



KRISHNA UNIVERSITY
COLLEGE OF ENGINEERING & TECHNOLOGY
MACHILIPATNAM – 521 004 (A.P) INDIA

R 20 Regulations

Computer Science & Engineering

II B.TECH.

Semester-III

S.No	Course Code	Course Name	Category	Hours per week			Credits
				L	T	P	
1.		Discrete Mathematics & Graph Theory	BS	3	0	0	3
2.		Digital Electronics& Microprocessors	ES	3	0	0	3
3.		Advanced Data Structures & Algorithms	PC	3	0	0	3
4.		Object Oriented Programming Through Java	PC	3	0	0	3
5.		Computer Organization	PC	3	0	0	3
6.		Digital Electronics& Microprocessors Lab	ES	0	0	3	1.5
7.		Advanced Data Structures and Algorithms Lab	PC	0	0	3	1.5
8.		Object Oriented Programming Through Java Lab	PC	0	0	3	1.5
9.		Skill Oriented Course_I Web application Development	SC	1	0	2	2
10.		Mandatory noncredit course - II Environmental Science	MC	3	0	0	0
Total							21.5

Semester-IV

S.No	Course Code	Course Name	Category	Hours per week			Credits
				L	T	P	
1.		Deterministic & Stochastic Statistical Methods	BS	3	0	0	3
2.		Database Management Systems	PC	3	0	0	3
3.		Operating Systems	PC	3	0	0	3
4.		Software Engineering	PC	3	0	0	3
5.		Humanities Elective_I Managerial Economics & Financial Analysis Organizational Behaviour Business Environment	HS	3	0	0	3
6.		Database Management Systems Lab	PC	0	0	3	1.5
7.		Operating Systems Lab	PC	0	0	3	1.5
8.		Software Engineering Lab	PC	0	0	3	1.5
9.		Skill Oriented Course_II Exploratory Data Analysis with R	SC	1	0	2	2
10.		Mandatory noncredit course III Design Thinking for Innovation	MC	2	1	0	0
11.		NSS/NCC/NSO Activities	MC	0	0	2	0
Total							21.5
Community Service Internship/Project(Mandatory) for 6 weeks duration during summer vacation							



R 20 Regulations

KRISHNA UNIVERSITY
COLLEGE OF ENGINEERING & TECHNOLOGY
MACHILIPATNAM – 521 004 (A.P) INDIA

Computer Science & Engineering

Note:

1. Eligible and interested students can register either for Honors or for a Minor in IV Semester as per the guidelines issued by the University
2. Students shall register for NCC/NSS/NSO activities and will be required to participate in an activity for two hours in a week during fourth semester.
3. Lateral entry students shall undergo a bridge course in Mathematics during third semester.

Note: The list of experiments not be restricted to the list. Detailed list of programming exercises can be prepared by the concerned faculty members.



Computer Science & Engineering

Course Code	Advanced Data Structures & Algorithms	L	T	P	C
		3	0	0	3
Pre-requisite	Data Structures	Semester		III	

Course Objectives:

- ❖ Learn asymptotic notations, and analyze the performance of different algorithms.
- ❖ Understand and implement various data structures.
- ❖ Learn and implement greedy, divide and conquer, dynamic programming and backtracking algorithms using relevant data structures.
- ❖ Understand non-deterministic algorithms, polynomial and non-polynomial problems.

Course Outcomes (CO):

After completion of the course, students will be able to

- ❖ Analyze the complexity of algorithms and apply asymptotic notations.
- ❖ Apply non-linear data structures and their operations.
- ❖ Understand and apply greedy, divide and conquer algorithms.
- ❖ Develop dynamic programming algorithms for various real-time applications.
- ❖ Illustrate Backtracking algorithms for various applications.

UNIT - I

Introduction to Algorithms

Algorithms, Pseudocode for expressing algorithms, Performance Analysis-Space complexity, Time complexity, Asymptotic Notation- Big oh, Omega, Theta notation and Little oh notation, Polynomial Vs Exponential Algorithms, Average, Best and Worst Case Complexities, Analysing Recursive Programs.

UNIT - II

Trees Part-I

Binary Search Trees: Definition and Operations, AVL Trees: Definition and Operations, Applications.

B Trees: Definition and Operations.

UNIT - III

Trees Part-II

Red-Black Trees, Splay Trees, Applications.

Hash Tables: Introduction, Hash Structure, Hash functions, Linear Open Addressing, Chaining and Applications.

UNIT - IV

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

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

UNIT - V

Dynamic Programming: General method, applications- 0/1 knapsack problem, All pairs shortest path problem, Travelling salesperson problem, Reliability design.

Backtracking: General method, applications-n-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

Introduction to NP-Hard and NP-Complete problems: Basic Concepts.

Textbooks:

1. Data Structures and algorithms: Concepts, Techniques and Applications, G A V Pai.
2. Fundamentals of Computer Algorithms, Ellis Horowitz, Sartaj Sahni and Rajasekharam, Galgotia publications Pvt. Ltd.



Computer Science & Engineering

Reference Books:

1. Classic Data Structures by D. Samanta, 2005, PHI
2. Design and Analysis of Computer Algorithms by Aho, Hopcraft, Ullman 1998, PEA.
3. Introduction to the Design and Analysis of Algorithms by Goodman, Hedetniemi, TMG.

Online Learning Resources:

https://www.tutorialspoint.com/advanced_data_structures/index.asp

<http://peterindia.net/Algorithms.html>



Computer Science & Engineering

Course Code	Object Oriented Programming Through Java		L	T	P	C
			3	0	0	3
Pre-requisite	Fundamental Programming	Semester	III			

Course Objectives:

- ❖ To understand object oriented concepts and problem solving techniques
- ❖ To obtain knowledge about the principles of inheritance and polymorphism
- ❖ To implement the concept of packages, interfaces, exception handling and concurrency mechanism.
- ❖ To design the GUIs using applets and swing controls.
- ❖ To understand the Java Database Connectivity Architecture

Course Outcomes (CO):

After completion of the course, students will be able to

- ❖ Solve real-world problems using OOP techniques.
- ❖ Apply code reusability through inheritance, packages and interfaces
- ❖ Solve problems using java collection framework and I/O classes.
- ❖ Develop applications by using parallel streams for better performance.
- ❖ Develop applets for web applications.
- ❖ Build GUIs and handle events generated by user interactions.
- ❖ Use the JDBC API to access the database

UNIT - I

Introduction: Introduction to Object Oriented Programming, The History and Evolution of Java, Introduction to Classes, Objects, Methods, Constructors, this keyword, Garbage Collection, Data Types, Variables, Type Conversion and Casting, Arrays, Operators, Control Statements, Method Overloading, Constructor Overloading, Parameter Passing, Recursion, String Class and String handling methods.

UNIT - II

Inheritance: Basics, Using Super, Creating Multilevel hierarchy, Method overriding, Dynamic Method Dispatch, Using Abstract classes, Using final with inheritance, Object class,

Packages: Basics, Finding packages and CLASSPATH, Access Protection, Importing packages.

Interfaces: Definition, Implementing Interfaces, Extending Interfaces, Nested Interfaces, Applying Interfaces, Variables in Interfaces.

UNIT - III

Exception handling - Fundamentals, Exception types, Uncaught exceptions, using try and catch, multiple catch clauses, nested try statements, throw, throws and finally, built-in exceptions, creating own exception subclasses.

