

SEMESTER-III

Course 5: Database Management Systems

Theory

Credits: 3

3 hrs/week

Course Objectives:

- To introduce about database management systems
- To give a good formal foundation on the relational model of data and usage of Relational Algebra
- To introduce the concepts of basic SQL as a universal Database language
- To demonstrate the principles behind systematic database design approaches by covering conceptual design, logical design through normalization
- To provide an overview of physical design of a database system, by discussing Database indexing techniques and storage techniques

Course Outcomes:

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

- Describe a relational database and object-oriented database
- Create, maintain and manipulate a relational database using SQL
- Describe ER model and normalization for database design
- Examine issues in data storage and query processing and can formulate appropriate solutions
- Outline the role and issues in management of data such as efficiency, privacy, security, ethical responsibility, and strategic advantage

Unit 1: Database Fundamentals

Introduction to Database Systems ,File System vs DBMS ,Advantages and Disadvantages of DBMS ,Database Environment ,Database Users ,Three-Level Architecture ,Data Models, Database Languages (DDL, DML, DCL, TCL)

Unit 2: Data Modelling and Database Design

Entity-Relationship (ER) Model ,Entities, Attributes, Relationships ,ER Diagrams ,Mapping ER Model to Relational Model ,Relational Algebra ,Relational Calculus (Introduction) ,Integrity Constraints ,Keys and Relationships

Unit 3: Structured Query Language (SQL)

SQL Basics ,Data Types ,CREATE, ALTER, DROP ,INSERT, UPDATE, DELETE ,SELECT Statement,WHERE, ORDER BY, GROUP BY, HAVING ,Aggregate Functions ,Joins ,Nested Queries ,Views

Unit 4: Normalization and Transaction Processing

Functional Dependencies ,Normalization ,First Normal Form (1NF) ,Second Normal Form (2NF) ,Third Normal Form (3NF) ,Boyce-Codd Normal Form (BCNF) ,Transactions ,ACID Properties ,Concurrency Control ,Locking and Deadlocks

Unit 5. PL/SQL:

Introduction, Shortcomings of SQL, Structure of PL/SQL, PL/SQL Language Elements,Data Types, Operators Precedence, Control Structures, Steps to Create a PL/SQL, Program, Iterative Control, Procedures, Functions.

Text Books:

- 1) Database Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke, TMH
- 2) Database System Concepts,5/e, Silberschatz, Korth, TMH

Reference Books:

- 1) Introduction to Database Systems, 8/e C J Date, PEA.
- 2) Database Management System, 6/e Ramez Elmasri, Shamkant B. Navathe, PEA
- 3) Database Principles Fundamentals of Design Implementation and Management, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall

Encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time

problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity))

B. General

1. Group Discussion
2. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS:

Some of the following suggested assessment methodologies could be adopted;

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Programming exercises,
4. Practical assignments and laboratory reports,
5. Observation of practical skills,
6. Individual and group project reports.
7. Efficient delivery using seminar presentations,
8. Viva voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs form individual and collaborative work.

Semester-III
Course 5: Data Base Management Systems

Practical

Credits: 1

2 hrs/week

List of Experiments:

1. Study the basic SQL environment and write commands to create, save, and execute SQL scripts.

2. Write SQL commands to demonstrate different Database Languages:

- DDL
- DML
- DCL
- TCL

3. Create a database for a Student Management System and perform the following:

- Create tables
- Define appropriate data types
- Assign Primary Keys

4. Design an Entity-Relationship (ER) Diagram for any one of the following systems:

- Library Management System
- Hospital Management System
- College Management System

5. Convert the ER Diagram created in Experiment 4 into a Relational Database Schema.

6. Create tables using appropriate constraints:

7. Demonstrate the use of different types of keys by creating suitable database tables and establishing relationships between them.

8. Write SQL statements to perform the following operations:

CREATE , ALTER, DROP ,TRUNCATE,

9. Write SQL queries using:

INSERT, UPDATE, DELETE

on Employee and Department tables.

10. Write SQL queries using:

- WHERE
- ORDER BY
- GROUP BY
- HAVING

to retrieve meaningful information from a database.

11. Write SQL queries using Aggregate Functions:

12. Write SQL queries using different Joins:

13. Demonstrate Transaction Processing using:

14. Write PL/SQL programs using control structures:

15. Write PL/SQL programs to create and execute:

- Procedures ,Functions and Demonstrate parameter passing and returning values

N. K. Sarvani

N. K. Sarvani

N. K. Sarvani

V. K. Sarvani

J. K. Sarvani

J. K. Sarvani

SEMESTER-III
Course 6: Data Science with R

Theory

Credits: 3

3 hrs/week

Course Objectives:

1. Introduce the data science process, lifecycle, and applications in real-world domains.
2. Build proficiency in R programming for data manipulation, exploration, and visualization.
3. Train students in handling structured, unstructured, and time-based data effectively.
4. Familiarize with basic machine learning and statistical modelling using R.
5. Develop awareness of ethical, interpretability, and responsible use of data science.

Course Outcomes:

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

1. Explain the Data Science process and perform EDA (Exploratory Data Analysis).
2. Write R programs using variables, functions, loops, and packages for basic analytics.
3. Perform data wrangling, cleaning, and visualization with R libraries (dplyr, tidyr).
4. Build and evaluate basic machine learning models such as regression and clustering.
5. Apply data science techniques to practical case studies.

Syllabus:

UNIT I

Introduction to Data Science and R: - Introduction to Data Science, Applications of Data Science, Data Science Process, Introduction to R Programming, Installing R and RStudio, R Data Types (Numeric, Character, Logical), Variables and Operators, Input and Output Functions

UNIT II

R Data Structures: - Vectors, Matrices, Arrays, Lists, Data Frames, Factors, Creating and Manipulating Data Structures, Basic Functions in R

UNIT III

Data Collection and Preprocessing: -Importing data from CSV, Excel, and databases, Reading and writing files in R, handling missing values and duplicates, Data cleaning and formatting, Data transformation using dplyr

UNIT IV

Exploratory Data Analysis (EDA): - Summary statistics: mean, median, mode, variance, Data distribution and outlier detection, Correlation analysis, Data visualization using ggplot2, Creating bar charts, histograms, scatter plots, and box plots

UNIT V

Data Science Applications and Project: - Clustering and segmentation basics, Predictive analytics, Model evaluation techniques, building a simple data science project in R, interpreting results and presenting insights

Text Book

1. R for Data Science, Hadley Wickham and Garrett Golemund, O'Reilly Media, 2nd Edition.
2. Data Science for Business, Foster Provost and Tom Fawcett, O'Reilly Media.
3. The Art of R Programming, Norman Matloff, No Starch Press

Reference Books

1. An Introduction to Statistical Learning, Springer.
2. Practical Data Science with R, Manning Publications.
3. Data Mining with R, Chapman & Hall/CRC.
4. ggplot2: Elegant Graphics for Data Analysis, Springer.
5. Hands-On Programming with R, O'Reilly Media.

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall

Encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity))

B. General

1. Group Discussion
2. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS:

Some of the following suggested assessment methodologies could be adopted;

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Programming exercises,
4. Practical assignments and laboratory reports,
5. Observation of practical skills,
6. Individual and group project reports.
7. Efficient delivery using seminar presentations,
8. Viva voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs form individual and collaborative work.

