge (ADB) and Logcat in Android Studio to debug logical errors, trace application crashes, and monitor app performance in the Emulator.
21. **Application Security and Permissions:** Implement and test application security measures, explicitly declaring and handling runtime permissions in the AndroidManifest.xml.
22. **App Signing and APK/AAB Generation:** Generate a signed release APK or Android App Bundle (AAB) using the Android Studio Build tool and Keystore generation wizard.
23. **Google Play Store Deployment Prep:** Prepare store listing assets (App Icon, feature graphics, screenshots) and write a draft App Store Optimisation (ASO) description for public deployment.

Integration Summary: These practicals integrate the Android Application Development theory syllabus with hands-on Java/Kotlin programming, providing real-world experience in mobile app creation, database integration, hardware interfacing, and software deployment.

Tools and Libraries:

- **Android Studio:** The official Integrated Development Environment (IDE) for Android app development.
- **Java / Kotlin:** Core programming languages for building Android application logic.
- **SQLite:** For local, lightweight, and structured data storage and management.

- **Google Play Services SDK:** For integrating location-based services, Maps, and push notifications.
- **Volley / Retrofit (Optional but recommended):** Modern libraries for handling asynchronous network requests and REST API integration.

****

5IMC305 MULTIMEDIA TECHNOLOGY

Course Objectives

1. Understand the fundamental concepts, elements, and applications of Multimedia.
2. Learn the hardware and software requirements for multimedia systems.
3. Explore the characteristics and formats of different media elements like text, images, audio, and video.
4. Understand the principles of compression and animation techniques.
5. Develop skills to design simple multimedia applications using authoring tools.

Course Outcomes: After completion of this course Students will be able to-

1. To demonstrate an understanding of multimedia components and system architecture.
2. To differentiate between various image, audio, and video file formats.
3. To apply basic concepts of compression and animation in multimedia projects.
4. To utilize multimedia tools and authoring software for content creation.
5. To design and develop interactive multimedia applications for various domains.

Unit 1: Introduction to Multimedia

Definition of Multimedia, Elements of Multimedia (Text, Audio, Image, Animation, Video), Characteristics of Multimedia Systems. Multimedia Applications: Education, Entertainment, Business, and Medical. Hardware Requirements: CPU, Memory, Storage Devices, Input/Output Devices. Software Requirements: Operating Systems, Multimedia Software Tools. Introduction to Hypermedia and Hypertext.

Unit 2: Text and Graphics Fundamentals

Text in Multimedia: Fonts, Typefaces, and Text Editing Tools. Introduction to Computer Graphics: Vector vs. Bitmap Graphics. Image File Formats: GIF, JPEG, PNG, BMP, and TIFF. Color Models: RGB, CMYK, HSB. Basic Image Editing Concepts: Cropping, Resizing, and Color Correction. Tools for creating and editing graphics (Overview of tools like Photoshop/GIMP basics).

Unit 3: Audio and Video Technology

Sound in Multimedia: Digital Audio vs. MIDI. Sampling Rate, Bit Depth, and Sound Quality. Audio File Formats: WAV, MP3, AAC, OGG. Basic Audio Editing tools and techniques. Video in Multimedia: Analog vs. Digital Video. Video Formats: AVI, MP4, MOV, WMV. Video Production Basics: Shooting, Capturing, and Editing. Overview of Video Editing Software (Adobe Premiere/Open source tools).

Unit 4: Animation and Compression

Introduction to Animation: Types (Cel, Path, Procedural). Principles of Animation. Computer Animation Techniques: Keyframing, Morphing. Overview of Animation Tools (Flash/Animate). Need for Data Compression. Lossy vs. Lossless Compression. Basic Compression Techniques: Run-Length Encoding (RLE), Huffman Coding (Conceptual understanding only). Overview of JPEG and MPEG compression standards.

Unit 5: Multimedia Authoring and Tools

Introduction to Multimedia Authoring: Metaphors (Scripting, Iconic, Timeline). The Authoring Process. Overview of Authoring Tools: Features of Macromedia Director, Flash, and HTML5 based tools. Designing User Interface: Navigation structures, Menus, Buttons. Integrating different media elements into a single project.

Unit 6: Multimedia Design and Project Development

Multimedia Project Life Cycle: Planning, Designing, Development, and Testing. Developing a Multimedia Storyboard. Designing for the Web vs. CD-ROM/DVD. Intellectual Property and Copyright issues in Multimedia. Future Trends in Multimedia: Virtual Reality (VR), Augmented Reality (AR) basics. Case Study: Designing a simple interactive presentation or website.

Text books:

1. Principles of Multimedia Author: Ranjan Parekh (McGraw Hill Education)
2. Multimedia: Making It Work Author: Tay Vaughan (Osborne/McGraw Hill)
3. Multimedia Systems Author: Parag Havaldar and Gerard Medioni (Pearson)

References Books:

1. Multimedia Technology and Applications Author: David Hillman (Galgotia Publications)

2. Multimedia: Computing, Communications & Applications Author: Ralf Steinmetz and Klara Nahrstedt (Pearson)
3. Introduction to Multimedia Systems Author: S. K. Khandai (Prentice Hall India)
4. Digital Multimedia Author: Nigel Chapman and Jenny Chapman (Wiley)
5. Computer Graphics and Multimedia Author: A.P. Godse and D.A. Godse (Technical Publications)

****

5IMC306 Data Communication & Networking- Lab

Course Objectives:

1. To provide basic practical exposure to networking concepts studied in theory.
2. To develop understanding of IP addressing and subnetting through guided exercises.
3. To familiarize students with commonly used network configuration and diagnostic tools.
4. To demonstrate working of Internet protocols using packet analysis tools.
5. To understand basic LAN setup and device connectivity.
6. To introduce students to wireless network configuration at beginner level.