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, Random access file operations, The Console class, Serialization, Enumerations, Autoboxing, Generics.

UNIT - IV

Multithreading: The Java thread model, Creating threads, Thread priorities, Synchronizing threads, Interthread communication.

The Collections Framework (java.util): Collections overview, Collection Interfaces, The Collectionclasses- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Hashtable, Properties, Stack, Vector, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter, Scanner.



Computer Science & Engineering

UNIT - V

Applet: Basics, Architecture, Applet Skeleton, requesting repainting, using the status window, passing parameters to applets

GUI Programming with Swings – The origin and design philosophy of swing, components and containers, layout managers, event handling, using a push button, jtextfield, jlabel and image icon, the swing buttons, jtext field, jscrollpane, jlist, jcombobox, trees, jtable, An overview of jmenubar, jmenu and jmenuitem, creating a main menu, showmessagedialog, showconfirmdialog, showinputdialog, showoptiondialog, jdialog, create a modeless dialog.

Accessing Databases with JDBC:

Types of Drivers, JDBC Architecture, JDBC classes and Interfaces, Basic steps in developing JDBC applications, Creating a new database and table with JDBC.

Textbooks:

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.
2. Java How to Program, 10th Edition, Paul Dietel, Harvey Dietel, Pearson Education.

Reference Books:

1. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.
2. Core Java Volume – 1 Fundamentals, Cay S. Horstmann, Pearson Education.
3. Java Programming for core and advanced learners, Sagayaraj, Dennis, Karthik and Gajalakshmi, University Press
4. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
5. Object Oriented Programming through Java, P. Radha Krishna, University Press.
6. Programming in Java, S. Malhotra, S. Chaudhary, 2nd edition, Oxford Univ. Press.
7. Java Programming and Object-oriented Application Development, R.A. Johnson, Cengage Learning.

Online Learning Resources:

https://www.w3schools.com/java/java_oop.asp

<http://peterindia.net/JavaFiles.html>



Computer Science & Engineering

Course Code	Computer Organization		L	T	P	C
			3	0	0	3
Pre-requisite	Digital Electronics	Semester	III			

Course Objectives:

- ❖ To learn the fundamentals of computer organization and its relevance to classical and modern problems of computer design
- ❖ To understand the structure and behavior of various functional modules of a computer.
- ❖ To learn the techniques that computers use to communicate with I/O devices
- ❖ To acquire the concept of pipelining and exploitation of processing speed.
- ❖ To learn the basic characteristics of multiprocessors

Course Outcomes (CO):

After completion of the course, students will be able to

- ❖ Understand computer architecture concepts related to the design of modern processors, memories and I/Os
- ❖ Identify the hardware requirements for cache memory and virtual memory
- ❖ Design algorithms to exploit pipelining and multiprocessors
- ❖ Understand the importance and trade-offs of different types of memories.
- ❖ Identify pipeline hazards and possible solutions to those hazards

UNIT - I

Basic Structure of Computer: Computer Types, Functional Units, Basic operational Concepts, Bus Structure, Software, Performance, Multiprocessors and Multicomputer.

Machine Instructions and Programs: Numbers, Arithmetic Operations and Programs, Instructions and Instruction Sequencing, Addressing Modes, Basic Input/output Operations, Stacks and Queues, Subroutines, Additional Instructions.

UNIT - II

Arithmetic: Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-operand Multiplication, Fast Multiplication, Integer Division, Floating-Point Numbers and Operations.

Basic Processing Unit: Fundamental Concepts, Execution of a Complete Instruction, Multiple-Bus Organization, Hardwired Control, and Multi programmed Control.

UNIT - III

The Memory System: Basic Concepts, Semiconductor RAM Memories, Read-Only Memories, Speed, Size and Cost, Cache Memories, Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage.

UNIT - IV

Input/Output Organization: Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces.

UNIT - V

Pipelining: Basic Concepts, Data Hazards, Instruction Hazards, Influence on Instruction Sets.

Large Computer Systems: Forms of Parallel Processing, Array Processors, The Structure of General-Purpose multiprocessors, Interconnection Networks.



Computer Science & Engineering

Textbooks:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, “Computer Organization”, 5th Edition, McGraw Hill Education, 2013.

Reference Books:

1. M.Morris Mano, “Computer System Architecture”, 3rd Edition, Pearson Education.
2. Themes and Variations, Alan Clements, “Computer Organization and Architecture”, CENGAGE Learning.
3. Smruti Ranjan Sarangi, “Computer Organization and Architecture”, McGraw Hill Education.
4. John P.Hayes, “Computer Architecture and Organization”, McGraw Hill Education

Online Learning Resources:

<https://nptel.ac.in/courses/106/103/106103068/>



Computer Science & Engineering

Course Code	Advanced Data Structures and Algorithms Lab		L	T	P	C
			0	0	3	1.5
Pre-requisite	Basics of Data Structures	Semester	III			

Course Objectives:

- Y Learn data structures for various applications.
- Y Implement different operations of data structures by optimizing the performance.
- Y Develop applications using Greedy, Divide and Conquer, dynamic programming.
- Y Implement applications for backtracking algorithms using relevant data structures.

Course Outcomes (CO):

After completion of the course, students will be able to

- Y Understand and apply data structure operations.
- Y Understand and apply non-linear data structure operations.
- Y Apply Greedy, divide and conquer algorithms.
- Y Develop dynamic programming algorithms for various real-time applications.
- Y Illustrate and apply backtracking algorithms, further able to understand non-deterministic algorithms.

List of Experiments:

1. Write a program to implement the following operations on Binary Search Tree:
a) Insert b) Delete c) Search d) Display
2. Write a program to perform a Binary Search for a given set of integer values.
3. Write a program to implement Splay trees.
4. Write a program to implement Merge sort for the given list of integer values.
5. Write a program to implement Quicksort for the given list of integer values.
6. Write a program to find the solution for the knapsack problem using the greedy method.
7. Write a program to find minimum cost spanning tree using Prim's algorithm
8. Write a program to find minimum cost spanning tree using Kruskal's algorithm
9. Write a program to find a single source shortest path for a given graph.
10. Write a program to find the solution for job sequencing with deadlines problems.
11. Write a program to find the solution for a 0-1 knapsack problem using dynamic programming.
12. Write a program to solve Sum of subsets problem for a given set of distinct numbers using backtracking.
13. Implement N Queen's problem using Back Tracking.

References:

1. Y Daniel Liang, "Introduction to Programming using Python", Pearson.
2. Benjamin Baka, David Julian, "Python Data Structures and Algorithms", Packt Publishers, 2017.
3. Rance D. Necaie, "Data Structures and Algorithms using Python", Wiley Student Edition.

Online Learning Resources/Virtual Labs:

- <http://cse01-iiith.vlabs.ac.in/>
- <http://peterindia.net/Algorithms.html>

Note: The list of experiments not be restricted to the above list. Detailed list of programming exercises can be prepared by the concerned faculty members.



Computer Science & Engineering

Course Code	Object Oriented Programming Through Java Lab		L	T	P	C
			0	0	3	1.5
Pre-requisite	Fundamental Programming	Semester	III			

Course Objectives:

- ❖ To introduce the concepts of Java.
- ❖ To Practice object-oriented programs and build java applications.
- ❖ To implement java programs for establishing interfaces.
- ❖ To implement sample programs for developing reusable software components.
- ❖ To establish database connectivity in java and implement GUI applications.