SEMESTER-III

Course 6: Data Science with R

Practical

Credits: 1

2 hrs/week

List of Experiments:

1. Write an R program to perform basic arithmetic operations using variables.
2. Create and manipulate different R data structures:
 - Vector
 - Matrix
 - List
 - Data Frame
3. Write an R program to demonstrate conditional statements (if, if-else, switch) and loops (for, while).
4. Write an R program to perform basic vector and matrix operations.
5. Import a CSV dataset into R and display its structure and summary.
6. Write an R program to identify and handle missing values in a dataset.
7. Perform data cleaning by removing duplicate rows, renaming columns, and changing data types.
8. Filter, sort, and merge datasets using R.
9. Calculate descriptive statistics (Mean, Median, Mode, Variance, Standard Deviation).
10. Draw the following charts using R:
 - Bar Chart
 - Pie Chart
 - Histogram
11. Create a Box Plot and identify outliers.
12. Draw a Scatter Plot and calculate the correlation coefficient between two variables.
13. Perform exploratory data analysis (EDA) on the Iris or mtcars dataset.
14. Perform Simple Linear Regression and predict values.
15. Perform Multiple Linear Regression on a dataset.
16. Build a Logistic Regression model for classification.

17. Split a dataset into training and testing sets and evaluate the model using a Confusion Matrix and Accuracy.

18. Perform K-Means Clustering on a dataset and visualize the clusters.

19. Analyse the Titanic or Iris dataset and prepare a report with visualizations and conclusions.

20. Develop a Mini Data Science Project in R by:

- Importing a dataset
- Cleaning the data
- Performing EDA
- Building a predictive model
- Visualizing results
- Drawing conclusions

N. K. S.

H. K.

N. H. K.

V. K. Sarvani

Shreelata Shrivastava

S. K.

SEMESTER-III

Course 7: Object Oriented Programming Using Java

Theory

Credits: 3

3 hrs/week

Course Objectives:

1. Introduce core OOP principles and contrast procedural and object-oriented paradigms within the Java ecosystem.
2. Equip learners with foundational and advanced Java syntax including variables, control statements, arrays, strings, and classes.
3. Enable the use of inheritance, polymorphism, interfaces, and exception handling to create maintainable and reusable code.
4. Train students in concurrent programming and stream-based I/O operations including file management and serialization.
5. Empower learners to design, build, and manage GUI programs using Swing components, layout managers, and event handling techniques.

Course Outcomes:

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

1. Apply OOP principles such as encapsulation, inheritance, and polymorphism in Java applications.
2. Write, compile, and debug Java code using control statements, arrays, classes, and methods effectively.
3. Construct modular code leveraging interfaces, abstract classes, and package hierarchies.
4. Manage thread lifecycles, synchronization, and I/O streams for file handling and console interaction.
5. Design user interfaces using Swing and handle keyboard/mouse input through event-driven programming.

Unit 1. OOPs Concepts and Java Programming:

Introduction to Object-Oriented concepts, procedural and object-oriented programming paradigm. Java programming: An Overview of Java, Java Environment, Data types, Variables, constants, scope and life time of variables, operators, type conversion and casting,

Accepting Input from the Keyboard, Reading Input with `Java.util.Scanner` Class, Displaying Output, Displaying Formatted Output with `String.format()`, Control Statements.

Unit 2. Arrays and OOP Constructs:

Arrays, Command Line Arguments, Strings-String Class Methods. Classes & Objects: Creating Classes, declaring objects, Methods, parameter passing, static fields, inheritance: final classes and methods, the object class and its methods; Polymorphism: Dynamic binding, method overriding, abstract classes and methods.

Unit 3. Interfaces, Packages & Exception Handling:

Interfaces VS Abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface; Packages: Defining, creating and accessing a package, importing packages. Exception Handling: Benefits of exception handling, the classification of exceptions, exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exception sub classes.

Unit 4. Multithreading & Stream based I/O:

Differences between multiple processes and multiple threads, thread states, thread life cycle, creating threads, interrupting threads, thread priorities, synchronizing threads, inter thread communication. Stream based I/O (`java.io`): The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, The Console class, Serialization.

Unit 5. GUI Programming with Swing:

Introduction, MVC architecture, components, containers. Understanding Layout Managers - Flow Layout, Border Layout, Grid Layout, Card Layout, GridBag Layout. Event Handling- The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes.

Text Books:

1. Java The Complete Reference, Herbert Schildt, 9th edition, McGraw Hill.
2. Programming in Java, S. Malhotra, S. Chudhary, 2nd edition, Oxford Univ. Press.

Reference Books:

1. Programming with JAVA - A Primer, E Balaguruswamy, 3rd Edition, McGraw Hill.
2. Head First Java: A Brain-Friendly Guide, Kathy Sierra & Bert Bates, 2nd Edition, O'Reilly.

Activities:

Outcome: Apply OOP principles such as encapsulation, inheritance, and polymorphism in Java applications.

Activity: Develop a class hierarchy for a zoo management system using inheritance and public getters/setters.

Evaluation Method: Code review and oral explanation focusing on class relationships, method overriding, and encapsulation practices.

Outcome: Write, compile, and debug Java code using control statements, arrays, classes, and methods effectively.

Activity: Create a console-based student grade calculator using loops, conditionals, arrays, and modular methods.

Evaluation Method: Practical test with debugging tasks and output validation across multiple input scenarios.

Outcome: Construct modular code leveraging interfaces, abstract classes, and package hierarchies.

Activity: Design a payment processing system with abstract classes for Payment, interfaces for Taxable, and organize classes into packages (e.g., com.billing, com.tax).

Evaluation Method: Project submission assessed for modularity, interface implementation, abstraction usage, and package structure.

Outcome: Manage thread lifecycles, synchronization, and I/O streams for file handling and console interaction.

Activity: Build a multithreaded logger that reads input from the console and writes to a file using synchronized threads and buffered streams.

Evaluation Method: Lab demonstration with thread state tracing and file output verification under concurrent input.

Outcome: Design user interfaces using Swing and handle keyboard/mouse input through event-driven programming.

Activity: Create a GUI-based quiz application using Swing components (JFrame, JButton, JTextField) with event listeners for mouse clicks and key presses.

Evaluation Method: Live demo and rubric-based assessment of UI responsiveness, event handling accuracy, and layout design.

SEMESTER-III

Course 7: Object Oriented Programming Using Java

Practical

Credits: 1

2 hrs/week

List of Experiments:

1. Write a Java program to print Fibonacci series.
2. Write a Java program to calculate multiplication of 2 matrices.
3. Write a Java program for sorting a given list of names in ascending order.
4. Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have readAttributes() method to read length and width from the user.
5. Write a Java program that implements method overloading.
6. Write a Java program to implement various types of inheritance: i. Single ii. Multi-Level iii. Hierarchical iv. Hybrid.
7. Write a Java program to implement runtime polymorphism.
8. Write a Java program which accepts withdrawal amount from the user and throws an exception In Sufficient Funds when withdrawal amount is more than available amount.
9. Write a Java program to create three threads that display good morning, for every one second, hello for every 2 seconds and welcome for every 3 seconds by using extending Thread class.
10. Write a Java program that creates three threads. The first thread displays OOPS, the second thread displays through and the third thread displays JAVA by using runnable interface.
11. Write a Java program that displays the number of characters, lines and words in a text file.
12. Implement a Java program for handling mouse events when the mouse entered, exited, clicked, pressed, released, dragged and moved in the client area.
13. Implement a Java program for handling key events when the keyboard is pressed, released, typed.
14. Write a Java swing program that reads two numbers from two separate text fields and displays the sum of two numbers in the third text field when the button add is pressed.
15. Write a Java program to design student registration form using Swing Controls. The form should have the following fields and button SAVE. Form Fields are: Name, RNO, Mailid, Gender, Branch, and Address.

N. K. Singh

N. K. Singh

N. K. Singh

V. K. Sarvani

Abhishek Mishra

Pranav

SEMESTER-IV

Course 8: Data Mining and Data Warehousing

Theory

Credits: 3

3 hrs/week

Course Objectives:

The course should enable the students to learn principles of Data warehousing and data mining with its architecture and understand data preprocessing methods to perform classification and prediction of data. Technical knowledge is helpful to implement Data Mining principles and techniques for real time applications.