Course Outcomes:

1. Identify networking components and describe their functions.
2. Configure basic IPv4 addressing and verify network connectivity.
3. Perform subnetting calculations with practical validation.
4. Use networking utilities such as ping, tracert and ipconfig/ifconfig.
5. Observe and analyze protocol behavior using packet analyzer tools.
6. Configure and test a simple wired and wireless network setup.

List of Practical Experiments:

1. Study of Networking Components and Network Topologies.
2. Execute and analyze the output of basic networking commands such as ipconfig/ifconfig, ping, tracert/traceroute for understanding system network configuration and connectivity.
3. To manually configure IPv4 addresses on two systems within the same network and verify connectivity using the ping command
4. To perform IPv4 subnetting for Class C networks and validate the calculated subnet configuration through practical IP assignment or simulation.
5. To observe and analyze the MAC address of a system and examine the ARP table to understand IP-to-MAC address mapping in a LAN.
6. To capture and analyze network packets using Wireshark and identify HTTP, DNS, and ICMP protocol messages.
7. To create a simple Local Area Network using Cisco Packet Tracer by connecting multiple PCs through a switch, assigning IP addresses, and testing connectivity.
8. To configure a basic wireless network using a wireless router in simulation software and test client device connectivity.

Tool-based References:

1. Cisco Packet Tracer
2. Wireshark

****

5IMC307 CLIENT SERVER COMPUTING- LAB

Course Objectives

1. To understand web design principles and client–server architecture.
2. To develop responsive web pages using HTML, CSS, JavaScript, and Bootstrap.
3. To implement database connectivity using JDBC with MySQL/PostgreSQL.
4. To build dynamic web applications using Servlets and JSP.
5. To apply MVC architecture and session management techniques.
6. To develop RESTful CRUD applications using Spring Boot with database integration.

Course Outcomes: After completing the course, students will be able to-

1. Explain web architecture, HTTP protocol, and client–server models.
2. Design responsive web pages using HTML, CSS, JavaScript, and Bootstrap.
3. Perform CRUD operations using JDBC.
4. Develop server-side applications using Servlets and JSP.
5. Implement MVC architecture and manage user sessions.
6. Build RESTful web applications using Spring Boot and JPA/Hibernate.

List of 15 Practicals (Aligned to Syllabus) :

1. Create a basic static website using HTML5 demonstrating headings, lists, tables, forms, and multimedia elements.
2. Design a responsive web page using CSS3 and Bootstrap with proper layout and styling.
3. Implement JavaScript for client-side validation and DOM manipulation.
4. Write a JDBC program to connect Java with MySQL/PostgreSQL and create a database and table.
5. Develop a JDBC program to insert, update, delete, and retrieve records using Statement and Prepared Statement.
6. Write a JDBC program to call a stored procedure using Callable Statement and display the results.
7. Create a Servlet application integrated with JDBC to insert and retrieve data from MySQL.
8. Develop a Servlet-based application demonstrating session management using Cookies and Http Session.
9. Develop a JSP page demonstrating directives, scripting elements, implicit objects, and Expression Language (EL).
10. Create a JSP application integrated with JDBC to display student subject-wise data from a database.
11. Create a Spring Boot project to develop a RESTful CRUD application connected to a database using JPA/Hibernate.
12. Develop a complete web application using Servlet + JDBC + JSP (MVC pattern) to perform CRUD operations with MySQL database.
13. Create and Run a Spring Boot Application
14. REST API with Database Connection
15. Develop a mini project integrating HTML, CSS, JavaScript, JDBC, Servlets, JSP, and Spring Boot covering all major concepts of the course.

****

5IMC308 MULTIMEDIA TECHNOLOGY – Lab.

Course Outcomes: After completion of this course Students will be able to-

1. To utilize multimedia hardware and software tools effectively.
2. To create and edit graphics and images using editing software.
3. To record, edit, and mix audio files for multimedia applications.
4. To develop animations and video sequences with transitions.
5. To design and integrate multimedia elements into a final project.

Tentative List of Practical Exercises:

1. Study of Multimedia Hardware: Identify and report on multimedia hardware components (RAM, Storage, I/O devices) and categorize installed software tools in the laboratory.
2. Typography and Text Design: Create a design layout using various fonts, typefaces, and text effects (Drop Shadow, Gradient) to understand the role of text in multimedia.
3. Image Editing and Formats: Create a raster and vector graphic. Export images in different formats (JPEG, PNG, GIF) and compare file sizes and quality to understand compression.
4. Audio Recording and Editing: Record a voice clip, remove background noise, and mix it with a background music track using audio editing software (e.g., Audacity).
5. Video Editing: Create a short video sequence by importing clips/images, adding transitions (Dissolve/Wipe), and rendering the final output as an MP4 file.
6. Frame-by-Frame Animation: Create a simple animation (e.g., bouncing ball) using frame-by-frame technique to understand the concept of frames.
7. Motion Tweening: Develop an animation demonstrating motion tweening to show object movement and transformation automatically.
8. Web Integration: Create a basic HTML webpage embedding an image, an audio file, and a video player to demonstrate multimedia integration.
9. Storyboarding: Design a hand-drawn or digital storyboard for a 30-second advertisement, outlining scenes, scripts, and audio cues.
10. Mini Project: Develop a comprehensive multimedia application (e.g., College Presentation or Personal Portfolio) combining text, graphics, audio, and video.