Course Outcomes (CO):

After completion of the course, students will be able to

- ❖ Recognize the Java programming environment.
- ❖ Develop efficient programs using multithreading.
- ❖ Design reliable programs using Java exception handling features.
- ❖ Extend the programming functionality supported by Java.
- ❖ Select appropriate programming constructs to solve a problem.

List of Experiments:

Week-1

a. Installation of Java software, study of any Integrated development environment, Use Eclipse or Netbeans platform and acquaint with the various menus. Create a test project, add a test class and run it.

See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with java program to find prime numbers between 1 to n.

b. Write a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a,b, c and use the quadratic formula.

c. Develop a Java application to generate Electricity bills. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e domestic or commercial). Compute the bill amount using the following tariff.

If the type of the EB connection is domestic, calculate the amount to be paid as follows:

- γ First 100 units - Rs. 1 per unit
- γ 101-200 units - Rs. 2.50 per unit
- γ 201 -500 units - Rs. 4 per unit
- γ > 501 units - Rs. 6 per unit

If the type of the EB connection is commercial, calculate the amount to be paid as follows:

- γ First 100 units - Rs. 2 per unit
- γ 101-200 units - Rs. 4.50 per unit
- γ 201 -500 units - Rs. 6 per unit
- γ > 501 units - Rs. 7 per unit

d. Write a Java program to multiply two given matrices.

Week-2

a. Write Java program on use of inheritance, preventing inheritance using final, abstract classes.

b. Write Java program on dynamic binding, differentiating method overloading and overriding.

c. Develop a java application to implement currency converter (Dollar to INR, EURO to INR, Yen) using

Interfaces.

Week-3

a. Write Java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read, display it only if it's not a duplicate of any number already read display the complete set of unique values input after the user enters each new value.



Computer Science & Engineering

b. Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

c. Write a Java program to read the time intervals (HH:MM) and to compare system time if the system Time between your time intervals print correct time and exit else try again to repute the same thing. By using StringTokenizer class.

Week-4

a. Write a Java program to implement user defined exception handling.

b. Write java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it's not a duplicate of any number already read. Display the complete set of unique values input after the user enters each new value.

Week-5

a. Write a Java program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

b. Write a Java program that creates three threads. First thread displays —Good Morning! every one second, the second thread displays —Hello! every two seconds and the third thread displays —Welcome! every three seconds.

Week-6

a. Write a java program to split a given text file into n parts. Name each part as the name of the original file followed by .part where n is the sequence number of the part file.

b. Write a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.

Week-7

a. Write a java program that displays the number of characters, lines and words in a text file.

b. Write a java program that reads a file and displays the file on the screen with line number before each line.

Week-8

a. Write a Java program that correctly implements the producer-consumer problem using the concept of inter thread communication.

b. Develop a Java application for stack operation using Buttons and JOptionPane input and Message dialog box.

c. Develop a Java application to perform Addition, Division, Multiplication and subtraction using the JOptionPane dialog Box and Textfields.

Week-9

a. Develop a Java application for the blinking eyes and mouth should open while blinking.

b. Develop a Java application that simulates a traffic light. The program lets the user select one of the three lights: Red, Yellow or Green with radio buttons. On selecting a button an appropriate message with —STOP! or —READY! or !GO! should appear above the buttons in the selected color. Initially, there is no message shown.

Week-10

a. Develop a Java application to implement the opening of a door while opening man should present before hut and closing man should disappear.

b. Develop a Java application by using JTextField to read decimal values and converting a decimal number into a binary number then print the binary value in another JTextField.

Week-11

a. Develop a Java application that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. Use adapter classes.

b. Develop a Java application to demonstrate the key event handlers.



Computer Science & Engineering

Week-12

- a. Develop a Java application to find the maximum value from the given type of elements using a generic function.
- b. Develop a Java application that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result.
- c. Develop a Java application for handling mouse events.

Week-13

- a. Develop a Java application to establish a JDBC connection, create a table student with properties name, register number, mark1, mark2, mark3. Insert the values into the table by using java and display the information of the students at front end.

References:

1. P. J. Deitel, H. M. Deitel, “Java for Programmers”, Pearson Education, PHI, 4th Edition, 2007.
2. P. Radha Krishna, “Object Oriented Programming through Java”, Universities Press, 2nd Edition, 2007
3. Bruce Eckel, “Thinking in Java”, Pearson Education, 4th Edition, 2006.
4. Sachin Malhotra, Saurabh Chaudhary, “Programming in Java”, Oxford University Press, 5th Edition, 2010.

Online Learning Resources/Virtual Labs:

<https://java-iitd.vlabs.ac.in/>

<http://peterindia.net/JavaFiles.html>

Note: The list of experiments not be restricted to the above list. Detailed list of programming exercises can be prepared by the concerned faculty members.



Computer Science & Engineering

Course Code	Web Application Development	L	T	P	C
		1	0	2	2
Pre-requisite		Semester		III	

Course Objectives:

- Υ Learn website development using HTML, CSS, JavaScript.
- Υ Understand the concepts of responsive web development using the bootstrap framework
- Υ Make use of the JQueryjavascript library to provide interactiveness to the websites.
- Υ Discover how to use Google Charts to provide a better way to visualize data on a website
- Υ Learn Content Management Systems to speed the development process

Course Outcomes (CO):

After completion of the course, students will be able to

- Υ Construct web sites with valid HTML, CSS, JavaScript
- Υ Create responsive Web designs that work on phones, tablets, or traditional laptops and wide-screen monitors.
- Υ Develop websites using jQuery to provide interactivity and engaging user experiences
- Υ Embed Google chart tools in a website for better visualization of data.
- Υ Design and develop web applications using Content Management Systems like WordPress

Activities:

Module - 1:

HTML: What is a browser?, What is HTML?, Elements and Tags, Basic HTML5 structure, Metadata, <title>, Adding favicon, Comments, headings

Task: Create a Basic HTML document

Module - 2:

HTML (continued): Block-Level Elements & Inline Elements, Links (Understand Absolute vs Relative paths), Lists, Images, iframe (embed youtube video)

Task: Create your Profile Page

Module - 3:

HTML (continued): Tables: <table>, <tr>, <th>, <td>, Attributes for each Table element

Task: Create a Class Timetable (to merge rows/columns, use rowspan/colspan)

Module - 4:

HTML (continued): Form Elements: <input>, <select>, <textarea>, <button>, Attributes for each Form element

Task: Create a Student Hostel Application Form

Module - 5:

Cascading Style Sheets (CSS): CSS Properties, Types of CSS, Selectors, box model, Pseudo-elements, z-index

Task: Make the Hostel Application Form designed in Module -4 beautiful using CSS (add colors, backgrounds, change font properties, borders, etc.)

Module - 6:

Bootstrap - CSS Framework: Layouts (Containers, Grid system), Forms, Other Components

Task: Style the Hostel Application Form designed in Module-5 still more beautiful using Bootstrap CSS (Re-size browser and check how the webpage displays in mobile resolution)

Module - 7:

HTTP & Browser Developer Tools: Understand HTTP Headers (Request & Response Headers), URL & its Anatomy, Developer Tools: Elements/Inspector, Console, Network, Sources, performance, Application Storage.



Computer Science & Engineering

Task: Analyse various HTTP requests (initiators, timing diagrams, responses) and identify problems if any.

Module - 8:

Javascript: Variables, Data Types, Operators, Statements, Objects, Functions, Events & Event Listeners, DOM.