Course Outcomes:

Upon successful completion of this course, students should have the knowledge and skills to

1. To understand the principles of Data warehouses and Data Mining.
2. To be familiar with the Data warehouse architecture and its Implementation.
3. To know the Architecture of a Data Mining system.
4. To understand the various Data preprocessing Methods.
5. To perform classification and prediction of data.

Syllabus:

UNIT I

Data Warehousing and Business Analysis: - Data warehouses Components Building a Data warehouse Data Warehouse Architecture DBMS Schemas for Decision Support Data Extraction, Cleanup, and Transformation Tools Metadata reporting Query tools and Applications Online Analytical Processing (OLAP) OLAP and Multidimensional Data Analysis.

UNIT II

Data Mining: - Data Mining Functionalities Data Pre-processing Data Cleaning Data Integration and Transformation Data Reduction Data Discretization and Concept Hierarchy Generation- Architecture Of A Typical Data Mining Systems- Classification Of Data Mining Systems.

Association Rule Mining: - Efficient and Scalable Frequent Item set Mining Methods Mining Various Kinds of Association Rules, Association Mining to Correlation Analysis Constraint-Based Association Mining.

UNIT III

Classification and Prediction: - Issues Regarding Classification and Prediction Classification by Decision Tree Introduction Bayesian Classification Rule Based Classification by Back propagation Support Vector Machines Associative Classification Lazy Learners Other Classification Methods Prediction Accuracy and Error Measures Evaluating the Accuracy of a Classifier or Predictor Ensemble Methods Model Section.

UNIT IV

Cluster Analysis: - Types of Data in Cluster Analysis A Categorization of Major Clustering Methods Partitioning Methods Hierarchical methods Density-Based Methods Grid-Based Methods Model- Based Clustering Methods Clustering High-Dimensional Data Constraint-Based Cluster Analysis Outlier Analysis.

UNIT V

Mining Object, Spatial, Multimedia, Text and Web Data: Multidimensional Analysis and Descriptive Mining of Complex Data Objects Spatial Data Mining, Multimedia Data Mining Text Mining, The World Wide Web.

Text Book

1. Jiawei Han, Micheline Kamber and Jian Mining Concepts and Third Edition, Elsevier, 2011.

Reference Books

1. Alex Berson and Stephen J. Smith Warehousing, Data Mining & Tata McGraw Hill Edition, Tenth Reprint 2007.
2. K.P. Soman, Shyam Diwakar and V. Ajay into Data mining Theory and Easter Economy Edition, Prentice Hall of India, 2006.
3. G. K. Gupta to Data Mining with Case Easter Economy Edition, Prentice Hall of India, 2006.
4. Pang-Ning Tan, Michael Steinbach and Vipin Kumar to Data Pearson Education, 2007.

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall

encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity))

B. General

1. Group Discussion
2. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS:

Some of the following suggested assessment methodologies could be adopted:

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Programming exercises,
4. Practical assignments and laboratory reports,
5. Observation of practical skills,
6. Individual and group project reports.
7. Efficient delivery using seminar presentations,
8. Viva voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs from individual and collaborative work.

SEMESTER-IV

Course 8: Data Mining and Data Warehousing

Practical

Credits: 1

2 hrs/week

Course Objectives: This course enables students to practically implement various data warehousing and data mining techniques using tools such as Python (with libraries like pandas, scikit-learn, and NLTK), SQL databases, or WEKA, to gain hands-on experience in real-world applications.

Course Outcomes:

By the end of this course, students will be able to:

- 1: Design and implement data warehouse schemas, perform ETL processes, and handle data conversions for practical applications. (PO5, PO7)
- 2: Apply data preprocessing techniques, build and evaluate mining models for classification, clustering, and prediction. (PO5, PO7)
- 3: Demonstrate advanced mining techniques including text, web, and multimedia data analysis. (PO4, PO7)

List of Experiments:

1. Design a Star Schema for a Data Warehouse: Using SQL (e.g., MySQL or SQLite), create fact and dimension tables for a retail sales dataset. Populate the tables with sample data and write queries to demonstrate roll-up and drill-down operations.
2. Design a Snowflakes Schema for a Data Warehouse for the schema in Experiment 1
3. Implement ETL Process: Use Python with pandas to extract data from CSV/Excel files, transform it (e.g., handling missing values, normalization), and load it into a SQL database simulating a data warehouse. Visualize the process with flowcharts.
4. Data Cleaning and Preprocessing: Load a noisy dataset. Apply techniques like outlier removal, noise smoothing, and binning to clean the data, then evaluate the impact on data quality using summary statistics.
5. To merge multiple heterogeneous datasets using pandas, perform attribute subset selection , and discretization then export
6. To apply the Apriori algorithm on the market basket dataset and generate association rules
7. Build a decision tree classifier and visualize the tree

8. To implement Web Mining with Scraping and Analysis
9. Implement a Naive Bayes classifier in Python
10. Spatial Clustering for Spatial Data Mining using DBSCAN
11. To perform Hierarchical Clustering
12. To perform Text Mining for Sentiment Analysis
13. To find outliers (unusual or abnormal data points) using the Isolation Forest

N. K. Kulkarni

Hemant

N. H. Kulkarni

V. K. Sarvani

Thorelalakshmi

Prasad

SEMESTER IV
Course 9: Exploratory Data Analysis & Visualization

Theory

Credits:3

3 hrs/week

Course Objectives

1. To introduce the concepts and importance of Exploratory Data Analysis (EDA) in data science.
2. To understand data preprocessing techniques and statistical methods used for data exploration.
3. To apply various visualization techniques for effective communication of insights.
4. To analyze structured datasets using Python libraries and visualization tools.
5. To develop practical skills in interpreting data and presenting analytical results.

Course Outcomes

By the end of the course, students will be able to:

1. Explain the concepts and significance of Exploratory Data Analysis.
2. Perform data cleaning, preprocessing, and transformation techniques.
3. Apply descriptive statistics and exploratory methods to summarize datasets.
4. Create meaningful visualizations using modern visualization libraries.
5. Interpret analytical results and communicate insights through interactive dashboards and reports.

Syllabus

Unit I: Introduction to Exploratory Data Analysis (9 Hours)

Introduction to Data Analytics and EDA, Types of data, Data collection methods, Data quality issues, Data pre-processing concepts, Importance of EDA in the data science lifecycle, CRISP-DM methodology, Introduction to Python for EDA.

Unit II: Data Cleaning and Pre-processing (9 Hours)

Handling missing values, duplicate detection and removal, Outlier detection and treatment, Data transformation techniques, Data normalization and standardization, Encoding categorical variables, Feature engineering basics.

Unit III: Descriptive Statistics and Exploratory Analysis (9 Hours)

Measures of central tendency and dispersion, Distribution analysis, Skewness and kurtosis, Correlation and covariance, Cross-tabulation, Group-wise analysis, Hypothesis generation using EDA techniques.

Unit IV: Data Visualization Techniques (9 Hours)

Principles of effective visualization, Charts and graphs: Bar chart, Line chart, Histogram, Scatter plot, Pie chart, Box plot, Heatmap, Pair plot, Violin plot, Time-series visualization, Geographic visualization basics, Visualization libraries: Matplotlib, Seaborn and Plotly.

Unit V: Interactive Dashboards and Case Studies (9 Hours)

Dashboard design principles, Interactive visualization using Plotly Dash and Tableau Public, Business intelligence concepts, Storytelling with data, Case studies in healthcare, finance, education and retail analytics, Ethical issues in data visualization.

