North Bengal St. Xavier's College

Department of Computer Science & Application

LESSON PLAN session 2024-2025

1st Sem

Paper: Digital design & Analysis (MAJ1)

Unit 1: Fundamentals of Computers (7 Lectures)

Week 1: Introduction to Computers (2 Lectures)

- Definition and Importance of Computers
- Overview of Computer Systems
- Basic Block Diagram of a Computer
- Von-Neumann Architecture

Week 2: Generation of Computers and Computer Languages (2 Lectures)

- Evolution of Computers (First to Fifth Generation)
- Characteristics of Each Generation
- Classification of Computers: Micro, Mini, Mainframe, and Supercomputers
- Overview of Computer Languages: Machine Language, Assembly Language, High-Level Languages

Week 3: Computer Hardware and Firmware (1 Lecture)

- Definition and Role of Hardware
- Overview of Firmware and Its Functionality
- Common Input and Output Devices
- Storage Classifications: Primary, Secondary, and Tertiary Storage

Week 4: Language Translators (2 Lectures)

- Definition and Purpose of Language Translators
- Types of Translators: Assemblers, Compilers, and Interpreters
- Comparison of Different Translators
- Examples of Commonly Used Translators

Unit 2: Number Systems and Codes (10 Lectures)

Week 5: Number Systems and Conversions (5 Lectures)

- Binary, Octal, Hexadecimal, and Decimal Number Systems

- Interconversion Between Number Systems
- BCD Numbers (8421-2421), Gray Code, Excess–3 Code
- ASCII and EBCDIC Codes: Advantages and Disadvantages

Week 6: Binary Arithmetic and Representations (5 Lectures)

- Binary Addition and Subtraction
- Negative Number Representation: Sign Magnitude, 1's & 2's Complement
- Signed and Unsigned Binary Numbers
- Fixed and Floating-Point Representation

Unit 3: Logic Gates (7 Lectures)

Week 7: Basic Logic Gates (3 Lectures)

- AND, OR, NOT Gates and Their Truth Tables
- NOR, NAND & XOR Gates
- Boolean Algebra: Basic Boolean Laws and Theorems

Week 8: Boolean Functions and Minimization (4 Lectures)

- De-Morgan's Theorem, Boolean Functions and Their Truth Tables
- Minimization Techniques: K-Map for 2, 3, and 4 Variables
- Sum of Product (SOP) & Product of Sum (POS) Forms, Don't Care Conditions

---

Unit 4: Logic Families (7 Lectures)

Week 9: Introduction to Logic Families (3 Lectures)

- Digital Logic Families: RTL, DTL, TTL, ECL, CMOS, IIR, HTL
- Comparative Study of Logic Families

Week 10: Circuit Characteristics (4 Lectures)

- Basic Circuits and Performance Characteristics
- Practical Applications of Logic Families

Unit 5: Combinational Logic (7 Lectures)

Week 11: Basic Combinational Circuits (4 Lectures)

- Half Adder and Full Adder
- Parallel Adder, Half Subtractor, Full Subtractor

- 4-bit Binary Adder cum Subtractor

Week 12: Advanced Combinational Circuits (3 Lectures)

- Multiplexer and Demultiplexer
- Decoder and BCD to Seven-Segment Decoder
- Encoders

Unit 6: Sequential Circuits (7 Lectures)

Week 13: Flip-Flops and Latches (4 Lectures)

- Set-Reset (SR) Latches, D Flip-Flop, R-S Flip-Flop, J-K Flip-Flop
- Master-Slave Flip-Flop, Edge-Triggered Flip-Flop, T Flip-Flop

Week 14: Counters and Registers (3 Lectures)

- Synchronous and Asynchronous Counters
- Up/Down Synchronous Counter, Ripple Counter, Applications of Counters
- Shift Registers: Serial In/Serial Out, Parallel In/Serial Out, Serial In/Parallel Out, Parallel In/Parallel Out
- Bi-Directional Registers and Their Applications

Lab Activities (Practical Sessions - 30 Hours)