Task: Design a simple calculator using JavaScript to perform sum, product, difference, and quotient operations:

Module - 9:

Dynamic HTML with JavaScript: Manipulate DOM, Error Handling, Promises, async/await, Modules.

Task: Design & develop a Shopping Cart Application with features including Add Products, Update Quantity, Display Price (Sub-Total & Total), Remove items/products from the cart.

Module - 10:

JQuery - A Javascript Library: Interactions, Widgets, Effects, Utilities, Ajax using JQuery.

Task: Validate all Fields and Submit the Hostel Application Form designed in Module-6 using JQuery

Module - 11:

Google Charts: Understand the Usage of Pie chart, Bar Chart, Histogram, Area & Line Charts, Gantt Charts.

Task: Develop an HTML document to illustrate each chart with real-time examples.

Module - 12:

Open Source CMS (Content Management System): What is a CMS?, Install CMS, Themes, Plugins.

Task: Develop an E-learning website using any CMS (for example WordPress)

References:

1. Deitel and Deitel and Nieto, —Internet and World Wide Web - How to Program, Prentice Hall, 5th Edition, 2011.
2. Web Technologies, Uttam K. Roy, Oxford Higher Education., 1st edition, 10th impression, 2015.
3. Stephen Wynkoop and John Burke —Running a Perfect Website, QUE, 2nd Edition, 1999.
4. Jeffrey C and Jackson, —Web Technologies A Computer Science Perspective Pearson Education, 2011.
5. Gopalan N.P. and Akilandeswari J., —Web Technology, Prentice Hall of India, 2011.

Online Learning Resources/Virtual Labs:

- a. HTML: <https://html.spec.whatwg.org/multipage/>
- b. HTML: <https://developer.mozilla.org/en-US/docs/Glossary/HTML5>
- c. CSS: <https://www.w3.org/Style/CSS/>
- d. Bootstrap - CSS Framework: <https://getbootstrap.com/>
- e. Browser Developer Tools: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_are_browser_developer_tools
- f. Javascript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- g. JQuery: <https://jquery.com>
- h. Google Charts: <https://developers.google.com/chart>
- i. Wordpress: <https://wordpress.com>

Note: The list of experiments not be restricted to the above list. Detailed list of programming exercises can be prepared by the concerned faculty members.



Computer Science & Engineering

Course Code	DATABASE MANAGEMENT SYSTEMS			L	T	P	C
				3	0	0	3
Pre-requisite	NIL	Semester	IV				

Course Objectives:

This course is designed to:

- ❖ Train in the fundamental concepts of database management systems, database modeling and design, SQL, PL/SQL and system implementation techniques.
- ❖ Enable students to model ER diagrams for any customized application
- ❖ Inducting appropriate strategies for optimization of queries.
- ❖ Provide knowledge on concurrency techniques
- ❖ Demonstrate the organization of Databases

Course Outcomes (CO):

After completion of the course, students will be able to

- ❖ Design a database for a real-world information system
- ❖ Define transactions that preserve the integrity of the database
- ❖ Generate tables for a database
- ❖ Organize the data to prevent redundancy
- ❖ Pose queries to retrieve the information from the database.

UNIT - I

Introduction: Database systems applications, Purpose of Database Systems, view of Data, Database Languages, Relational Databases, Database Design, Data Storage and Querying, Transaction Management, Database Architecture, Data Mining and Information Retrieval, Specialty Databases, Database users and Administrators, **Introduction to Relational Model:** Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, Relational Operations

UNIT - II

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Sub-queries, Modification of the Database. **Intermediate SQL:** Joint Expressions, Views, Transactions, Integrity Constraints, SQL Data types and schemas, Authorization. **Advanced SQL:** Accessing SQL from a Programming Language, Functions and Procedures, Triggers, Recursive Queries, OLAP, Formal relational query languages.

UNIT - III

Database Design and the E-R Model: Overview of the Design Process, The Entity-Relationship Model, Constraints, Removing Redundant Attributes in Entity Sets, Entity-Relationship Diagrams, Reduction to Relational Schemas, Entity-Relationship Design Issues. **Relational Database Design:** Features of Good Relational Designs, Atomic Domains and First Normal Form, Decomposition Using Functional Dependencies, Functional-Dependency Theory, Algorithms for Decomposition, Decomposition Using Multivalued Dependencies, More Normal Forms.

UNIT - IV

Query Processing: Overview, Measures of Query cost, Selection operation, sorting, Join Operation, other operations, Evaluation of Expressions. **Query optimization:** Overview, Transformation of Relational Expressions, Estimating statistics of Expression results, Choice of Evaluation Plans, Materialized views, Advanced Topics in Query Optimization.

UNIT - V

Transaction Management, Concurrency Control, Recovery System

Transaction Management:

Transactions: Concept, A Simple Transactional Model, Storage Structures, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Isolation and Atomicity, Transaction Isolation Levels, Implementation of Isolation Levels, Transactions as SQL Statements.



KRISHNA UNIVERSITY
COLLEGE OF ENGINEERING & TECHNOLOGY
MACHILIPATNAM – 521 004 (A.P) INDIA

R 20 Regulations

Computer Science & Engineering

Concurrency Control: Lock-based Protocols, Deadlock Handling, Multiple granularity, Timestamp-based Protocols, and Validation-based Protocols.

Recovery System: Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm, Buffer Management, Failure with Loss of Nonvolatile Storage, Early Lock Release and Logical Undo Operations.

Textbooks:

1. A.Silberschatz, H.F.Korth, S.Sudarshan, “Database System Concepts”,6/e, TMH 2019

Reference Books:

1. Database Management System, 6/e RamezElmasri, Shamkant B. Navathe, PEA
2. Database Principles Fundamentals of Design Implementation and Management, Carlos Coronel, Steven Morris, Peter Robb, Cengage Learning.
- 3.Database Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke,TMH

Online Learning Resources:

https://onlinecourses.nptel.ac.in/noc21_cs04/preview



Computer Science & Engineering

Course Code	OPERATING SYSTEMS		L	T	P	C
			3	0	0	3
Pre-requisite	Basics of CO and DBMS	Semester	IV			

Course Objectives:

The course is designed to

- ❖ Understand basic concepts and functions of operating systems
- ❖ Understand the processes, threads and scheduling algorithms.
- ❖ Provide good insight on various memory management techniques
- ❖ Expose the students with different techniques of handling deadlocks
- ❖ Explore the concept of file-system and its implementation issues
- ❖ Familiarize with the basics of the Linux operating system
- ❖ Implement various schemes for achieving system protection and security

Course Outcomes (CO):

After completion of the course, students will be able to

- ❖ Realize how applications interact with the operating system
- ❖ Analyze the functioning of a kernel in an Operating system.
- ❖ Summarize resource management in operating systems
- ❖ Analyze various scheduling algorithms
- ❖ Examine concurrency mechanism in Operating Systems
- ❖ Apply memory management techniques in the design of operating systems
- ❖ Understand the functionality of the file system
- ❖ Compare and contrast memory management techniques.
- ❖ Understand deadlock prevention and avoidance.
- ❖ Perform administrative tasks on Linux based systems.

UNIT - I

Operating Systems Overview: Introduction, Operating system functions, Operating systems operations, Computing environments, Open-Source Operating Systems

System Structures: Operating System Services, User and Operating-System Interface, systems calls, Types of System Calls, system programs, Operating system Design and Implementation, Operating system structure, Operating system debugging, System Boot.