Prescribed Textbooks

- Roger D. Peng – Exploratory Data Analysis with R, Leanpub.
- Claus O. Wilke – Fundamentals of Data Visualization, O'Reilly.
- Cole Nussbaumer Knaflitz – Storytelling with Data, Wiley.

Reference Books

- Wes McKinney – *Python for Data Analysis*, O'Reilly.
- Jake VanderPlas – *Python Data Science Handbook*, O'Reilly.
- Andy Kirk – *Data Visualization: A Handbook for Data Driven Design*, Sage Publications.

SEMESTER IV
Course 9: Exploratory Data Analysis & Visualization

Practical

Credits:1

2 hrs/week

List of Experiments

- Installation and setup of Python environment (Anaconda/Jupyter Notebook/Google Colab).
- Importing datasets using Pandas and NumPy.
- Data cleaning and handling missing values.
- Data transformation and preprocessing.
- Performing descriptive statistical analysis.
- Correlation analysis and feature relationships.
- Creating visualizations using Matplotlib.
- Data visualization using Seaborn and Plotly.
- Building an interactive dashboard using Tableau Public or Power BI.
- Mini Project: End-to-end Exploratory Data Analysis and Visualization of a real-world dataset.

Co-curricular Practical Activities:

- Hands-on workshop on Python libraries for EDA (Pandas, NumPy, Matplotlib, Seaborn, Plotly).
- Dataset exploration using open government datasets (data.gov.in, Kaggle, UCI Repository).
- Prepare visualization reports for healthcare, education or financial datasets.
- Develop an interactive dashboard using Tableau Public or Power BI.
- Student presentations on best practices in data visualization and storytelling.
- Mini Hackathon: Solve a real-world analytical problem using EDA techniques.

N. K. S.

H. S.

N. Maikub

V. K. Sarvani

J. S. S.

S. S.

SEMESTER IV

Course 10: Software Engineering

Theory

Credits: 3

3 hrs/week

Course Objectives

The course aims to provide students with an understanding of the principles and practices of Software Engineering, including:

1. Estimation techniques for software development projects.
2. Software design methodologies.
3. Software testing strategies.
4. Software quality management.
5. Software process models.
6. Software requirements engineering.
7. Software process and product metrics.
8. Risk management.
9. Unified Modelling Language (UML) for software design.

Course Outcomes (COs)

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

- 1:** Translate end-user requirements into system and software requirements using UML and prepare a Software Requirements Document (SRD).
- 2:** Identify and apply appropriate software architectures and design patterns for high-level system design and compare alternative architectural choices.
- 3:** Develop testing strategies, perform software testing, and prepare simple testing reports.

UNIT – I: Introduction to Software Engineering:

The evolving role of software, changing nature of software, software myths. **A Generic View of Process:** Software engineering as a layered technology, a process framework, the Capability Maturity Model Integration (CMMI), process patterns, process assessment, personal process models, and team process models.

Process Models: The Waterfall Model, Incremental Process Models, Evolutionary Process Models, and the Unified Process.

UNIT – II : Software Requirements:

Functional and non-functional requirements, user requirements, system requirements, interface specification, and the Software Requirements Document (SRD).

Requirements Engineering Process: Feasibility studies, requirements elicitation and analysis, requirements validation, and requirements management.

System Models: Context models, behavioural models, data models, object models, and structured methods.

UNIT – III: Design Engineering: Design process and design quality, design concepts, and the design model.

Creating an Architectural Design: Software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, and component diagrams.

UNIT – IV :

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box testing, white-box testing, validation testing, system testing, and the art of debugging.

Product Metrics: Software quality, metrics for the analysis model, metrics for the design model, metrics for source code, metrics for testing, and metrics for maintenance.

UNIT – V:

Metrics for Process and Products: Software measurement and metrics for software quality.

Risk Management: Reactive vs. proactive risk strategies, software risks, risk identification, risk projection, risk refinement, Risk Mitigation, Monitoring, and Management (RMMM), and the RMMM plan.

Quality Management: Quality concepts, Software Quality Assurance (SQA), software reviews, formal technical reviews, statistical software quality assurance, software reliability, and ISO 9000 quality standards.

Text Books

1. **Roger S. Pressman**, *Software Engineering: A Practitioner's Approach*, 6th Edition, McGraw-Hill International.
2. **Ian Sommerville**, *Software Engineering*, 7th Edition, Pearson Education.
3. **Grady Booch, James Rumbaugh, Ivar Jacobson**, *The Unified Modeling Language User Guide*, Pearson Education.

Reference Books

1. James F. Peters & Witold Pedrycz, *Software Engineering: An Engineering Approach*, John Wiley.
2. Waman S. Jawadekar, *Software Engineering: Principles and Practice*, McGraw-Hill.
3. Meilir Page-Jones, *Fundamentals of Object-Oriented Design Using UML*, Pearson Education.

Recommended Co-Curricular Activities

1. **Seminars:** Deliver presentations on emerging Software Engineering topics such as Agile Development, DevOps, Cloud Computing, and Software Testing.
2. **UML Modelling Exercises:** Prepare UML diagrams such as Use Case, Class, Sequence, and Component diagrams for real-world software applications.
3. **Mini Project:** Develop a small software application by following the Software Development Life Cycle (SDLC), including requirement analysis, design, implementation, testing, and documentation.

4. **Software Testing Activity:** Design test cases, perform black-box and white-box testing, identify defects, and prepare a software testing report.
5. **Technical Quiz:** Participate in quizzes covering software process models, software requirements, UML, software testing, software metrics, and quality management.

Recommended Continuous Assessment Methods

1. **Written Assignments:** Assess students through assignments on software process models, requirements engineering, UML diagrams, software testing, and software quality.
2. **Quiz Tests:** Conduct periodic quizzes to evaluate understanding of Software Engineering concepts, SDLC models, testing techniques, metrics, and risk management.
3. **Seminar Presentation:** Evaluate students based on presentations on emerging trends and real-world applications in Software Engineering.
4. **Mini Project:** Assess students through the design, development, testing, and documentation of a small software application following the Software Development Life Cycle (SDLC).
5. **Internal Written Tests:** Conduct unit-wise internal examinations to assess conceptual understanding and problem-solving skills.

SEMESTER-IV

Course 10: Software Engineering

Practical

Credits: 1

2 hrs/week

List of Experiments

Perform the following experiments for any two projects from the sample project list or any other suitable software project.

Experiment 1: Problem Statement Development

Develop a clear and complete problem statement by identifying the objectives, scope, users, and functional requirements of the selected software project.

Experiment 2: Software Requirement Specification (SRS)

Prepare the Software Requirement Specification (SRS) document, software design document, and software testing-related documents following IEEE standards.

Experiment 3: Software Configuration and Risk Management

Prepare Software Configuration Management (SCM) documentation and Risk Management documentation, including risk identification, analysis, mitigation, and monitoring plans.

Experiment 4: Study of CASE Tools

Study the features and applications of any Design Phase CASE (Computer-Aided Software Engineering) tool used for software modelling and documentation.

Experiment 5: Software Design Using CASE Tools

Perform software design using a CASE tool by developing UML diagrams such as Use Case, Class, Sequence, Activity, and Component diagrams.

Experiment 6: Unit and Integration Testing

Develop suitable test cases for Unit Testing and Integration Testing to verify the functionality of the software modules.

Experiment 7: White-box and Black-box Testing

Design and execute test cases using White-box Testing and Black-box Testing techniques and document the testing results.

Experiment 8: Final Project Documentation

Compile all project documents, including SRS, UML diagrams, design documents, SCM plan, Risk Management Plan, test cases, testing reports, and project summary into a comprehensive software project report.