1. Identifying Different Hardware Components in a Computer System
2. Demonstration of Input and Output Devices
3. Exploring BIOS and Firmware Settings
4. Writing and Executing a Simple Assembly Language Program
5. Hands-on Practice with Language Translators (Compilers and Interpreters)
6. Implementation of Basic Logic Gates using Breadboard or Software Simulations
7. Verification of De-Morgan's Theorem
8. Design of Half Adder, Full Adder, and Subtractor Circuits
9. Implementation of Multiplexers and Demultiplexers
10. Design and Testing of Flip-Flops (SR, D, JK, T Flip-Flops)
11. Implementation of Synchronous and Asynchronous Counters
12. Working with Shift Registers and Their Applications

Assessment Methods

- **Assignments (20%)**: Weekly quizzes on theoretical concepts
- **Mid-Semester Exam (20%)**: Multiple-choice and short answer questions
- **Project (20%)**: A presentation on computer generations and architecture
- **Final Exam (40%)**: Comprehensive test covering all topics

Course Objective	**Expected learning outcome**
To give the students an idea about Boolean algebra and different techniques for minimization. Also give the idea about gates used in digitalsystem. And also the students an idea about combinational circuit and sequential circuit	Upon the successful completion of this lesson Students will be able to understand the details about computer system**.** Students will be able to minimize Boolean algebra using K-map. Also they will know about logic gates like AND, OR, XOR, etc. Students will be able to design different types of logic circuit like Adder, Multiplexer, decoder, Flip-flop, Register etc.
To give the students an idea about various numbersystems, its representationand different operation performed on them.	Upon the successful completion of this lesson Students will be able to convert between different number system .Also they can add,subtract binary number. Theycan also know the representationof floating point number in computer.
To give the students an idea about Computer Registers,bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt, Interconnection Structures, Bus Interconnection design of basic computer.	Upon the successful completion of this lesson Students will be able to understand different types of Computer Registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt, Interconnection Structures, Bus Interconnection.

3rd Sem

Lesson Plan for MAJOR 3: Computer Architecture and Organization

Course Code: UCMSMAJ11001
Credits: 3
Total Lectures: 45

Unit 1: Register Transfer and Microoperations (6 Lectures)

Week 1: Introduction to Registers and Microoperations (3 Lectures)

- Definition and Purpose of Registers
- Register Transfer and Bus System
- Arithmetic Microoperations (Addition, Subtraction, Increment, Decrement)

Week 2: Logical and Shift Microoperations (3 Lectures)

- Logic Microoperations (AND, OR, XOR, NOT)
- Shift Microoperations (Logical, Circular, and Arithmetic Shifts)
- Interconnection Structures and Bus Interconnection

Unit 2: Basic Computer Organization and Design (7 Lectures)

Week 3: Instruction Set Architecture (3 Lectures)

- Instruction Set and Formats
- Timing and Control Signals
- Instruction Cycle (Fetch, Decode, Execute)

Week 4: Memory and Register References (4 Lectures)

- Hardwired Instruction Format
- Memory Reference, Register Reference, and I/O Instructions
- Design of a Basic Computer and Interrupt Mechanisms

Unit 3: Central Processing Unit (12 Lectures)

Week 5: Stack Organization and Microprogrammed Control (4 Lectures)

- Stack Operations and Stack Organization
- Microprogrammed Control Unit and Instruction Execution
- Microprogrammed Instruction Formats

Week 6: Addressing Modes and Instruction Codes (4 Lectures)

- Addressing Modes and Their Applications
- Instruction Codes and Machine Language
- Assembly Language Programming Basics

Week 7: CPU Architectures (4 Lectures)

- Design of CPU (Control Unit and ALU)
- Comparison of RISC and CISC Architectures

Unit 4: Computer Arithmetic (10 Lectures)

Week 8: Addition and Subtraction (3 Lectures)

- Binary Addition and Subtraction
- Overflow and Carry Operations

Week 9: Multiplication and Division (4 Lectures)

- Multiplication Algorithms (Booth's Algorithm)
- Division Algorithms (Restoring and Non-Restoring Division)

Week 10: Advanced Arithmetic Operations (3 Lectures)

- Floating Point Arithmetic and Normalization
- Error Detection and Correction Techniques

Unit 5: Memory and Input-Output Organization (10 Lectures)

Week 11: Memory Hierarchy (3 Lectures)