****

SEMESTER VI

6IMC301 ADVANCED DATABASE MANAGEMENT SYSTEMS

Course Objectives (COs) After completion of this course, students will be able to:

- Understand advanced database concepts such as distributed and parallel databases.
- Apply transaction processing and concurrency control mechanisms.
- Analyze database recovery and fault-tolerance techniques.
- Evaluate query optimization and performance tuning strategies.
- Design distributed database schemas and architectures
- Explore emerging database technologies like NoSQL and Big Data systems.

Course Outcomes : After successful completion of the course, students will be able to:

- Explain advanced DBMS architectures and models.
- Implement transaction and concurrency control techniques.
- Analyze recovery mechanisms and distributed database issues.
- Evaluate query processing and optimization methods.
- Design distributed and parallel database solutions
- Compare relational and NoSQL database systems.

Unit 1: Advanced Database Architecture (7 Hrs)
Review of DBMS architecture, Client–Server and 3-Tier architecture, Parallel databases, Distributed databases: concepts and advantages, Homogeneous and heterogeneous DDBMS.

Unit 2: Distributed Database Design (7 Hrs)
Data fragmentation (horizontal, vertical, hybrid), Data replication, Allocation strategies, Transparency in distributed DBMS, Distributed catalog management.

Unit 3: Transaction Processing (7 Hrs)
Transaction concepts, ACID properties, Distributed transactions, Two-Phase Commit Protocol (2PC)

Unit 4: Concurrency Control (7 Hrs)
Lock-based protocols, Timestamp-based protocols, Deadlock handling (prevention, detection, recovery), Multiversion concurrency control.

Unit 5: Recovery and Query Optimization (7 Hrs)
Failure types, Log-based recovery, Checkpointing, Query processing in distributed DB, Query optimization techniques.

Unit 6: Emerging Database Technologies (7 Hrs)
Introduction to NoSQL databases (Key-Value, Document, Column, Graph) and Cloud databases, Introduction to MongoDB and HBase.

Text Books:-

1. Database System Concepts – Korth & Sudarshan
2. NoSQL Distilled – Pramod Sadalage.

Reference Books:

1. Distributed Database Systems – Ozsu & Valduriez
2. Advanced Database Systems – Elmasri & Navathe.

****

6IMC302 Introduction to Machine Learning

Course Objectives (Max 6)

1. To introduce the fundamental concepts and paradigms of machine learning.
2. To understand the process of data preparation, feature engineering, and model evaluation.
3. To learn and implement supervised learning algorithms for classification and regression tasks.
4. To explore unsupervised learning techniques for pattern discovery and clustering.
5. To apply basic ensemble methods to improve model performance and robustness.
6. To provide hands-on experience with Python libraries for machine learning.

Course Outcomes (Max 5)

After completion of this course Students will be able to:

1. To perform data preprocessing and exploratory data analysis on real-world datasets.
2. To implement and evaluate supervised learning models like regression, SVM, and decision trees.
3. To apply unsupervised learning techniques such as clustering for data segmentation.
4. To use ensemble methods to enhance the accuracy and stability of machine learning models.
5. To develop and assess a basic machine learning solution for a given problem using Python.

Unit 1: Introduction to Machine Learning (6 hours)

What is Machine Learning? Applications of ML, Types of Machine Learning: Supervised, Unsupervised, Semi-supervised, and Reinforcement Learning. Key Concepts: Features, Labels, Training, Validation, and Testing. The Machine Learning Workflow. Introduction to the Python ML Ecosystem: NumPy, Pandas, Matplotlib, and Scikit-learn.

Unit 2: Data Pre-processing and Exploration (7 hours)

Data Acquisition and Loading. Handling Missing Data. Data Cleaning and Transformation. Feature Scaling: Normalization and Standardization. Exploratory Data Analysis (EDA): Summary Statistics, Data Visualization (Histograms, Box Plots, Scatter Matrices) for understanding data distributions and relationships.

Unit 3: Supervised Learning - Regression (7 hours)

Introduction to Regression. Simple Linear Regression: Concept, Assumptions, and Implementation. Multiple Linear Regression. Model Evaluation Metrics for Regression: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.

Unit 4: Supervised Learning - Classification (7 hours)

Introduction to Classification. Logistic Regression for Binary Classification. k-Nearest Neighbors (k-NN) Algorithm. Decision Trees: Concepts of Entropy and Information Gain. Support Vector Machines (SVM)

Unit 5: Unsupervised Learning and Ensemble Methods (7 hours)

Unsupervised Learning: Introduction to Clustering. k-Means Clustering: Algorithm, Choosing the right number of clusters (Elbow Method). Introduction to Dimensionality Reduction using Principal Component Analysis (PCA).

Unit 6: Model Evaluation and Deployment Concepts (6 hours)

Bias-Variance Tradeoff. Overfitting and Underfitting. Cross-Validation Techniques (k-fold). Introduction to concepts in ML Model Deployment: APIs and web services.

Textbooks:

1. Introduction to Machine Learning with Python: A Guide for Data Scientists Author: Andreas C. Müller & Sarah Guido
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow Author: Aurélien Géron
3. The Hundred-Page Machine Learning Book Author: Andriy Burkov.

References Books:

1. Pattern Recognition and Machine Learning Author: Christopher M. Bishop.
2. Machine Learning for Absolute Beginners: A Plain English Introduction Author: Oliver Theobald.
3. Python Machine Learning Author: Sebastian Raschka & Vahid Mirjalili
4. An Introduction to Statistical Learning: with Applications in R (Python compatible concepts) Author: Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
5. Data Science from Scratch: First Principles with Python Author: Joel Grus.