Sample Projects

Students may choose **any two** of the following projects or any other suitable software application.

1. Passport Automation System
2. Book Bank Management System
3. Online Examination Registration System
4. Stock Maintenance System
5. Online Course Reservation System
6. E-Ticketing System
7. Software Personnel Management System
8. Credit Card Processing System
9. E-Book Management System
10. Recruitment Management System
11. **Text Books**

1. **Roger S. Pressman**, *Software Engineering: A Practitioner's Approach*, 6th Edition, McGraw-Hill International.
2. **Ian Sommerville**, *Software Engineering*, 7th Edition, Pearson Education.

N. K. S. S.

N. K. S. S.

N. K. S. S.

N. K. S. S.

N. K. S. S.

N. K. S. S.

KRISHNA UNIVERSITY:: MACHILIPATNAM - 521004

MODEL QUESTION PAPER

SEMESTER-III

COURSE 5: Database Management Systems

Time: 3 hrs.

Max Marks: 70

SECTION – A

Answer any five of the following questions

5 × 4 = 20 M

1. Explain the advantages and disadvantages of a Database Management System (DBMS). Compare it with the traditional file system.(Unit 1)
2. Describe the Three-Level Architecture of DBMS and explain how it provides data independence.(Unit 1)
3. Explain the Entity-Relationship (ER) Model. Discuss entities, attributes, and relationships with a suitable example.(Unit 2)
4. What are keys in a relational database? Explain the different types of keys and their significance.(Unit 2)
5. Explain the SQL SELECT statement and discuss the purpose of WHERE, GROUP BY, HAVING, and ORDER BY clauses with suitable examples.(Unit 3)
6. What is normalization? Explain First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF).(Unit 4)
7. Explain the ACID properties of transactions. Why are they important in database systems?(Unit 4)
8. Why was PL/SQL introduced? Explain the basic structure of a PL/SQL block and its language elements.(Unit 5)

SECTION – B

Answer the following questions

5 × 10 = 50 M

- 9. a)** Explain the Data Science process with a neat diagram. Discuss the importance of each stage in solving real-world problems. **(Unit 1)**

OR

- b)** Explain the features of the R programming language. Discuss R variables, data types, operators, and input/output functions with suitable examples. **(Unit 1)**

10. a) Explain the different data structures available in R. Illustrate Vectors, Matrices, Arrays, Lists, Data Frames, and Factors with suitable examples. **(Unit 2)**

OR

b) Explain the creation and manipulation of different data structures in R. Discuss commonly used built-in functions with suitable examples. **(Unit 2)**

11. a) Explain the process of importing data from CSV and Excel files into R. Discuss data cleaning techniques, handling missing values, and removing duplicate records. **(Unit 3)**

OR

b) Explain data preprocessing in R. Discuss data transformation techniques and the use of the **dplyr** package with suitable examples. **(Unit 3)**

12. a) Explain Exploratory Data Analysis (EDA). Discuss summary statistics, correlation analysis, and methods for detecting outliers with suitable examples. **(Unit 4)**

OR

b) Explain data visualization using **ggplot2** in R. Illustrate the creation of Bar Charts, Histograms, Scatter Plots, and Box Plots with suitable examples. **(Unit 4)**

13. a) Explain clustering techniques and predictive analytics in Data Science. Discuss different methods for evaluating the performance of predictive models. **(Unit 5)**

OR

b) Explain the steps involved in developing a simple Data Science project using R. Discuss data collection, preprocessing, model building, result interpretation, and presentation of insights. **(Unit 5)**

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004
BLUEPRINT
Course 5: Database Management Systems

Unit	S.A.Q Section :A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	1	2	24
III	1	2	24
IV	2	2	28
V	2	2	28
Total	8	10	132

N. Sampath

N. Maikate

U. Maikate

Thirumalaiah

V. K. Sarvani

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004
MODEL QUESTION PAPER

Course 6: Data Science with R

Time: 3 hrs.

Max Marks: 70

SECTION – A

Answer any five of the following questions

5 × 4 = 20 M

1. Explain the applications of Data Science.
2. Differentiate between Numeric, Character, and Logical data types in R with examples.
3. Explain the differences between Vectors and Lists in R.
4. What is a Data Frame? Explain how to create and manipulate a Data Frame.
5. Explain the importance of handling missing values and duplicate records.
6. Write short notes on dplyr and its commonly used functions.
7. Explain Mean, Median, Mode, and Variance with suitable examples.
8. What is Correlation Analysis? Explain its importance in Data Science.

SECTION – B

Answer the following questions

5 × 8 = 40 M

9.a) Explain the Data Science Process with a neat diagram. Discuss the applications of Data Science in different fields.

OR

b) Explain the R programming language. Discuss R data types, variables, operators, and input/output functions with suitable examples.

10.a) Explain the different data structures in R. Illustrate Vectors, Matrices, Arrays, Lists, Data Frames, and Factors with examples.

OR

b) Write R programs to create and manipulate different data structures. Explain basic functions used in R.

11.a) Explain the process of importing data from CSV and Excel files. Discuss data cleaning, handling missing values, and duplicate records in R.

OR

b) Explain data preprocessing in R using suitable examples. Discuss reading/writing files and data transformation using dplyr.

12.a) Explain Exploratory Data Analysis (EDA). Discuss summary statistics, correlation analysis, and outlier detection with examples.

OR

b) Explain data visualization in R using ggplot2. Illustrate Bar Chart, Histogram, Scatter Plot, and Box Plot with suitable examples.

13.a) Explain clustering and predictive analytics in Data Science. Discuss model evaluation techniques with suitable examples.

OR

b) Explain the steps involved in building a simple Data Science project in R. Discuss how to interpret results and present insights.

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004
BLUEPRINT

Course 6: Data Science with R

Unit	S.A.Q Section :A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	1	2	24
III	1	2	24
IV	2	2	28
V	2	2	28
Total	8	10	132

N. K. S. S.

N. H. K.

U. S.

Abhishek Lakshmi

V. K. Sarvani

KRISHNA UNIVERSITY:: MACHILIPATNAM - 521004

MODEL QUESTION PAPER

Course 7: Object Oriented Programming Using Java

Time: 3 hrs.

Max Marks: 70

SECTION-A

*Answer any **FIVE** of the following questions*

5 x 4 M = 20 M

1. What is Object-Oriented Programming? Distinguish between procedural and object-oriented programming paradigms.
2. Explain data types, variables, and the scope and lifetime of variables in Java.
3. Write about command line arguments in Java with a suitable example.
4. Explain any five String class methods with examples.
5. Distinguish between an interface and an abstract class in Java.
6. Explain the life cycle of a thread in Java.
7. Differentiate between byte streams and character streams in Java.
8. Explain the MVC architecture used in Swing programming.

SECTION-B

*Answer **ALL** Questions*
50 M

5 x 10 M =

9. Explain the control statements available in Java with suitable examples.

OR

10. Explain the java.util.Scanner class for reading input and String.format() for displaying formatted output, with examples.
11. What is a class? Explain how classes and objects are created in Java, along with methods and parameter passing.

OR

12. Explain method overriding and dynamic binding in Java.
13. Explain exception handling in Java. Describe the usage of try, catch, throw, throws and finally with an example.

OR

14. Explain how packages are defined, created, accessed, and imported in Java, with a suitable example.
15. Explain how to create threads in Java using the Thread class and the Runnable interface.

OR

16. Explain the File class in Java. Describe how to read from and write to files.
17. Explain the various layout managers available in Java Swing - Flow, Border, Grid, Card, and GridBag Layout.

OR

18. Explain the delegation event model in Java. Describe event sources, event listeners, event classes, and adapter classes with an example of handling mouse and keyboard events.