- Memory Hierarchy and Performance
- Main Memory Organization and Addressing

Week 12: Cache and Virtual Memory (4 Lectures)

- Cache Memory and Mapping Techniques
- Virtual Memory and Page Replacement Algorithms

Week 13: Input/Output Organization (3 Lectures)

- External Devices and Interfaces
- Programmed I/O, Interrupt-Driven I/O, and Direct Memory Access (DMA)

Lab Activities (Practical Sessions - 30 Hours)

1. Simulation of Register Transfer Operations
2. Implementation of Basic Arithmetic and Logical Microoperations
3. Designing and Implementing Simple Assembly Language Programs
4. Demonstrating Instruction Cycles Using Simulators
5. Implementation of Memory Reference Instructions
6. Experimenting with Cache Memory and Page Replacement Algorithms
7. Simulating CPU Execution for RISC and CISC Architectures
8. Designing Booth's Multiplication and Division Algorithms

9. Implementation of Interrupt-Driven I/O
10. Demonstrating Direct Memory Access (DMA) Operations

Assessment Methods

- **Assignments (20%):** Weekly problem-solving exercises
- **Mid-Semester Exam (20%):** Theory and programming-based assessment
- **Project (20%):** A mini-project on CPU or Memory Organization
- **Final Exam (40%):** Comprehensive test covering all topics

Course Objective	**Expected learning outcome**
To give the students an idea about Boolean algebra and different techniques for minimization. Also give the idea about gates used in digitalsystem. And also the students an idea about combinational circuit and sequential circuit	Upon the successful completion of this lesson Students will be able to understand the details about computer system**.** Students will beable to minimize Boolean algebra using K-map.Also they will knowabout logic gates like AND, OR, XOR, etc. Students will be able to design different types of logic circuit like Adder, Multiplexer, decoder, Flip-flop, Register etc.
To give the students an idea about various numbersystems, its representationand different operation performed on them.	Upon the successful completion of this lesson Students will be able to convert between different number system .Also they can add, subtract binary number. Theycan also know the representation of floating point number in computer.
To give the students an idea about Computer Registers,bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt, Interconnection Structures, Bus Interconnection design of basic computer.	Upon the successful completion of this lesson Students will be able to understand different types of Computer Registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt, Interconnection Structures, Bus Interconnection.

5th Semester

Lesson Plan for CC 51: Internet Technologies

Course Code: CC 51
Credits: 4
Total Lectures: 60

Unit 1: Java (5 Lectures)

Week 1: Introduction to Java and Objects (2 Lectures)

- Use of Objects in Java
- Understanding Classes and Object-Oriented Programming Principles

Week 2: Arrays and ArrayList (3 Lectures)

- Creating and Using Arrays in Java
- Working with ArrayList Class
- Advantages of ArrayList over Arrays

Unit 2: JavaScript (15 Lectures)

Week 3: JavaScript Basics (5 Lectures)

- Introduction to JavaScript
- Data Types and Variables
- Operators and Expressions
- Functions in JavaScript

Week 4: Control Structures and Event Handling (5 Lectures)

- Conditional Statements (if, switch-case)
- Looping Constructs (for, while, do-while)
- JavaScript Events and Event Handling

Week 5: Advanced JavaScript (5 Lectures)

- Working with DOM (Document Object Model)
- Form Validation Techniques
- Error Handling in JavaScript

Unit 3: JDBC (10 Lectures)

Week 6: JDBC Fundamentals (5 Lectures)

- Overview of JDBC and its Importance
- Establishing Database Connectivity
- Working with Connection Interface

- Creating and Executing SQL Statements

Week 7: Advanced JDBC Concepts (5 Lectures)

- Using Prepared Statements and Callable Statements
- Working with Result Set Objects
- Handling Transactions in JDBC
- Best Practices for JDBC Programming

Unit 4: JSP (20 Lectures)

Week 8: Introduction to JSP (5 Lectures)

- Introduction to Java Server Pages (JSP)
- HTTP and Servlet Basics
- Problems with Servlets and Need for JSP
- JSP Page Anatomy and Processing

Week 9: JSP Application Design (5 Lectures)

- JSP with MVC (Model-View-Controller)
- Setting Up the JSP Environment
- Implicit JSP Objects and Their Uses