****

6IMC303 PROGRAM ELECTIVE COURSE-II (NATURAL LANGUAGE PROCESSING)

Course Objectives

1. To introduce fundamental concepts of Natural Language Processing and linguistic structures.
2. To develop understanding of text preprocessing and feature representation techniques.
3. To explain machine learning approaches used in NLP tasks.
4. To explore major real-world applications of NLP.
5. To introduce neural network-based approaches and word embeddings.
6. To familiarize students with transformer architecture and ethical aspects of modern NLP systems.

Course Outcomes: After completion of this course Students will be able to

1. Understand the basic concepts, challenges, and linguistic foundations of Natural Language Processing.
2. Apply text preprocessing and feature extraction techniques for NLP tasks.
3. Explain machine learning models and evaluation metrics used in text classification.
4. Analyze major NLP applications such as sentiment analysis, named entity recognition, and summarization.
5. Describe distributed word representations and recurrent neural network concepts.
6. Demonstrate conceptual understanding of transformer architecture and responsible AI practices in NLP.

Unit I: Introduction to Natural Language Processing (7 Hours)

Introduction to Natural Language Processing; Applications of NLP; NLP Pipeline; Ambiguity in Natural Language; Levels of Language Analysis; Morphology; Syntax; Semantics; Pragmatics; Challenges in NLP.

Unit II: Text Preprocessing and Feature Engineering in NLP (7 Hours)

Text Cleaning; Tokenization; Stop Word Removal; Stemming; Lemmatization; Regular Expressions; N-grams; Bag of Words Model; Term Frequency; Inverse Document Frequency.

Unit III: Text Classification and Machine Learning for NLP (7 Hours)

Introduction to Text Classification; Workflow of NLP Model Building; Introduction to Naive Bayes; Introduction to Logistic Regression; Introduction to Support Vector Machine; Confusion Matrix; Accuracy; Precision; Recall; F1-Score; Concept of Overfitting and Underfitting.

Unit IV: Core NLP Applications (7 Hours)

Sentiment Analysis; Spam Detection; Named Entity Recognition; Information Extraction; Text Summarization; Machine Translation; Question Answering Systems; Chatbot Fundamentals.

Unit V: Word Embeddings and Deep Learning Foundations (7 Hours)

Limitations of Bag of Words; Distributed Word Representation; Introduction to Word2Vec; Introduction to GloVe; Semantic Similarity; Introduction to Neural Networks; Recurrent Neural Networks; Limitations of Recurrent Models.

Unit VI: NLP in AI and Emerging Trends in NLP (7 Hours)

Hallucination in Language Models; Responsible AI in NLP; Future Trends in NLP.

Text Books:

1. Speech and Language Processing, Daniel Jurafsky and James Martin, 3rd Edition (Draft), 2024, Pearson.
2. Natural Language Processing with Python, Steven Bird, Ewan Klein, and Edward Loper, 1st Edition, 2009, O'Reilly Media.

Reference Books:

1. Neural Network Methods for Natural Language Processing, Yoav Goldberg, 1st Edition, 2017, Morgan & Claypool Publishers.
2. Transformers for Natural Language Processing, Denis Rothman and Antonio Gulli, 2nd Edition, 2022, Packt Publishing.
3. Python Machine Learning, Sebastian Raschka and Vahid Mirjalili, 3rd Edition, 2019, Packt Publishing.
4. Hugging Face Transformers Documentation.

****

6IMC303 Program Elective Course-II (Big Data Analytics)

Course Objectives:

1. To introduce fundamental concepts and architecture of Big Data systems.
2. To explain distributed storage and processing frameworks used in Big Data.
3. To develop understanding of modern data processing technologies such as Hadoop and Spark.
4. To explore NoSQL databases and large-scale storage systems.
5. To introduce concepts of data engineering and Big Data pipelines.
6. To familiarize students with cloud-based Big Data platforms and modern trends.

Course Outcomes:

1. Understand characteristics, architecture, and challenges of Big Data systems.
2. Explain distributed storage concepts in the Hadoop ecosystem.
3. Describe Spark architecture and large-scale data processing approaches.
4. Analyze different types of NoSQL databases and storage models.
5. Explain ETL, data pipelines, and real-time processing concepts.
6. Demonstrate conceptual understanding of cloud-based Big Data services and emerging trends.

Unit I: Foundations of Big Data (7 Hours)

Definition of Big Data; Characteristics of Big Data; 5Vs of Big Data; Big Data Architecture Overview; Distributed Computing Basics; CAP Theorem; Challenges in Big Data Systems.

Unit II: Hadoop Ecosystem (8 Hours)

Introduction to Hadoop; Hadoop Architecture Overview; HDFS Architecture; NameNode and DataNode; Block Storage Concept; Data Replication; Introduction to YARN; MapReduce Programming Model; Limitations of MapReduce.

Unit III: Apache Spark and Modern Distributed Processing (8 Hours)

Need for Apache Spark; Spark Architecture Overview; Resilient Distributed Datasets; DataFrames Concept; In-Memory Processing; Spark and Hadoop Comparison; Batch Processing; Introduction to Stream Processing.

Unit IV: NoSQL Databases and Modern Data Storage Systems (7 Hours)

Limitations of Relational Databases; Introduction to NoSQL; Types of NoSQL Databases; Key-Value Stores; Document Databases; Column-Family Databases; Graph Databases; Overview of MongoDB; Data Lakes; Data Warehouses; Comparison of Data Lakes and Data Warehouses.