UNIT - II

Process Concept: Process scheduling, Operations on processes, Inter-process communication, Communication in client server systems.

Multithreaded Programming: Multithreading models, Thread libraries, Threading issues, Examples.

Process Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling, Thread scheduling, Examples.

Inter-process Communication: Race conditions, Critical Regions, Mutual exclusion with busy waiting, Sleep and wakeup, Semaphores, Mutexes, Monitors, Message passing, Barriers, Classical IPC Problems - Dining philosophers problem, Readers and writers problem.

UNIT - III

Memory-Management Strategies: Introduction, Swapping, Contiguous memory allocation, Paging, Segmentation, Examples.

Virtual Memory Management: Introduction, Demand paging, Copy on-write, Page replacement, Frame allocation, Thrashing, Memory-mapped files, Kernel memory allocation, Examples.



Computer Science & Engineering

Deadlocks: Resources, Conditions for resource deadlocks, Ostrich algorithm, Deadlock detection And recovery, Deadlock avoidance, Deadlock prevention.

File Systems: Files, Directories, File system implementation, management and optimization.

Secondary-Storage Structure: Overview of disk structure, and attachment, Disk scheduling, RAID structure, Stable storage implementation.

UNIT - V

System Protection, System Security

Lecture 8Hrs

System Protection: Goals of protection, Principles and domain of protection, Access matrix, Access control, Revocation of access rights.

System Security: Introduction, Program threats, System and network threats, Cryptography as a security, User authentication, implementing security defenses, firewalling to protect systems and networks, Computer security classification.

Case Studies: Linux, Microsoft Windows.

Textbooks:

1. Silberschatz A, Galvin P B, and Gagne G, Operating System Concepts, 9th edition, Wiley, 2016.
2. Tanenbaum A S, Modern Operating Systems, 3rd edition, Pearson Education, 2008. (Topics: Inter-process Communication and File systems.)

Reference Books:

1. Tanenbaum A S, Woodhull A S, Operating Systems Design and Implementation, 3rd edition, PHI, 2006.
2. Dhamdhare D M, Operating Systems A Concept Based Approach, 3rd edition, Tata McGraw-Hill, 2012.
3. Stallings W, Operating Systems -Internals and Design Principles, 6th edition, Pearson Education, 2009
4. Nutt G, Operating Systems, 3rd edition, Pearson Education, 2004

Online Learning Resources:

<https://nptel.ac.in/courses/106/106/106106144/>

<http://peterindia.net/OperatingSystems.html>



Computer Science & Engineering

Course Code	Software Engineering	L	T	P	C
		3	0	0	3
Pre-requisite	Semester	IV			

Course Objectives:

- ❖ To learn the basic concepts of software engineering and life cycle models
- ❖ To explore the issues in software requirements specification and enable to write SRS documents for software development problems
- ❖ To elucidate the basic concepts of software design and enable to carry out procedural and object oriented design of software development problems
- ❖ To understand the basic concepts of black box and white box software testing and enable to design test cases for unit, integration, and system testing
- ❖ To reveal the basic concepts in software project management

Course Outcomes (CO):

After completion of the course, students will be able to

- ❖ Obtain basic software life cycle activity skills.
- ❖ Design software requirements specifications for given problems.
- ❖ Implement structure, object oriented analysis and design for given problems.
- ❖ Design test cases for given problems.

Y Apply quality management concepts at the application level.

UNIT - I

Basic concepts in software engineering and software project management

Basic concepts: abstraction versus decomposition, evolution of software engineering techniques, Software development life cycle (SDLC) models: Iterative waterfall model, Prototype model, Evolutionary model, Spiral model, RAD model, Agile models, software project management: project planning, project estimation, COCOMO, Halstead's Software Science, project scheduling, staffing, Organization and team structure, risk management, configuration management.

UNIT - II

Requirements analysis and specification

The nature of software, The Unique nature of Webapps, Software Myths, Requirements gathering and analysis, software requirements specification, Traceability, Characteristics of a Good SRS Document, IEEE 830 guidelines, representing complex requirements using decision tables and decision trees, overview of formal system development techniques, axiomatic specification, algebraic specification.

UNIT - III

Software Design

Good Software Design, Cohesion and coupling, Control Hierarchy: Layering, Control Abstraction, Depth and width, Fan-out, Fan-in, Software design approaches, object oriented vs. function oriented design. Overview of SA/SD methodology, structured analysis, Data flow diagram, Extending DFD technique to real life systems, Basic Object oriented concepts, UML Diagrams, Structured design, Detailed design, Design review, Characteristics of a good user interface, User Guidance and Online Help, Mode-based vs Mode-less Interface, Types of user interfaces, Component-based GUI development, User interface design methodology: GUI design methodology.

UNIT - IV

Coding and Testing

Coding standards and guidelines, code review, software documentation, Testing, Black Box Testing, White Box Testing, debugging, integration testing, Program Analysis Tools, system testing, performance testing, regression testing, Testing Object Oriented Programs.

UNIT - V

Software quality, reliability, and other issues

Software reliability, Statistical testing, Software quality and management, ISO 9000, SEI capability maturity model (CMM), Personal software process (PSP), Six sigma, Software quality metrics, CASE and its scope, CASE environment, CASE support in software life cycle, Characteristics of software maintenance, Software reverse engineering, Software maintenance processes model, Estimation maintenance cost. Basic issues in any reuse program, Reuse approach, Reuse at organization level.



KRISHNA UNIVERSITY
COLLEGE OF ENGINEERING & TECHNOLOGY
MACHILIPATNAM – 521 004 (A.P) INDIA

R 20 Regulations

Computer Science & Engineering

Textbook:

1. Rajib Mall, “Fundamentals of Software Engineering”, 5th Edition, PHI, 2018.
2. Pressman R, “Software Engineering- Practioner Approach”, McGraw Hill.

Reference Books:

1. Somerville, “Software Engineering”, Pearson 2.
2. Richard Fairley, “Software Engineering Concepts”, Tata McGraw Hill.
3. JalotePankaj, “An integrated approach to Software Engineering”, Narosa

Online Learning Resources:

<https://nptel.ac.in/courses/106/105/106105182/>

<http://peterindia.net/SoftwareDevelopment.html>



KRISHNA UNIVERSITY
COLLEGE OF ENGINEERING & TECHNOLOGY
MACHILIPATNAM – 521 004 (A.P) INDIA

R 20 Regulations

Computer Science & Engineering

Course Code	Database Management Systems Laboratory	L	T	P	C
		0	0	3	1.5
Pre-requisite		Semester		IV	

Course Objectives:

- ❖ To implement the basic knowledge of SQL queries and relational algebra.
- ❖ To construct database models for different database applications.
- ❖ To apply normalization techniques for refining of databases.
- ❖ To practice various triggers, procedures, and cursors using PL/SQL.
- ❖ To design and implementation of a database for an organization

Course Outcomes (CO):

After completion of the course, students will be able to

- ❖ Design database for any real world problem
- ❖ Implement PL/SQL programs
- ❖ Define SQL queries
- ❖ Decide the constraints
- ❖ Investigate for data inconsistency

List of Experiments:

Week-1: CREATION OF TABLES

1. Create a table called Employee with the following structure.

Name	Type
Empno	Number
Ename	Varchar2(20)
Job	Varchar2(20)
Mgr	Number
Sal	Number

- a. Add a column commission with domain to the Employee table.
- b. Insert any five records into the table.
- c. Update the column details of job
- d. Rename the column of Employ table using alter command.
- e. Delete the employee whose empno is 19.