KRISHNA UNIVERSITY:: MACHILIPATNAM - 521004

BLUEPRINT

Course 7: Object Oriented Programming Using Java

\	Section: A (including choice) 4 Marks	Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	1	2	24
IV	2	2	28
V	1	2	24
Total	8	10	132

N. Samp.

N. Maikta

Umesh

Thirudekatarhini

V. K. Sarvani

KRISHNA UNIVERSITY:: MACHILIPATNAM - 521004
MODEL QUESTION PAPER

Course 8: Data Mining and Data Warehousing

Time: 3 hrs.

Max Marks: 70

SECTION – A

Answer any five of the following questions

5 × 4 = 20 M

1. Differentiate between OLAP and OLTP
2. What is metadata? Explain its role in a data warehouse.
3. Define data transformation and list its techniques.
4. How is correlation analysis applied in association rule mining
5. How is the accuracy of a classifier defined?
6. Explain methods for evaluating classifier accuracy
7. What is grid-based clustering?
8. Explain Density based clustering methods
9. Describe web data mining.
10. List applications of mining complex data types.

SECTION – B

Answer the following questions

5 × 8 = 40 M

11. Discuss the architecture of a data warehouse with a neat diagram.
(OR)
12. Explain Star Schema, Snowflake and Fact Constellation Schema in detail.
13. What is data cleaning? List its methods.
(OR)
14. Explain about the Apriori algorithm for finding frequent item sets with an example
15. Describe classification by decision tree induction with an example.
(OR)
16. Explain about Bayesian classification
17. Categorize major clustering methods and explain partitioning methods
(OR)
18. Explain about Hierarchical methods in cluster analysis
19. Explain spatial data mining techniques with examples.
(OR)
20. Describe text mining processes and applications.

KRISHNA UNIVERSITY:: MACHILIPATNAM - 521004
BLUEPRINT

Course 8: Data Mining and Data Warehousing

Unit	Section: A (including choice) 4 Marks	Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	1	2	24
IV	2	2	28
V	1	2	24
Total	8	10	132

N Kemp.

N. Hinkley

Heinrich

Dore de la Roche

V. K. Sarvani

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004
MODEL QUESTION PAPER
Course 9: Exploratory Data Analysis

Time: 3 hrs.

Max Marks: 70

Answer any FIVE questions:

5 × 4 = 20 Marks

1. Define Exploratory Data Analysis (EDA). Explain its importance.
2. Differentiate between structured and unstructured data.
3. What are missing values? Explain any two methods for handling them?
4. Differentiate between normalization and standardization.
5. Explain correlation with a suitable example.
6. What is a Box Plot? Mention its applications.
7. Write short notes on Plotly and Tableau Public.
8. Explain the importance of dashboards in business analytics.

Section B

Answer ALL questions:

5 × 10 = 50 Marks

1.(a) Explain the concept of Exploratory Data Analysis and discuss the CRISP-DM methodology with suitable examples.

OR

(b) Discuss the data science lifecycle and explain the role of EDA in decision-making.

2.(a) Explain data preprocessing techniques including handling missing values, outlier detection, normalization, and encoding.

OR

(b) Discuss various data transformation techniques and feature engineering with examples.

3.(a) Explain descriptive statistics including measures of central tendency, dispersion, skewness, and kurtosis with suitable examples.

OR

(b) Discuss correlation, covariance, and exploratory analysis techniques used for hypothesis generation.

4.(a) Explain various data visualization techniques such as Histogram, Scatter Plot, Box Plot, Heat Map, Pair Plot, and Violin Plot with suitable examples.

OR

(b) Discuss Matplotlib, Seaborn, and Plotly libraries used for creating effective visualizations.

5.(a) Explain dashboard design principles and discuss interactive dashboards using Tableau Public or Power BI with suitable examples.

OR

(b) Discuss storytelling with data, ethical issues in visualization, and real-world case studies in healthcare, finance, and education.

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004
BLUEPRINT

Course 9: Exploratory Data Analysis

Unit	S.A.Q Section-A (Including Choice) 4 Marks	E.Q Section-B (Including Choice) 10 Marks	Marks Allotted
I	1	1	14
II	2	2	28
III	1	2	24
IV	2	2	28
V	2	3	38
Total	8	10	132

N. K. S. S.

N. H. K.

U. S. S.

T. S. S.

V. K. S.

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004
MODEL QUESTION PAPER

Course 10: Software Engineering

Time: 3 hrs.

Max Marks: 70

SECTION – A

Answer any five of the following questions

5 × 4 = 20 M

1. Define Software Engineering. Explain the evolving role of software and software myths.
2. Explain the layered technology of Software Engineering with a neat diagram.
3. Differentiate between functional and non-functional requirements with suitable examples.
4. What is a Software Requirements Specification (SRS)? List its essential components.
5. Explain the purpose of UML. Draw and explain any one UML diagram.
6. Distinguish between Black-box Testing and White-box Testing.
7. Define software metrics. Explain any four software quality metrics.
8. What is software risk? Explain the steps involved in risk identification.

SECTION – B

Answer the following questions

5 × 8 = 40 M

9. (a) Explain the generic software process framework and discuss the Capability Maturity Model Integration (CMMI).

OR

(b) Describe the Waterfall Model and Incremental Model with their advantages and disadvantages.

10. (a) Discuss the Requirements Engineering Process in detail, including feasibility study, elicitation, analysis, validation, and requirements management.

OR

(b) Explain different system models used in software engineering with suitable examples.

11. (a) Describe the software design process and explain various architectural styles and patterns.

OR

(b) Draw and explain the following UML diagrams:

- Class Diagram
- Sequence Diagram
- Use Case Diagram

12. (a) Explain different software testing strategies, including Black-box Testing, White-box Testing, Validation Testing, and System Testing.

OR

(b) Discuss software product metrics used for analysis, design, source code, testing, and maintenance.

13. (a) Explain software risk management. Discuss reactive and proactive risk strategies and the RMMM plan.

OR

(b) Explain Software Quality Assurance (SQA), Formal Technical Reviews (FTR), Software Reliability, and ISO 9000 Quality Standards.

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004
BLUEPRINT

Course 10: Software Engineering

Unit	S.A.Q Section – A (4 Marks each, including choice)	E.Q Section – B (8 Marks each, including choice)	Marks Allotted
I	2	2	24
II	2	2	24
III	1	2	20
IV	2	2	24
V	1	2	20
Total	8	10	112

N. K. S. S.

N. H. K. S.

U. S. S.

Chandrasekhar

V. K. Sarvani



Krishna University
కృష్ణ విశ్వవిద్యాలయం

(Established under AP Act No. 29 of 2008)

Machilipatnam, Andhra Pradesh, India - 521004



KRISHNA UNIVERSITY
MACHILIPATNAM

BOARD OF STUDIES AY: 2025-26

B.C.A (Data Science)

III Semester to IV Semester

Minor Papers

MEETING HELD ON 14-11-2025

(ONLINE)

Through Google Meet

SEMESTER-III
Minor
Course I: Database Management Systems

Theory

Credits: 3

3 hrs/week

Course Objectives:

- To introduce about database management systems
- To give a good formal foundation on the relational model of data and usage of Relational Algebra
- To introduce the concepts of basic SQL as a universal Database language
- To demonstrate the principles behind systematic database design approaches by covering conceptual design, logical design through normalization
- To provide an overview of physical design of a database system, by discussing Database indexing techniques and storage techniques

Course Outcomes:

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

- Describe a relational database and object-oriented database
- Create, maintain and manipulate a relational database using SQL
- Describe ER model and normalization for database design
- Examine issues in data storage and query processing and can formulate appropriate solutions
- Outline the role and issues in management of data such as efficiency, privacy, security, ethical responsibility, and strategic advantage