Unit V: Data Engineering and Data Pipeline Architectures (7 Hours)

Introduction to Data Engineering; ETL and ELT Concepts; Data Ingestion; Data Pipelines; Batch Processing Systems; Real-Time Processing Systems; Introduction to Apache Kafka; Data Quality Concepts; Data Governance Basics.

Unit VI: Cloud-Based Big Data Analytics and Future Trends (8 Hours)

Big Data in Cloud Environment; Cloud-Based Big Data Architecture; Managed Big Data Services Concept; Serverless Data Processing; Big Data Security and Privacy Concepts; Integration of Big Data with Artificial Intelligence; Future Trends in Big Data Analytics.

Textbooks:

1. "Big Data and Analytics" by Seema Acharya and Subhashini Chellappan, 2nd Edition, Wiley India.
2. "Spark: The Definitive Guide" by Bill Chambers and Matei Zaharia, 1st Edition, O'Reilly Media.

Reference Books:

1. "Hadoop: The Definitive Guide" by Tom White, 4th Edition, O'Reilly Media.
2."Designing Data-Intensive Applications" by Martin Kleppmann, 1st Edition, O'Reilly Media.
3. "Big Data: Principles and Best Practices of Scalable Real-Time Data Systems" by Nathan Marz and James Warren, 1st Edition, Manning Publications.

****

6IMC304 PROGRAMME ELECTIVE COURSE –III - DATA WAREHOUSE DATA MINING

Course Objectives : After completion of this course, students will be able to:

1. Understand the concepts of data warehousing and OLAP systems.
2. Explain data preprocessing and data integration techniques.
3. Apply data mining algorithms for classification and clustering.
4. Analyze association rules and pattern discovery methods.
5. Evaluate different data mining models and performance measures.
6. Design a simple data warehouse and mining solution for real-world data..

Course Outcomes: After successful completion of the course, students will be able to:

1. Describe architecture and components of data warehouse systems.
2. Perform data preprocessing and transformation on datasets.
3. Implement classification, clustering, and association mining techniques.
4. Analyze patterns and trends using OLAP and mining results.
5. Evaluate effectiveness of mining models using metrics.
6. Develop a basic data warehouse and mining application.

Unit 1: Data Warehousing Fundamentals

Data warehouse concept and characteristics, OLTP vs OLAP, Data warehouse architecture, Data marts, Metadata and ETL process, Applications of data warehouse.

Unit 2: Data Warehouse Design & Modelling

Dimensional modelling, Fact and dimension tables, Star schema, Snowflake schema, Schema design process, Data cube concept.

Unit 3: Data Preprocessing

Data cleaning, Data integration, Data transformation, Data reduction, Handling missing and noisy data, Data discretization.

Unit 4: Association Rule Mining
Market basket analysis, Frequent itemsets, Apriori algorithm, Association rules, Support and confidence, Rule generation and pruning.

Unit 5: Classification & Clustering
Classification concepts, Decision tree classification, Naïve Bayes classification, Clustering concepts, K-means clustering, Cluster evaluation.

Unit 6: OLAP & Advanced Topics
OLAP operations (roll-up, drill-down, slice, dice), OLAP cube, Data mining applications, Web mining, Text mining, Big data & data mining tools overview.

Recommended Books:

1. Data Mining: Concepts and Techniques – Jiawei Han & Micheline Kamber
2. Data Warehousing in the Real World – Sam Anahory
3. Introduction to Data Mining – Pang-Ning Tan
4. Data Warehouse Toolkit – Ralph Kimball.

****

6IMC304 PROGRAM ELECTIVE COURSE-III IMAGE PROCESSING

Course Objectives (Max 6)

1. Understand the fundamental concepts of digital image processing and its applications.
2. Learn image enhancement techniques to improve image quality for human interpretation.
3. Explore image restoration and morphological operations.
4. Understand the concepts of image segmentation and feature extraction.

Apply image compression techniques for efficient storage and transmission.

Course Outcomes (Max 6)
After completion of this course Students will be able to:

1. To demonstrate an understanding of digital image fundamentals and pixel relationships.
2. To apply spatial domain techniques for image enhancement and filtering.
3. To implement frequency domain transforms for image processing tasks.
4. To utilize morphological operations for image analysis.
5. To apply segmentation and compression methods for practical image processing solutions.

Unit 1: Introduction and Digital Image Fundamentals-I: (6 hours)

Introduction to Digital Image Processing: Examples and Applications. Elements of Visual Perception: Structure of the Human Eye, Image Formation. Light and the Electromagnetic Spectrum.

Unit 1: Introduction and Digital Image Fundamentals-II: (6 hours)

Image Sensing and Acquisition. Image Sampling and Quantization. Basic Relationships between Pixels: Neighbors, Adjacency, Connectivity, and Distance Measures.

Unit 2: Image Enhancement in Spatial Domain (7 hours)
Introduction to Image Enhancement. Basic Gray Level Transformations. Histogram Processing: Histogram Equalization and Specification. Basics of Spatial Filtering.

Unit 3: Image Enhancement in Frequency Domain (7 hours)
Introduction to the Frequency Domain. Basic Concepts of the Fourier Transform (Conceptual understanding, avoiding heavy math). Introduction to filtering in the Frequency Domain.

Unit 4: Image Restoration and Morphological Processing (7 hours)
Image Restoration: Degradation models (Noise models: Gaussian, Salt & Pepper). Mean Filters and Order-Statistic Filters for Restoration.

Unit 5: Image Segmentation (7 hours)
Introduction to Segmentation. Detection of Discontinuities: Point, Line, and Edge Detection. Edge Detection Operators: Roberts, Prewitt, Sobel, and Canny Edge Detector.