Week 10: Dynamic Content in JSP (5 Lectures)

- Conditional Processing and Looping in JSP
- Displaying Values Using Expressions
- Declaring Variables and Methods in JSP

Week 11: JSP Debugging and Database Access (5 Lectures)

- Error Handling and Debugging in JSP
- Sharing Data Between JSP Pages, Requests, and Users
- Integrating JSP with Databases using JDBC

Unit 5: Java Beans (10 Lectures)

Week 12: JavaBeans Fundamentals (5 Lectures)

- Introduction to JavaBeans
- Creating and Using JAR Files
- Introspection in JavaBeans

Week 13: Advanced JavaBeans (5 Lectures)

- Developing a Simple JavaBean Component
- Connecting JavaBeans to a Database
- Using JavaBeans in JSP Applications

Lab Activities (Practical Sessions - 30 Hours)

1. Creating and Using Java Objects and Array Lists
2. Writing JavaScript Programs for Form Validation and Event Handling
3. Implementing JDBC for Database Connectivity and SQL Execution
4. Developing JSP Pages for Dynamic Web Applications
5. Setting Up and Configuring JSP and Servlet Environments
6. Working with JSP Implicit Objects and Expressions
7. Debugging JSP Applications and Handling Errors
8. Using JavaBeans in JSP for Data Handling
9. Integrating JSP, JavaBeans, and JDBC for a Complete Web Application
10. Mini-Project: Developing a Dynamic Web-Based Application

Assessment Methods

- **Assignments (20%):** Weekly programming and web development tasks
- **Mid-Semester Exam (20%):** Theory and practical-based assessment
- **Project (20%):** A mini-project on web-based application development
- **Final Exam (40%):** Comprehensive test covering all topics

Course Objective	**Expected learning outcome**
This course will prepare the students to give the idea about Use of Objects, Array and ArrayList class	Upon the successful completion of this lesson Students will Apply the javaobject in different console application and the implementation of array andarray list.
This course will prepare the students to give the idea about the java script Data types, operators, functions, control structures, events and event handling.	The successful completion of this lesson will enable the students to apply theflow control, Define functions and methods. Demonstrate handling web page events. Describe the Document Object Model (DOM).

To give the students an idea about the JDBC Fundamentals, Establishing Connectivity and working with connection interface, working with statements, Creating and Executing SQL Statements, Working with Result Set Objects.	After completing this lesson , students will be able to: Compose JDBC applications DefineJDBC-related terms Describe the major components of a JDBC application Describe the relationship of JDBC to Java and to database systems Describe how to use JDBC in transactions

---

Lesson Plan for Digital Electronics

Course Code: MAJ 3
Credits: 3
Total Lectures: 45

Unit 1: Number Systems (10 Lectures)

Week 1: Introduction to Number Systems (5 Lectures)

- Binary, Octal, Hexadecimal, and Decimal Number Systems
- Interconversion Between Number Systems
- BCD Numbers (8421-2421), Gray Code, Excess–3 Code
- ASCII and EBCDIC Codes: Advantages and Disadvantages

Week 2: Binary Arithmetic and Representations (5 Lectures)

- Binary Addition and Subtraction
- Negative Number Representation: Sign Magnitude, 1's & 2's Complement
- Signed and Unsigned Binary Numbers
- Fixed and Floating-Point Representation

Unit 2: Gates in Digital Design (10 Lectures)

Week 3: Basic Logic Gates (5 Lectures)

- AND, OR, NOT Gates and Their Truth Tables
- NOR, NAND & XOR Gates
- Boolean Algebra: Basic Boolean Laws and Theorems

Week 4: Boolean Functions and Minimization (5 Lectures)

- De-Morgan's Theorem, Boolean Functions and Their Truth Tables
- Minimization Techniques: K-Map for 2, 3, and 4 Variables

- Sum of Product (SOP) & Product of Sum (POS) Forms, Don’t Care Conditions

---

Unit 3: Logic Families (7 Lectures)

Week 5: Introduction to Logic Families (7 Lectures)

- Digital Logic Families: RTL, DTL, TTL, ECL, CMOS, IIR, HTL
- Comparative Study of Logic Families
- Basic Circuit and Performance Characteristics