2. Create department table with the following structure.

Name	Type
Deptno	Number
Deptname	Varchar2(20)
location	Varchar2(20)

- a. Add column designation to the department table.
- b. Insert values into the table.
- c. List the records of emp table grouped by deptno.
- d. Update the record where deptno is 9.
- e. Delete any column data from the table

3. Create a table called Customertable



KRISHNA UNIVERSITY
COLLEGE OF ENGINEERING & TECHNOLOGY
MACHILIPATNAM – 521 004 (A.P) INDIA

R 20 Regulations

Computer Science & Engineering

Name	Type
Cust name	Varchar2(20)
Cust street	Varchar2(20)
Cust city	Varchar2(20)

- a. Insert records into the table.
- b. Add salary column to the table.
- c. Alter the table column domain.
- d. Drop salary column of the customer table.
- e. Delete the rows of customer table whose cust_city is 'hyd'.
- f. Create a table called branch table.

Name	Type
Branch name	Varchar2(20)
Branch city	Varchar2(20)
asserts	Number

4. Increase the size of data type for asserts to the branch.
 - a. Add and drop a column to the branch table.
 - b. Insert values to the table.
 - c. Update the branch name column
 - d. Delete any two columns from the table
5. Create a table called sailor table

Name	Type
sid	Number
Sname	Varchar2(20)
rating	Varchar2(20)

- a. Add column age to the sailor table.
 - b. Insert values into the sailor table.
 - c. Delete the row with rating > 8.
 - d. Update the column details of sailor.
 - e. Insert null values into the table.
6. Create a table called reserves table

Name	Type
Boat id	Integer
sid	Integer
day	Integer

- a. Insert values into the reserves table.
- b. Add column time to the reserves table.
- c. Alter the column day data type to date.
- d. Drop the column time in the table.
- e. Delete the row of the table with some condition.

Week-2: QUERIES USING DDL AND DML

1. a. Create a user and grant all permissions to the user.
b. Insert the any three records in the employee table and use rollback. Check the result.



Computer Science & Engineering

- c. Add primary key constraint and not null constraint to the employeetable.
- d. Insert null values to the employee table and verify theresult.
2. a. Create a user and grant all permissions to theuser.
b. Insert values in the department table and usecommit.
c. Add constraints like unique and not null to the departmenttable.
d. Insert repeated values and null values into thetable.
3. a. Create a user and grant all permissions to theuser.
b. Insert values into the table and use commit.
c. Delete any three records in the department table and use rollback.
d. Add constraint primary key and foreign key to thetable.
4. a. Create a user and grant all permissions to theuser.
b. Insert records in the sailor table and usecommit.
c. Add save point after insertion of records and verify save point.
d. Add constraints not null and primary key to the sailortable.
5. a. Create a user and grant all permissions to theuser.
b. Use revoke command to remove userpermissions.
c. Change password of the usercreated.
d. Add constraint foreign key and notnull.
6. a. Create a user and grant all permissions to theuser.
b. Update the table reserves and use savepointandrollback.
c. Add constraint primary key , foreign key and not null to the reserves table
d. Delete constraint not null to the tablecolumn

Week-3:QUERIES USING AGGREGATE FUNCTIONS

1. a. By using the group by clause, display the enames who belongs to deptno 10 alongwithaveragesalary.
b. Display lowest paid employee details under eachdepartment.
c. Display number of employees working in each department and their departmentnumber.
d. Using built in functions, display number of employees working in each department and their department name from dept table. Insert deptname to dept table and insert deptname for each row, do the required thing specified above.
e. List all employees which start with either B or C.
f. Display only these ename of employees where the maximum salary is greater than or equalto 5000.
2. a. Calculate the average salary for each differentjob.
b. Show the average salary of each job excludingmanager.
c. Show the average salary for all departments employing more than threepople.
d. Display employees who earn more than thelowest salary in department 30
e. Show that value returned by sign (n)function.
f. How many days between day of birth to currentdate
3. a. Show that two substring as singlestring.
b. List all employee names, salary and 15% rise insalary.
c. Display lowest paid emp details under eachmanager
d. Display the average monthly salary bill for eachdeptno.
e. Show the average salary for all departments employing more than twopeople.
f. By using the group by clause, display the eid who belongs to deptno 05 along withaverage salary.
4. a. Count the number of employees in department20
b. Find the minimum salary earned byclerk.
c. Find minimum, maximum, average salary of allemployees.
d. List the minimum and maximum salaries for each jobtype.
e. List the employee names in descendingorder.



Computer Science & Engineering

- f. List the employee id, names in ascending order by empid.
5. a. Find the sids ,names of sailors who have reserved all boats called“INTERLAKE
Find the age of youngest sailor who is eligible to vote for each rating level with at least two such sailors.
 - b. Find the sname , bid and reservation date for each reservation.
 - c. Find the ages of sailors whose name begin and end with B and has at least 3 characters.
 - d. List in alphabetic order all sailors who have reserved redboat.
 - e. Find the age of youngest sailor for each rating level.
6. a. List the Vendors who have delivered products within 6 months from orderdate.
 - b. Display the Vendor details who have supplied both Assembled and Subparts.
 - c. Display the Sub parts by grouping the Vendor type (Local or NonLocal).
 - d. Display the Vendor details in ascending order.
 - e. Display the Sub part which costs more than any of the Assembled parts.
 - f. Display the second maximum cost Assembled part

Week-4: PROGRAMS ON PL/SQL

1. a. Write a PL/SQL program to swaptwonumbers.
 - b. Write a PL/SQL program to find the largest of threenumbers.
2. a. Write a PL/SQL program to find the total and average of 6 subjects and display the grade.
 - b. Write a PL/SQL program to find the sum of digits in a given number.
3. a. Write a PL/SQL program to display the number in reverse order.
 - b. Write a PL/SQL program to check whether the given number is prime or not.
4. a. Write a PL/SQL program to find the factorial of a given number.
 - b. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns radius and area.
5. a. Write a PL/SQL program to accept a string and remove the vowels from the string. (When 'hello' passed to the program it should display 'Hll' removing e and o from the worldHello).
 - b. Write a PL/SQL program to accept a number and a divisor. Make sure the divisor is less than or equal to 10. Else display an error message. Otherwise Display the remainder in words.

Week-5: PROCEDURES AND FUNCTIONS

1. Write a function to accept employee number as parameter and return Basic +HRA together as single column.
2. Accept year as parameter and write a Function to return the total net salary spent for a given year.
3. Create a function to find the factorial of a given number and hence find NCR.
4. Write a PL/SQL block to print prime Fibonacci series using local functions.
5. Create a procedure to find the lucky number of a given birthdate.
6. Create function to the reverse of given number

Week-6: TRIGGERS

1. Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values:



Computer Science & Engineering

CUSTOMERS table:

ID	NAME	AGE	ADDRESS	SALARY
1	Alive	24	Khammam	2000
2	Bob	27	Kadappa	3000
3	Catri	25	Guntur	4000
4	Dena	28	Hyderabad	5000
5	Eeshwar	27	Kurnool	6000
6	Farooq	28	Nellore	7000