Unit 6: Image Compression and Applications (6 hours)
Fundamentals of Image Compression: Coding, Interpixel, and Psychovisual Redundancy. Applications of Image Processing: Medical Imaging (X-ray, MRI), Remote Sensing, and Fingerprint Recognition.

Text books: (Max 3)(Compulsory)

1. Digital Image Processing Author: S. Jayaraman, S. Esakkirajan, and T. Veerakumar (Tata McGraw Hill)
2. Digital Image Processing Author: Rafael C. Gonzalez and Richard E. Woods (Pearson)
3. Digital Image Processing using MATLAB Author: C. Gonzalez, R. Woods, and S. Eddins (Pearson)

References Books: (Max 5) (Not Compulsory)

1. Digital Image Processing Author: Anil K. Jain (Prentice Hall)
2. Image Processing, Analysis, and Machine Vision Author: Milan Sonka, Vaclav Hlavac, and Roger Boyle (Cengage Learning)
3. Fundamentals of Digital Image Processing Author: Anil K. Jain (Pearson)
4. Digital Image Processing Author: S. Sridhar (Oxford University Press)
5. Introduction to Digital Image Processing with MATLAB Author: Alasdair McAndrew (Cengage Learning)

****

6IMC305 COMPUTER GRAPHICS

Course Objectives:

1. Introduce the basic concepts and components of computer graphics systems.
2. Understand and implement line, circle, and polygon drawing algorithms.
3. Study 2D transformations and viewing techniques used in graphics applications.
4. Learn clipping methods and basic 3D graphics transformations.
5. Understand projection techniques, hidden surface removal, and illumination models.
6. Explore advanced topics such as OpenGL, animation, color models, ray tracing, and virtual reality.

Course Outcomes:

After completion of this course Students will be able to

1. Describe the working of graphics systems, display devices, and input devices.
2. Apply different line and circle drawing algorithms to generate graphical primitives.
3. Apply 2D transformations and viewing techniques to graphical objects.
4. Perform clipping operations and apply 3D transformations in graphics applications.
5. Create basic animation and graphics programs using OpenGL.
6. Explain modern graphics concepts such as ray tracing, VR, and AR.

Unit 1: Fundamentals of Computer Graphics (6 hours)
Introduction to Computer Graphics, Components of Graphics System, Video Display Devices: CRT, LCD, LED, Input Devices: Keyboard, Mouse, Light Pen, Scanner, Raster Scan and Random Scan, Graphics Software and Hardware

Unit 2: Output Primitives and Drawing Algorithms (7 hours)
Points and Lines, DDA Line Algorithm, Bresenham Line Algorithm, Circle Drawing Algorithm, Polygon Drawing, Boundary Fill and Flood Fill

Unit 3: 2D Transformations and Viewing Techniques (7 hours)
Basic Transformations-Translation, Rotation, Scaling, Reflection, Shearing, Homogeneous Coordinates, Composite Transformations, Windowing and Viewport, 2D Viewing Pipeline

Unit 4: Clipping Techniques and 3D Graphics (7 hours)
Clipping Operations-Cohen-Sutherland Line Clipping, Liang-Barsky Algorithm, Sutherland-Hodgman Polygon Clipping, Introduction to 3D Graphics, 3D Transformations-Translation, Rotation, Scaling, 3D Viewing Pipeline

Unit 5: Projection, Hidden Surface Removal and Illumination (7 hours)
Parallel Projection, Perspective Projection, Vanishing Points, Hidden Surface Removal- Z-buffer Algorithm, Back-Face Detection, Painter's Algorithm, Illumination Models-Ambient, Diffuse, Specular Reflection, Phong Illumination Model

Unit 6: Advanced Graphics and Applications (6 hours)
Introduction to OpenGL, Basics of Animation, Multimedia Systems, Color Models (RGB, CMY, HSV), Introduction to Ray Tracing, Virtual Reality and Augmented Reality.

Text Books:

1. Donald Hearn & M. Pauline Baker – *Computer Graphics (C Version)*
2. Fundamentals of Computer Graphics – Peter Shirley & Steve Marschner (CRC Press)

3. **Computer Graphics: Principles and Practice** – James D. Foley, Andries van Dam, Steven Feiner & John Hughes (often called the classic CG book)

Reference Books:

1. **Computer Graphics (Second Edition)** – Zhigang Xiang & Roy A. Plastock, *Schaum's Outline* (Tata McGraw-Hill)
2. **Procedural Elements for Computer Graphics** – David F. Rogers (Tata McGraw-Hill).
3. **Principles of Interactive Computer Graphics** – William M. Newman & Robert F. Sproull (TMH).
4. **Principles of Computer Graphics** – Shalini Govil-Pai (Springer).
5. **Computer Graphics: A Programming Approach** – Steven Harrington (TMH)

****

6IMC309 Programme Elective Course - III_ Data Warehouse Data Mining – Lab.

List of Experiments:

1. Create a small data warehouse schema (star schema) using SQL.
2. Design fact and dimension tables for sales dataset.
3. Perform ETL process (extract CSV → load into DB).
4. Data cleaning: handle missing values in dataset.
5. Data transformation and normalization using Python/Excel.
6. Generate frequent itemsets using Apriori (WEKA/Python).
7. Association rule mining with support & confidence.
8. Classification using Decision Tree.
9. Classification using Naïve Bayes.
10. Clustering using K-means algorithm.

****

6IMC309 Program Elective Course–III –Lab Image Processing Lab.