Unit 4: Combinational Logic (8 Lectures)

Week 6: Basic Combinational Circuits (4 Lectures)

- Half Adder and Full Adder
- Parallel Adder, Half Subtractor, Full Subtractor
- 4-bit Binary Adder cum Subtractor

Week 7: Advanced Combinational Circuits (4 Lectures)

- Multiplexer and Demultiplexer
- Decoder and BCD to Seven-Segment Decoder
- Encoders

Unit 5: Sequential Logic (10 Lectures)

Week 8: Flip-Flops and Latches (5 Lectures)

- Set-Reset (SR) Latches, D Flip-Flop, R-S Flip-Flop, J-K Flip-Flop
- Master-Slave Flip-Flop, Edge-Triggered Flip-Flop, T Flip-Flop

Week 9: Counters and Registers (5 Lectures)

- Synchronous and Asynchronous Counters
- Up/Down Synchronous Counter, Ripple Counter, Applications of Counters
- Shift Registers: Serial In/Serial Out, Parallel In/Serial Out, Serial In/Parallel Out, Parallel In/Parallel Out
- Bi-Directional Registers and Their Applications

Lab Activities (Practical Sessions - 30 Hours)

1. Identifying Different Number Systems and Conversions
2. Performing Binary Arithmetic Operations
3. Implementation of Basic Logic Gates using Breadboard or Software Simulations

4. Verification of De-Morgan's Theorem
5. Design of Half Adder, Full Adder, and Subtractor Circuits
6. Implementation of Multiplexers and Demultiplexers
7. Design and Testing of Flip-Flops (SR, D, JK, T Flip-Flops)
8. Implementation of Synchronous and Asynchronous Counters
9. Working with Shift Registers and Their Applications
10. Mini-Project: Implementing a Simple Digital Circuit

Assessment Methods

- **Assignments (20%):** Weekly problem-solving exercises
- **Mid-Semester Exam (20%):** Theory and practical-based assessments
- **Project (20%):** A mini-project on digital circuit design
- **Final Exam (40%):** Comprehensive test covering all topics

---

Course Objective	**Expected learning outcome**
To give the students an idea about Boolean algebra and different techniques for minimization. Also give the idea about gates used in digitalsystem. And also the students an idea about combinational circuit and sequential circuit	Upon the successful completion of this lesson Students will be able to understand the details about computer system**.** Students will be able to minimize Boolean algebra using K-map. Also they will know about logic gates like AND, OR, XOR, etc. Students will be able to design different types of logic circuit like Adder, Multiplexer, decoder, Flip-flop, Register etc.
To give the students an idea about various numbersystems, its representationand different operation performed on them.	Upon the successful completion of this lesson Students will be able to convert between different number system .Also they can add, subtract binary number. Theycan also know the representation of floating point number in computer.

To give the students an idea about Computer Registers,bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt, Interconnection Structures, Bus Interconnection design of basic computer.	Upon the successful completion of this lesson Students will be able to understand different types of Computer Registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt, Interconnection Structures, Bus Interconnection.

4th Semester

Lesson Plan for Database Management Systems (DBMS)

Course Code: Major 6
Credits: 3
Total Lectures: 45

Unit 1: Introduction to DBMS (10 Lectures)

Week 1: Fundamentals of DBMS (5 Lectures)

- Definition and Purpose of DBMS
- File Processing System vs DBMS
- Limitations of File Processing System
- Characteristics and Components of a Database

- Database Users and Their Roles

Week 2: Database Architecture and Administration (5 Lectures)

- Database System Architecture: 3-Layer and 2-Layer
- Data Independence and Data Dictionary
- View of Data: Data Abstraction, Instances, and Schemas
- Database Administrator (DBA): Roles and Responsibilities

Unit 2: Database Models (7 Lectures)

Week 3: Overview of Database Models (7 Lectures)

- Introduction to Database Models
- Hierarchical Model: Structure and Features
- Network Model: Structure and Features
- Relational Model: Structure and Features
- Comparison of Hierarchical, Network, and Relational Models

Unit 3: Database Design and ER-Diagram (12 Lectures)

Week 4: ER Model and Constraints (6 Lectures)