2. Creation of insert trigger, delete trigger, update trigger practice triggers using the passenger database.
Passenger(Passport_ id INTEGER PRIMARY KEY, Name VARCHAR (50) NotNULL, Age Integer Not NULL, Sex Char, Address VARCHAR (50) NotNULL);
 - a. Write a Insert Trigger to check the Passport_id is exactly six digits ornot.
 - b. Write a trigger on passenger to display messages '1 Record is inserted', '1 record is deleted', '1 record is updated' when insertion, deletion and updation are done on passengerrespectively.
3. Insert row in employee table using Triggers. Every trigger is created with name any trigger have same name must be replaced by new name. These triggers can raised before insert, update or delete rows on data base. The main difference between a trigger and a stored procedure is that the former is attached to a table and is only fired when an INSERT, UPDATE or DELETEoccurs.
4. Convert employee name into uppercase whenever an employee record is inserted or updated. Trigger to fire before the insert orupdate.
5. Trigger before deleting a record from emp table. Trigger will insert the row to be deleted into table called delete _emp and also record user who has deleted the record and date and time ofdelete.
6. Create a transparent audit system for a table CUST_MSTR. The system must keep track of the records that are being deleted orupdated

Week-7:PROCEDURES

1. Create the procedure for palindrome of givennumber.
2. Create the procedure for GCD: Program should load two registers with two Numbers and then apply the logic for GCD of two numbers. GCD of two numbers is performed by dividing the greater number by the smaller number till the remainder is zero. If it is zero, the divisor is the GCD if not the remainder and the divisors of the previous division are the new set of two numbers. The process is repeated by dividing greater of the two numbers by the smaller number till the remainder is zero and GCD isfound.
3. Write the PL/SQL programs to create the procedure for factorial of givennumber.
4. Write the PL/SQL programs to create the procedure to find sum of N naturalnumber.
5. Write the PL/SQL programs to create the procedure to find Fibonacciserries.
6. Write the PL/SQL programs to create the procedure to check the given number is perfect ornot

Week-8: CURSORS

1. Write a PL/SQL block that will display the name, dept no, salary of fist highest paidemployees.
2. Update the balance stock in the item master table each time a transaction takes place in the item transaction table. The change in item master table depends on the item id is already present in the item master then update operation is performed to decrease the balance stock by the quantity specified in the item transaction in case the item id is not present in the item master table then the record is inserted in the item mastertable.
3. Write a PL/SQL block that will display the employee details along with salary usingcursors.
4. To write a Cursor to display the list of employees who are working as a ManagersorAnalyst.
5. To write a Cursor to find employee with given job anddeptno.
6. Write a PL/SQL block using implicit cursor that will display message, the salaries of all the



Computer Science & Engineering

employees in the 'employee' table are updated. If none of the employee's salary are updated we get a message 'None of the salaries were updated'. Else we get a message like for example, 'Salaries for 1000 employees are updated' if there are 1000 rows in 'employee' table

Week-9: CASE STUDY: BOOK PUBLISHING COMPANY

A publishing company produces scientific books on various subjects. The books are written by authors who specialize in one particular subject. The company employs editors who, not necessarily being specialists in a particular area, each take sole responsibility for editing one or more publications.

A publication covers essentially one of the specialist subjects and is normally written by a single author. When writing a particular book, each author works with one editor, but may submit another work for publication to be supervised by other editors. To improve their competitiveness, the company tries to employ a variety of authors, more than one author being a specialist in a particular subject for the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams

Week-10: CASE STUDY GENERAL HOSPITAL

A General Hospital consists of a number of specialized wards (such as Maternity, Pediatric, Oncology, etc.). Each ward hosts a number of patients, who were admitted on the recommendation of their own GP and confirmed by a consultant employed by the Hospital. On admission, the personal details of every patient are recorded. A separate register is to be held to store the information of the tests undertaken and the results of a prescribed treatment. A number of tests may be conducted for each patient. Each patient is assigned to one leading consultant but may be examined by another doctor, if required. Doctors are specialists in some branch of medicine and may be leading consultants for a number of patients, not necessarily from the same ward. For the above case study, do the following.

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams **Week-**

11: CASE STUDY: CAR RENTAL COMPANY

A database is to be designed for a car rental company. The information required includes a description of cars, subcontractors (i.e. garages), company expenditures, company revenues and customers. Cars are to be described by such data as: make, model, year of production, engine size, fuel type, number of passengers, registration number, purchase price, purchase date, rent price and insurance details. It is the company policy not to keep any car for a period exceeding one year. All major repairs and maintenance are done by subcontractors (i.e. franchised garages), with whom CRC has long-term agreements. Therefore the data about garages to be kept in the database includes garage names, addresses, range of services and the like. Some garages require payments immediately after a repair has been made; with others CRC has made arrangements for credit facilities. Company expenditures are to be registered for all outgoings connected with purchases, repairs, maintenance, insurance etc. Similarly the cash inflow coming from all sources: Car hire, car sales, insurance claims must be kept of file. CRC maintains a reasonably stable client base. For this privileged category of customers special credit card facilities are provided. These customers may also book in advance a particular car. These reservations can be made for any period of time up to one month. Casual customers must pay a deposit for an estimated time of



Computer Science & Engineering

rental, unless they wish to pay by credit card. All major credit cards are accepted. Personal details such as name, address, telephone number, driving license, number about each customer are kept in the database. For the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams

Week-12: CASE STUDY: STUDENT PROGRESS MONITORING SYSTEM

A database is to be designed for a college to monitor students' progress throughout their course of study. The students are reading for a degree (such as BA, BA (Hons.) M.Sc., etc) within the framework of the modular system. The college provides a number of modules, each being characterized by its code, title, credit value, module leader, teaching staff and the department they come from. A module is coordinated by a module leader who shares teaching duties with one or more lecturers. A lecturer may teach (and be a module leader for) more than one module. Students are free to choose any module they wish but the following rules must be observed: Some modules require pre-requisites modules and some degree programs have compulsory modules. The database is also to contain some information about students including their numbers, names, addresses, degrees they are read for, and their past performance.

i.e. modules taken and examination results. For the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.
3. Create the logical data model i.e., ER diagrams.
4. Comprehend the data given in the case study by creating respective tables with primary keys and foreign keys wherever required.
5. Insert values into the tables created (Be vigilant about Master- Slave tables).
6. Display the Students who have taken M.Sc. course
7. Display the Module code and Number of Modules taught by each Lecturer.
8. Retrieve the Lecturer names who are not Module Leaders.
9. Display the Department name which offers 'English' module.
10. Retrieve the Prerequisite Courses offered by every Department (with Department names).
11. Present the Lecturer ID and Name who teaches 'Mathematics'.
12. Discover the number of years a Module is taught.
13. List out all the Faculties who work for 'Statistics' Department.
14. List out the number of Modules taught by each Module Leader.
15. List out the number of Modules taught by a particular Lecturer.
16. Create a view which contains the fields of both Department and Module tables.
(Hint- The fields like Module code, title, credit, Department code and its name).
17. Update the credits of all the prerequisite courses to 5. Delete the Module 'History' from the Module table.

References:

1. Ramez Elmasri, Shamkant, B. Navathe, "Database Systems", Pearson Education, 6th Edition, 2013.
2. Peter Rob, Carles Coronel, "Database System Concepts", Cengage Learning, 7th Edition, 2008.

Online Learning Resources/Virtual Labs:

<http://www.scoopworld.in>

<http://vlabs.iitb.ac.in/vlabs-dev/labs/dblab/index.php>

Note: The list of experiments not be restricted to the above list. Detailed list of programming exercises can be prepared by the concerned faculty members.