Course Outcomes: After completion of this course students will be able to:

1. To use image processing tools to read, write, and display digital images.
2. To apply basic enhancement techniques to improve image quality.
3. To remove noise from images using filtering techniques.
4. To perform morphological operations and edge detection.
5. To implement basic segmentation and compression methods.
6. List of Practical Exercises:

1 **Image Basics**: Load an image, display it, and convert a color image (RGB) into a grayscale image.
2 **Image Arithmetic**: Add two images to create a watermark effect and subtract two images to find the differences between them.
3 **Brightness and Contrast**: Adjust the brightness (adding a constant) and contrast (multiplying by a factor) of an image manually.
4 **Histogram Equalization**: Plot the histogram of a dark image and use Histogram Equalization to automatically improve its contrast.
5 **Noise Removal (Smoothing)**: Add "Salt and Pepper" noise to an image and remove it using a Median Filter.
6 **Image Sharpening**: Apply a sharpening filter (like the Laplacian mask) to a blurred image to make it look clearer.
7 **Edge Detection**: Detect the boundaries/edges of objects in an image using the Sobel or Canny Edge Detector.
8 **Morphological Operations**: Perform Erosion and Dilation on a binary (black and white) image to remove small noise or fill holes.
9 **Image Thresholding**: Convert a grayscale image into a binary image (Black & White) using a threshold value to separate the object from the background.

Image Compression: Save an image in JPEG format with different quality settings (High, Medium, Low) and compare the file sizes to understand compression.

****

6IMC306 Advance Database Management System Lab.

Software: MySQL / PostgreSQL / MongoDB / Oracle / Distributed DB simulator

List of Experiments (Minimum 10)

1. Study and installation of distributed database environment (PostgreSQL/MySQL cluster).
2. Fragmentation of database tables (horizontal & vertical).
3. Data replication and allocation in distributed database.
4. Implementation of transaction with COMMIT and ROLLBACK.
5. Demonstration of ACID properties using SQL.
6. Simulation of concurrency control using locking.
7. Deadlock creation and detection using transactions.
8. Implementation of Two-Phase Commit (2PC) simulation.
9. Query optimization-using indexing and explain its plan.
10. Log-based recovery and checkpoint demonstration.

******

6IMC307 Introduction to Machine Learning- Lab

Course Objectives

1. Implement machine learning concepts using Python programming.
2. Perform data preprocessing and visualization on real datasets.
3. Apply supervised and unsupervised learning algorithms.
4. Evaluate performance of machine learning models.
5. Develop simple real-world predictive applications.

Course Outcomes
After completion of this course students will be able to:

1. Use Python libraries (NumPy, Pandas, Matplotlib, Scikit-Learn) for data analysis.
2. Prepare datasets using preprocessing and feature engineering techniques.
3. Implement regression and classification models.
4. Apply clustering techniques to discover hidden patterns.
5. Evaluate machine learning models and interpret results.

List of Practical's (Implement the below experiments using Python):

1. **Introduction to Python Scientific Libraries**
   Write a Python program to create NumPy arrays and perform basic operations such as addition, multiplication, and transpose.
2. **Dataset Loading and Inspection**
   Load a CSV dataset using Pandas and display dataset structure, datatype information, and statistical summary.
3. **Handling Missing Data**
   Perform data cleaning by detecting and handling missing values using deletion and mean/median imputation techniques.
4. **Data Visualization**
   Plot histogram, bar chart, scatter plot, and correlation heatmap using Matplotlib/Seaborn.
5. **Linear Regression**
   Implement Linear Regression using Scikit-Learn and predict continuous output values for a dataset.
6. **Polynomial Regression**
   Implement Polynomial Regression and compare performance with Linear Regression.
7. **K-Nearest Neighbors (KNN) Classification**
   Train and test a KNN classifier for a given dataset and analyze the effect of different values of K.
8. **Decision Tree / Naïve Bayes Classification**
   Implement a Decision Tree classifier and Naïve Bayes classifier and compare their accuracy.
9. **K-Means Clustering**
   Apply K-Means clustering algorithm and determine the optimal number of clusters using the Elbow method.

******

6IMC308 Programme Elective Course-II-Lab (Natural Language Processing Lab)

Course Objectives:

1. To provide practical exposure to basic NLP techniques using Python.
2. To implement text preprocessing and feature extraction methods.
3. To build simple text classification models.
4. To explore major NLP applications using standard libraries.
5. To demonstrate modern NLP concepts using pre-trained tools.

Course Outcomes:

1. Perform text preprocessing using NLP libraries.
2. Generate numerical representations of text data.
3. Build and evaluate basic text classification models.
4. Implement fundamental NLP applications such as sentiment analysis and named entity recognition.
5. Use pre-trained language models for simple NLP tasks.

List of Practical Experiments:

1. Text Preprocessing Using Python and NLTK Including Tokenization, Stop Word Removal, and Lemmatization.
2. Implementation of N-grams and Bag of Words Model Using scikit-learn.
3. Implementation of TF-IDF Representation for Text Dataset.
4. Building a Simple Text Classification Model Using Naive Bayes.
5. Building a Text Classification Model Using Logistic Regression and Evaluating Using Accuracy and F1-Score.
6. Performing Sentiment Analysis on Movie Review Dataset Using Preprocessed Features.
7. Named Entity Recognition using spaCy Library.
8. Implementation of Text Similarity and Cosine Similarity Using TF-IDF.

Tools and Programming Languages:

- Programming Language: Python
- Development Environment: Google Colab / Jupyter Notebook
- Libraries: NLTK, spaCy, scikit-learn, gensim
- Supporting Libraries: NumPy, Pandas, Matplotlib.