- Entity-Relationship (ER) Model
- ER Diagrams and Their Components
- Constraints in ER Modeling
- Weak Entity Sets and Their Importance

Week 5: Normalization and Codd's Rules (6 Lectures)

- E. F. Codd's Rules
- Keys in DBMS: Primary, Candidate, Composite, Alternate, Foreign, and Super Keys
- Relational Schemas and Functional Dependencies
- Normalization Process: Lossless and Lossy Decomposition
- Normal Forms: 1NF, 2NF, 3NF, BCNF

Unit 4: Transaction Processing (4 Lectures)

Week 6: ACID Properties and Concurrency Control (4 Lectures)

- Understanding Transactions and ACID Properties
- Concurrency Control Mechanisms

- Introduction to Locking and Deadlocks
- Importance of Transaction Management

Unit 5: SQL and ORACLE (12 Lectures)

Week 7: SQL Basics and Data Operations (6 Lectures)

- Introduction to SQL and Oracle
- Oracle Commands and Data Types
- Operators in SQL: Arithmetic, Logical, Range Searching, Pattern Matching
- SQL Functions: Aggregate, Numeric, String, Conversion, Date, and Time
- Working with Oracle Tables (DUAL Table)

Week 8: SQL Queries and Database Constraints (6 Lectures)

- Grouping Data: GROUP BY and HAVING Clauses
- SQL Language Components: DDL, DML, DCL
- Queries and Subqueries in SQL
- Constraints and Indexes in SQL

Lab Activities (Practical Sessions - 30 Hours)

1. Creating and Managing Databases in Oracle
2. Writing and Executing Basic SQL Queries
3. Implementing Constraints and Keys in SQL
4. Performing Data Normalization Techniques
5. Creating ER Diagrams Using Database Tools
6. Implementing Transactions and ACID Properties in SQL
7. Writing and Using Complex SQL Subqueries
8. Performing Data Aggregation and Grouping in SQL
9. Working with SQL Functions for Data Manipulation
10. Implementing Stored Procedures and Triggers

Assessment Methods

- **Assignments (20%):** Weekly SQL exercises and database design problems
- **Mid-Semester Exam (20%):** Theory and SQL-based assessments
- **Project (20%):** A mini-project on database design and implementation

- **Final Exam (40%):** Comprehensive test covering all topics

- **Course Objectives:**
- **Knowledge acquired:**
- Upon completion of the course, students should be able to design, implement, and manage relational
- databases effectively. They will gain practical skills in SQL, understand database optimization techniques,
- and be prepared for roles in database administration, database development, and related fields.
- **Skills gained:**
- Studying Database Management Systems (DBMS) provides individuals with a variety of skills that are
- valuable in the field of information technology, software development, and data management. Here are key
- skills gained from learning DBMS: Database design, SQL proficiency, data modeling, query optimization,
- Normalization Database security etc.
- **Competency Developed:**
- Studying Database Management Systems (DBMS) develops a range of competencies that are essential for
- individuals pursuing careers in information technology, database administration, software development, and
- data management. Here are key competencies developed from learning DBMS: Database connectivity,
- Backup and Recovery, Database administration, Data warehousing, Data Mining, Distributed database etc.
- These competencies collectively prepare individuals for roles as database administrators, database
- developers, data analysts, and other positions where a deep understanding of DBMS is fundamental for
- effective data management and utilization.

Outcome:

- Upon completion of the course, students should be able to design, implement, and manage relational
- databases effectively. They will gain practical skills in SQL, understand database optimization techniques
- These skills collectively prepare individuals for roles in database administration, database development, data
- analysis, and other areas where a deep understanding of DBMS is crucial for effective data management and utilization.