Syllabus:

Unit 1: Database Fundamentals

Introduction to Database Systems ,File System vs DBMS ,Advantages and Disadvantages of DBMS ,Database Environment ,Database Users ,Three-Level Architecture ,Data Models, Database Languages (DDL, DML, DCL, TCL)

Unit 2: Data Modelling and Database Design

Entity-Relationship (ER) Model ,Entities, Attributes, Relationships ,ER Diagrams ,Mapping ER Model to Relational Model ,Relational Algebra ,Relational Calculus (Introduction) ,Integrity Constraints ,Keys and Relationships

Unit 3: Structured Query Language (SQL)

SQL Basics ,Data Types ,CREATE, ALTER, DROP ,INSERT, UPDATE, DELETE ,SELECT Statement,WHERE, ORDER BY, GROUP BY, HAVING ,Aggregate Functions ,Joins ,Nested Queries ,Views

Unit 4: Normalization and Transaction Processing

Functional Dependencies ,Normalization ,First Normal Form (1NF) ,Second Normal Form (2NF) ,Third Normal Form (3NF) ,Boyce-Codd Normal Form (BCNF) ,Transactions ,ACID Properties ,Concurrency Control ,Locking and Deadlocks

Unit 5. PL/SQL:

Introduction, Shortcomings of SQL, Structure of PL/SQL, PL/SQL Language Elements,Data Types, Operators Precedence, Control Structures, Steps to Create a PL/SQL, Program, Iterative Control, Procedures, Functions.

Text Books:

- 1) Database Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke, TMH
- 2) Database System Concepts,5/e, Silberschatz, Korth, TMH

Reference Books:

- 1) Introduction to Database Systems, 8/e C J Date, PEA.
- 2) Database Management System, 6/e Ramez Elmasri, Shamkant B. Navathe, PEA
- 3) Database Principles Fundamentals of Design Implementation and Management, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall

Encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity))

B. General

1. Group Discussion
2. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS:

Some of the following suggested assessment methodologies could be adopted;

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Programming exercises,
4. Practical assignments and laboratory reports,

5. Observation of practical skills,
6. Individual and group project reports.
7. Efficient delivery using seminar presentations,
8. Viva voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs form individual and collaborative work.

Semester-III
Minor
Course I: Data Base Management Systems

Practical

Credits: 1

2 hrs/week

List of Experiments:

1. Study the basic SQL environment and write commands to create, save, and execute SQL scripts.

2. Write SQL commands to demonstrate different Database Languages:

- DDL
- DML
- DCL
- TCL

3. Create a database for a Student Management System and perform the following:

- Create tables
- Define appropriate data types
- Assign Primary Keys

4. Design an Entity-Relationship (ER) Diagram for any one of the following systems:

- Library Management System
- Hospital Management System
- College Management System

5. Convert the ER Diagram created in Experiment 4 into a Relational Database Schema.

6. Create tables using appropriate constraints:

7. Demonstrate the use of different types of keys by creating suitable database tables and establishing relationships between them.

8. Write SQL statements to perform the following operations:

CREATE , ALTER, DROP ,TRUNCATE,

9. Write SQL queries using:

INSERT, UPDATE, DELETE

10. Write SQL queries using:

- to retrieve meaningful information from a database.

12. Write SQL queries using different Joins:

14. Write PL/SQL programs using control structures:

15. Write PL/SQL programs to create and execute:

- Procedures ,Functions and Demonstrate parameter passing and returning values

N. Kempf.

Handwritten signature

N. Mark

V. K. Sarvani

Haridev Lakshmi

Yamod

SEMESTER IV

Course II: Software Engineering

Theory

Credits: 3

3 hrs/week

Course Objectives

The course aims to provide students with an understanding of the principles and practices of Software Engineering, including:

1. Estimation techniques for software development projects.
2. Software design methodologies.
3. Software testing strategies.
4. Software quality management.
5. Software process models.
6. Software requirements engineering.
7. Software process and product metrics.
8. Risk management.
9. Unified Modelling Language (UML) for software design.

Course Outcomes (COs)

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

- 1:** Translate end-user requirements into system and software requirements using UML and prepare a Software Requirements Document (SRD).
- 2:** Identify and apply appropriate software architectures and design patterns for high-level system design and compare alternative architectural choices.
- 3:** Develop testing strategies, perform software testing, and prepare simple testing reports.

UNIT – I: Introduction to Software Engineering:

The evolving role of software, changing nature of software, software myths. A Generic View of Process: Software engineering as a layered technology, a process framework, the Capability Maturity Model Integration (CMMI), process patterns, process assessment, personal process models, and team process models.

Process Models: The Waterfall Model, Incremental Process Models, Evolutionary Process Models, and the Unified Process.

UNIT – II : Software Requirements:

Functional and non-functional requirements, user requirements, system requirements, interface specification, and the Software Requirements Document (SRD).

Requirements Engineering Process: Feasibility studies, requirements elicitation and analysis, requirements validation, and requirements management.

System Models: Context models, behavioural models, data models, object models, and structured methods.

UNIT – III: Design Engineering: Design process and design quality, design concepts, and the design model.

Creating an Architectural Design: Software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, and component diagrams.

UNIT – IV :

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box testing, white-box testing, validation testing, system testing, and the art of debugging.

Product Metrics: Software quality, metrics for the analysis model, metrics for the design model, metrics for source code, metrics for testing, and metrics for maintenance.

UNIT – V:

Metrics for Process and Products: Software measurement and metrics for software quality.

Risk Management: Reactive vs. proactive risk strategies, software risks, risk identification, risk projection, risk refinement, Risk Mitigation, Monitoring, and Management (RMMM), and the RMMM plan.

Quality Management: Quality concepts, Software Quality Assurance (SQA), software reviews, formal technical reviews, statistical software quality assurance, software reliability, and ISO 9000 quality standards.

Text Books

1. **Roger S. Pressman**, *Software Engineering: A Practitioner's Approach*, 6th Edition, McGraw-Hill International.
2. **Ian Sommerville**, *Software Engineering*, 7th Edition, Pearson Education.
3. **Grady Booch, James Rumbaugh, Ivar Jacobson**, *The Unified Modeling Language User Guide*, Pearson Education.

Reference Books

1. James F. Peters & Witold Pedrycz, *Software Engineering: An Engineering Approach*, John Wiley.
2. Waman S. Jawadekar, *Software Engineering: Principles and Practice*, McGraw-Hill.
3. Meilir Page-Jones, *Fundamentals of Object-Oriented Design Using UML*, Pearson Education.

Recommended Co-Curricular Activities

1. **Seminars:** Deliver presentations on emerging Software Engineering topics such as Agile Development, DevOps, Cloud Computing, and Software Testing.
2. **UML Modelling Exercises:** Prepare UML diagrams such as Use Case, Class, Sequence, and Component diagrams for real-world software applications.
3. **Mini Project:** Develop a small software application by following the Software Development Life Cycle (SDLC), including requirement analysis, design, implementation, testing, and documentation.

4. **Software Testing Activity:** Design test cases, perform black-box and white-box testing, identify defects, and prepare a software testing report.
5. **Technical Quiz:** Participate in quizzes covering software process models, software requirements, UML, software testing, software metrics, and quality management.

Recommended Continuous Assessment Methods

1. **Written Assignments:** Assess students through assignments on software process models, requirements engineering, UML diagrams, software testing, and software quality.
2. **Quiz Tests:** Conduct periodic quizzes to evaluate understanding of Software Engineering concepts, SDLC models, testing techniques, metrics, and risk management.
3. **Seminar Presentation:** Evaluate students based on presentations on emerging trends and real-world applications in Software Engineering.
4. **Mini Project:** Assess students through the design, development, testing, and documentation of a small software application following the Software Development Life Cycle (SDLC).
5. **Internal Written Tests:** Conduct unit-wise internal examinations to assess conceptual understanding and problem-solving skills.