****

6IMC308 Programme Elective Course-II-Lab (Big Data Analytics Lab)

Course Objectives:

1. To provide basic hands-on exposure to Big Data processing tools.
2. To demonstrate distributed data concepts using simplified environments.
3. To work with large datasets using Spark and NoSQL tools.
4. To understand batch and streaming concepts through guided exercises.
5. To explore cloud-based Big Data environments through demonstrations.

Course Outcomes:

1. Perform basic data processing using Spark in local mode.
2. Execute simple data transformation and aggregation operations.
3. Store and retrieve data using a NoSQL database.
4. Implement basic batch data workflows.
5. Understand streaming and cloud-based Big Data concepts through practical exposure.

List of Practical Experiments:

1. Study of Big Data Ecosystem and Architecture Using Case Study Analysis.

2. Working with Hadoop Distributed File System Commands (Pre-configured Environment).

3. Installation and Setup of Apache Spark in Local Mode or Google Colab Environment.

4. Performing Basic Data Transformations Using Spark RDD.

5. Performing Data Analysis and Aggregation Using Spark DataFrames.

6. Loading and Querying Data Using MongoDB Atlas (Cloud-Based NoSQL).

7. Comparative Analysis of Structured Data in MySQL and Unstructured Data in MongoDB.

8. Implementation of Simple Batch Processing Task Using Spark.

Tools and Programming Languages:

- Programming Language: Python
- Big Data Frameworks: Apache Spark, Apache Hadoop, MongoDB Atlas, Apache Kafka
- Development Environment: Google Colab / Jupyter Notebook
- Supporting Libraries: NumPy, Pandas, Matplotlib.

****

6IMC310- Computer Graphics-Lab

Course Objectives

1. To introduce students to the fundamental concepts, components, and working of computer graphics systems..
2. To develop the ability to implement line, circle, and polygon drawing algorithms.
3. To enable students to apply 2D transformations and viewing techniques in graphics applications..
4. To understand and implement clipping techniques and basic 3D transformations..
5. To study projection methods, hidden surface removal techniques, and illumination models.
6. To explore advanced graphics concepts including OpenGL, animation, color models, ray tracing, and virtual/augmented reality.

Course Outcomes

After completion of this course, Students will be able to:

1. Explain the architecture of graphics systems, display devices, and input devices..
2. Implement DDA, Bresenham, and circle drawing algorithms to generate graphical primitives..
3. Apply 2D transformations and viewing operations on graphical objects.
4. Perform clipping and 3D transformation operations in graphics applications.
5. Develop basic graphics and animation programs using OpenGL.

List of Experiments:

1. Study of Graphics System Components
2. Implementation of DDA Line Drawing Algorithm
3. Implementation of Bresenham Line Drawing Algorithm
4. Circle Drawing using Bresenham/Midpoint Algorithm
5. Implementation of Boundary Fill and Flood Fill Algorithm
6. Implementation of 2D Transformations
7. Implementation of Line Clipping (Cohen-Sutherland Algorithm)
8. Implementation of 3D Transformations
9. Implementation of Projection Techniques
10. Basic Animation using OpenGL

*****

6IMC312AE UNICEF and The Sustainable Development Goals (SDGs)

Course Objectives

1. To introduce students to UNICEF as a key international organization working for children's welfare
2. To develop awareness about the concept and importance of sustainable development
3. To familiarize students with the United Nations' Sustainable Development Goals
4. To help students understand the role of UNICEF in achieving selected SDGs
5. To encourage social responsibility, global citizenship, and community participation among students

Course Outcomes

After completing the course, students will be able to:

1. Explain the purpose, functions, and areas of work of UNICEF
2. Describe the concept of sustainable development and the need for SDGs
3. Identify major Sustainable Development Goals
4. Understand the contribution of UNICEF in achieving selected SDGs
5. Apply values of responsible citizenship and sustainable living.

Unit 1: UNICEF – An Overview (6 hours)
Meaning and full form of UNICEF, History and establishment of UNICEF, Objectives and mission, Major functions of UNICEF, Key areas of work:Child health and nutrition, Education and gender equality, Child protection, Water, sanitation, and hygiene (WASH), UNICEF initiatives in India.

Unit 2: Sustainable Development Goals (SDGs)(7 hours)
Concept of sustainable development, Need and importance of sustainable development, Overview of the 17 Sustainable Development Goals, Study of selected SDGs:No Poverty, Zero Hunger, Good Health and Well-being, Quality Education, Gender Equality, Clean Water and Sanitation, Role of governments, communities, and individuals.

Unit 3: UNICEF and the SDGs (7 hours)
Relationship between UNICEF and SDGs, Role of UNICEF in achieving SDGs, UNICEF's contribution to selected SDGs, SDG-Based Career Opportunities.

Text Books:

1. NCERT, *Contemporary India / Democratic Politics / Economics* (Relevant chapters on development, global organizations, and sustainability), Government of India.
2. UNICEF, *Children's Rights and Sustainable Development* (Student-friendly learning material), UNICEF Publications.

Reference Books:

1. United Nations, *Transforming Our World: The 2030 Agenda for Sustainable Development*.
2. NCERT, *India and the Contemporary World* – Selected sections on global cooperation.
3. IGNOU, *Sustainable Development* – Foundation level reading material.
4. UNICEF India, *Annual Report* (Selected sections for students).

Web Links for Study:

1. UNICEF Official Website: https://www.unicef.org
2. UNICEF India: https://www.unicef.org/india
3. United Nations – Sustainable Development Goals: https://sdgs.un.org
4. National Council of Educational Research and Training (NCERT): https://ncert.nic.in
5. Ministry of Education, Government of India: https://www.education.gov.in

****