6th Semester

Lesson Plan for DSE3,E2: Introduction to Data Sciences

Course Code: DSE3
Credits: 4
Total Lectures: 60

Unit 1: Data Scientist's Tool Box (10 Lectures)

Week 1: Introduction to Data Science and Tools (5 Lectures)

- Understanding Data Science and its Applications
- Turning Data into Actionable Knowledge
- Introduction to Version Control: Git and GitHub

Week 2: Development Environment and Tools (5 Lectures)

- Introduction to Markdown and Documentation
- Setting Up R and RStudio
- Basics of Using R for Data Science

Unit 2: R Programming Basics (15 Lectures)

Week 3: Introduction to R (5 Lectures)

- Overview of R Programming Language
- Data Types and Objects in R
- Reading and Writing Data in R

Week 4: Control Structures and Functions (5 Lectures)

- Conditional Statements and Loop Functions
- Writing Functions in R
- Scoping Rules and Variable Environments

Week 5: Advanced R Programming (5 Lectures)

- Working with Dates and Times
- Debugging Tools and Error Handling
- Simulation and Code Profiling

Unit 3: Getting and Cleaning Data (15 Lectures)

Week 6: Data Collection and Acquisition (5 Lectures)

- Obtaining Data from the Web
- Working with APIs and Web Scraping
- Connecting to Databases and Querying Data

Week 7: Data Cleaning Techniques (5 Lectures)

- Handling Missing Data and Outliers
- Data Transformation and Normalization

- Introduction to Tidy Data Principles

Week 8: Organizing and Structuring Data (5 Lectures)

- Reshaping and Merging Datasets
- Working with Various Data Formats (CSV, JSON, XML)
- Automating Data Cleaning Processes

Unit 4: Exploratory Data Analysis (10 Lectures)

Week 9: Summarizing Data (5 Lectures)

- Statistical Summaries and Data Distributions
- Identifying Patterns and Relationships in Data
- Visualizing Data Using ggplot2 and Base R Graphics

Week 10: Multivariate Data Analysis (5 Lectures)

- Principal Component Analysis (PCA) and Feature Selection
- High-Dimensional Data Visualization Techniques
- Clustering and Dimensionality Reduction

Unit 5: Reproducible Research (10 Lectures)

Week 11: Reproducibility Concepts (5 Lectures)

- Importance of Reproducibility in Data Science
- Writing Reports with R Markdown
- Integrating R Code into a Literate Statistical Program

Week 12: Reproducible Workflow (5 Lectures)

- Compiling R Markdown Documents Using knitr
- Organizing Data Analysis for Reproducibility
- Sharing and Publishing Reproducible Research

Lab Activities (Practical Sessions - 30 Hours)

1. Setting up Git and GitHub for Version Control
2. Writing and Executing R Programs
3. Importing, Cleaning, and Formatting Data
4. Implementing Control Structures and Functions in R
5. Exploring and Visualizing Data Using ggplot2

6. Collecting Data Using APIs and Web Scraping
7. Conducting Data Cleaning and Transformation Exercises
8. Performing Multivariate Data Analysis
9. Writing Reports Using R Markdown
10. Compiling and Presenting Reproducible Research

Assessment Methods

- **Assignments (20%):** Weekly programming and data analysis tasks
- **Mid-Semester Exam (20%):** Theory and practical-based assessment
- **Project (20%):** A mini-project on real-world data analysis
- **Final Exam (40%):** Comprehensive test covering all topics

Course Objective	**Expected learning outcome**
The course gives an overview of the data, questions, and tools that data analysts and data scientists work with. There are two components to this course. The first is a conceptual introduction to the ideas behind turning data into actionable knowledge. The second is a practical introduction to the tools that will be used in the program like version control, markdown, git, GitHub, R, and RStudio.	On successful completion of the course, students will be able to learn to Set up R, R-Studio, Github and other useful tools. To Understand the data,problems, to and tools that dataanalysts use Explain essential study design concepts Create a Github repository.
This course enables the students to understand the Overview of R, R data types and objects, reading and writing data, Control structures, functions, scoping rules, dates and times, Loop functions, debugging tools, Simulation, code profiling	At the end of this course, each student will be able to: • Understand basic concepts such as data type and index and use them in their work • Demonstrate use of basic functions • Conceptualize and create loops to solve different types of problems • Create their own customized functions • Construct tables and figures for descriptive statistics Learn to understand new data sets and functions by yourself

This course will cover the basic ways that data can be obtained. The course will cover obtaining data from the web, from APIs, from databases and from colleagues in various formats. It will also cover the basics of data cleaning and how to make data “tidy”. Tidy data dramatically speed downstream data analysis tasks.	On successful completion of the course, students will be able to understand common data storage system, Apply data cleaning basic to make data tidy and obtain usable data from web, APIs, and database