SEMESTER-IV
Minor

Course II: Software Engineering

Practical

Credits: 1

2 hrs/week

List of Experiments

Perform the following experiments for any two projects from the sample project list or any other suitable software project.

Experiment 1: Problem Statement Development

Develop a clear and complete problem statement by identifying the objectives, scope, users, and functional requirements of the selected software project.

Experiment 2: Software Requirement Specification (SRS)

Prepare the Software Requirement Specification (SRS) document, software design document, and software testing-related documents following IEEE standards.

Experiment 3: Software Configuration and Risk Management

Prepare Software Configuration Management (SCM) documentation and Risk Management documentation, including risk identification, analysis, mitigation, and monitoring plans.

Experiment 4: Study of CASE Tools

Study the features and applications of any Design Phase CASE (Computer-Aided Software Engineering) tool used for software modelling and documentation.

Experiment 5: Software Design Using CASE Tools

Perform software design using a CASE tool by developing UML diagrams such as Use Case, Class, Sequence, Activity, and Component diagrams.

Experiment 6: Unit and Integration Testing

Develop suitable test cases for Unit Testing and Integration Testing to verify the functionality of the software modules.

Experiment 7: White-box and Black-box Testing

Design and execute test cases using White-box Testing and Black-box Testing techniques and document the testing results.

Experiment 8: Final Project Documentation

Compile all project documents, including SRS, UML diagrams, design documents, SCM plan, Risk Management Plan, test cases, testing reports, and project summary into a comprehensive software project report.

Sample Projects

Students may choose **any two** of the following projects or any other suitable software application.

1. Passport Automation System
2. Book Bank Management System
3. Online Examination Registration System
4. Stock Maintenance System
5. Online Course Reservation System
6. E-Ticketing System
7. Software Personnel Management System
8. Credit Card Processing System
9. E-Book Management System
10. Recruitment Management System
11. **Text Books**

1. **Roger S. Pressman**, *Software Engineering: A Practitioner's Approach*, 6th Edition, McGraw-Hill International.
2. **Ian Sommerville**, *Software Engineering*, 7th Edition, Pearson Education. Bottom of Form

N. K. S.

N. K. S.

N. K. S.

N. K. S.

N. K. S.

N. K. S.

KRISHNA UNIVERSITY:: MACHILIPATNAM - 521004

MODEL QUESTION PAPER

SEMESTER-III

COURSE 1 Minor : Database Management Systems

Time: 3 hrs.

Max Marks: 70

SECTION – A

Answer any five of the following questions

5 × 4 = 20 M

1. Explain the advantages and disadvantages of a Database Management System (DBMS). Compare it with the traditional file system.(Unit 1)
2. Describe the Three-Level Architecture of DBMS and explain how it provides data independence.(Unit 1)
3. Explain the Entity-Relationship (ER) Model. Discuss entities, attributes, and relationships with a suitable example.(Unit 2)
4. What are keys in a relational database? Explain the different types of keys and their significance.(Unit 2)
5. Explain the SQL SELECT statement and discuss the purpose of WHERE, GROUP BY, HAVING, and ORDER BY clauses with suitable examples.(Unit 3)
6. What is normalization? Explain First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF).(Unit 4)
7. Explain the ACID properties of transactions. Why are they important in database systems?(Unit 4)
8. Why was PL/SQL introduced? Explain the basic structure of a PL/SQL block and its language elements.(Unit 5)

SECTION – B

Answer the following questions

5 × 10 = 50 M

- 9. a)** Explain the Data Science process with a neat diagram. Discuss the importance of each stage in solving real-world problems. **(Unit 1)**

OR

- b)** Explain the features of the R programming language. Discuss R variables, data types, operators, and input/output functions with suitable examples. **(Unit 1)**

10. a) Explain the different data structures available in R. Illustrate Vectors, Matrices, Arrays, Lists, Data Frames, and Factors with suitable examples. **(Unit 2)**

OR

b) Explain the creation and manipulation of different data structures in R. Discuss commonly used built-in functions with suitable examples. **(Unit 2)**

11. a) Explain the process of importing data from CSV and Excel files into R. Discuss data cleaning techniques, handling missing values, and removing duplicate records. **(Unit 3)**

OR

b) Explain data preprocessing in R. Discuss data transformation techniques and the use of the **dplyr** package with suitable examples. **(Unit 3)**

12. a) Explain Exploratory Data Analysis (EDA). Discuss summary statistics, correlation analysis, and methods for detecting outliers with suitable examples. **(Unit 4)**

OR

b) Explain data visualization using **ggplot2** in R. Illustrate the creation of Bar Charts, Histograms, Scatter Plots, and Box Plots with suitable examples. **(Unit 4)**

13. a) Explain clustering techniques and predictive analytics in Data Science. Discuss different methods for evaluating the performance of predictive models. **(Unit 5)**

OR

b) Explain the steps involved in developing a simple Data Science project using R. Discuss data collection, preprocessing, model building, result interpretation, and presentation of insights. **(Unit 5)**

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004
BLUEPRINT

Course 1 Minor: Database Management Systems

Unit	S.A.Q Section :A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	1	2	24
III	1	2	24
IV	2	2	28
V	2	2	28
Total	8	10	132

N. Venkatesh

N. Maikata

U. Maikata

Thorekalekshmi

V. K. Sarvani

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004
MODEL QUESTION PAPER

Course 2 Minor: Software Engineering

Time: 3 hrs.

Max Marks: 70

SECTION – A

Answer any five of the following questions

5 × 4 = 20 M

1. Define Software Engineering. Explain the evolving role of software and software myths.
2. Explain the layered technology of Software Engineering with a neat diagram.
3. Differentiate between functional and non-functional requirements with suitable examples.
4. What is a Software Requirements Specification (SRS)? List its essential components.
5. Explain the purpose of UML. Draw and explain any one UML diagram.
6. Distinguish between Black-box Testing and White-box Testing.
7. Define software metrics. Explain any four software quality metrics.
8. What is software risk? Explain the steps involved in risk identification.

SECTION – B

Answer the following questions

5 × 8 = 40 M

9. (a) Explain the generic software process framework and discuss the Capability Maturity Model Integration (CMMI).

OR

(b) Describe the Waterfall Model and Incremental Model with their advantages and disadvantages.

10. (a) Discuss the Requirements Engineering Process in detail, including feasibility study, elicitation, analysis, validation, and requirements management.

OR

(b) Explain different system models used in software engineering with suitable examples.

11. (a) Describe the software design process and explain various architectural styles and patterns.

OR

(b) Draw and explain the following UML diagrams:

- Class Diagram
- Sequence Diagram
- Use Case Diagram

12. (a) Explain different software testing strategies, including Black-box Testing, White-box Testing, Validation Testing, and System Testing.

OR

(b) Discuss software product metrics used for analysis, design, source code, testing, and maintenance.

13. (a) Explain software risk management. Discuss reactive and proactive risk strategies and the RMMM plan.

OR

(b) Explain Software Quality Assurance (SQA), Formal Technical Reviews (FTR), Software Reliability, and ISO 9000 Quality Standards.

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004
BLUEPRINT

Course 2 Minor: Software Engineering

Unit	S.A.Q Section – A (4 Marks each, including choice)	E.Q Section – B (8 Marks each, including choice)	Marks Allotted
I	2	2	24
II	2	2	24
III	1	2	20
IV	2	2	24
V	1	2	20
Total	8	10	112

N. Kempf.

Handwritten signature

N. Mark

V. K. Sarvani

Harade Lakshmi

ग्रामोद