Computer Science & Engineering

Course Code	OPERATING SYSTEMS LAB		L	T	P	C
			0	0	3	1.5
Pre-requisite	Basics of CO and DBMS	Semester	IV			

Course Objectives:

- ❖ To familiarize students with the architecture of OS.
- ❖ To provide necessary skills for developing and debugging CPU Scheduling algorithms.
- ❖ To elucidate the process management and scheduling and memory management.
- ❖ To explain the working of an OS as a resource manager, file system manager, process manager, memory manager, and page replacement tool.
- ❖ To provide insights into system calls, file systems and deadlock handling.

Course Outcomes (CO):

After completion of the course, students will be able to

- ❖ Trace different CPU Scheduling algorithms (L2).
- ❖ Implement Bankers Algorithms to Avoid and prevent the Dead Lock (L3).
- ❖ Evaluate Page replacement algorithms (L5).
- ❖ Illustrate the file organization techniques (L4).
- ❖ Illustrate shared memory process (L4).
- ❖ Design new scheduling algorithms (L6)

List of Experiments:

1. Practicing of Basic UNIX Commands.
2. Write programs using the following UNIX operating system calls
Fork, exec, getpid, exit, wait, close, stat, opendir and readdir
3. Simulate UNIX commands like cp, ls, grep, etc.,
4. Simulate the following CPU scheduling algorithms
a) Round Robin b) SJF c) FCFS d) Priority
5. Implement a dynamic priority scheduling algorithm.
6. Assume that there are five jobs with different weights ranging from 1 to 5. Implement round robin algorithm with time slice equivalent to weight.
7. Implement priority scheduling algorithm. While executing, no process should wait for more than 10 seconds. If the waiting time is more than 10 seconds that process has to be executed for at least 1 second before waiting again.
8. Control the number of ports opened by the operating system with
a) Semaphore b) Monitors.
9. Simulate how parent and child processes use shared memory and address space.
10. Simulate sleeping barber problem.
11. Simulate dining philosopher's problem.
12. Simulate producer-consumer problem using threads.
13. Implement the following memory allocation methods for fixed partition
a) First fit b) Worst fit c) Best fit
14. Simulate the following page replacement algorithms
a) FIFO b) LRU c) LFU etc.,
15. Simulate Paging Technique of memory management
16. Simulate Bankers Algorithm for Dead Lock avoidance and prevention
17. Simulate the following file allocation strategies
a) Sequential b) Indexed c) Linked
18. Simulate all File Organization Techniques
a) Single level directory b) Two level c) Hierarchical d) DAG



KRISHNA UNIVERSITY
COLLEGE OF ENGINEERING & TECHNOLOGY
MACHILIPATNAM – 521 004 (A.P) INDIA

R 20 Regulations

References:

1. “Operating System Concepts”, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, Eighth Edition, John Wiley.
2. “Operating Systems: Internals and Design Principles”, Stallings, Sixth Edition–2009, Pearson Education
3. “Modern Operating Systems”, Andrew S Tanenbaum, Second Edition, PHI.
4. “Operating Systems”, S.Haldar, A.A.Aravind, Pearson Education.
5. “Principles of Operating Systems”, B.L.Stuart, Cengage learning, India Edition.2013-2014
6. “Operating Systems”, A.S.Godbole, Second Edition, TMH.
7. “An Introduction to Operating Systems”, P.C.P. Bhatt, PHI.

Online Learning Resources/Virtual Labs:

<https://www.cse.iitb.ac.in/~mythili/os/>

<http://peterindia.net/OperatingSystems.html>

Note: The list of experiments not be restricted to the above list. Detailed list of programming exercises can be prepared by the concerned faculty members.



Computer Science & Engineering

Course Code	SOFTWARE ENGINEERING LAB	L	T	P	C
		0	0	3	1.5
Pre-requisite	Semester	IV			

Course Objectives:

- γ To learn and implement the fundamental concepts of Software Engineering.
- γ To explore functional and non-functional requirements through SRS.
- γ To practice the various design diagrams through the appropriate tool.
- γ To learn to implement various software testing strategies.

Course Outcomes (CO):

After completion of the course, students will be able to

- γ Acquaint with historical and modern software methodologies
- γ Understand the phases of software projects and practice the activities of each phase
- γ Practice clean coding
- γ Take part in project management
- γ Adopt skills such as distributed version control, unit testing, integration testing, build management, and deployment.

List of Experiments:

- 1 Draw the Work Breakdown Structure for the system to be automated
- 2 Schedule all the activities and sub-activities Using the PERT/CPM charts
- 3 Define use cases and represent them in use-case document for all the stakeholders of the system to be automated
- 4 Identify and analyze all the possible risks and its risk mitigation plan for the system to be automated
- 5 Diagnose any risk using Ishikawa Diagram (Can be called as Fish Bone Diagram or Cause& Effect Diagram)
- 6 Define Complete Project plan for the system to be automated using Microsoft Project Tool
- 7 Define the Features, Vision, Business objectives, Business rules and stakeholders in the vision document
- 8 Define the functional and non-functional requirements of the system to be automated by using Use cases and document in SRS document
- 9 Define the following traceability matrices :
 1. Use case Vs. Features
 2. Functional requirements Vs. Usecases
- 10 Estimate the effort using the following methods for the system to be automated:
 1. Function point metric
 2. Usecase point metric
- 11 Develop a tool which can be used for quantification of all the non-functional requirements
- 12 Write C/C++/Java/Python program for classifying the various types of coupling.
- 13 Write a C/C++/Java/Python program for classifying the various types of cohesion.
- 14 Write a C/C++/Java/Python program for object oriented metrics for design proposed by Chidamber and Kremer. (Popularly called CK metrics)
- 15 Convert the DFD into appropriate architecture styles.
- 16 Draw a complete class diagram and object diagrams using Rational tools
- 17 Define the design activities along with necessary artifacts using Design Document.
- 18 Reverse Engineer any object-oriented code to an appropriate class and object diagrams.
- 19 Test a piece of code that executes a specific functionality in the code to be tested and asserts a certain behavior or state using Junit.
- 20 Test the percentage of code to be tested by unit test using any code coverage tools
- 21 Define appropriate metrics for at least 3 quality attributes for any software application of your interest.



KRISHNA UNIVERSITY
COLLEGE OF ENGINEERING & TECHNOLOGY
MACHILIPATNAM – 521 004 (A.P) INDIA

R 20 Regulations

Computer Science & Engineering

- 22 Define a complete call graph for any C/C++ code. (Note: The student may use any tool that generates call graph for source code)

References:

1. Software Engineering? A Practitioner's Approach, Roger S. Pressman, 1996, MGH.
2. Software Engineering by Ian Sommerville, Pearson Edu, 5th edition, 1999
3. An Integrated Approach to software engineering by Pankaj Jalote , 1991 Narosa

Online Learning Resources:

Virtual Labs:<http://vlabs.iitkgp.ac.in/se/>

Note: The list of experiments not be restricted to the above list. Detailed list of programming exercises can be prepared by the concerned faculty members.



Computer Science & Engineering

Course Code	Exploratory Data Analytics with R		L	T	P	C
			0	0	3	1.5
Pre-requisite	Fundamental Programming	Semester	IV			

Course Objectives:

The students will be able to learn:

- Y How to manipulate data within R and to create simple graphs and charts used in introductory statistics.
- Y The given data using different distribution functions in R.
- Y The hypothesis testing and calculate confidence intervals; perform linear regression models for data analysis.
- Y The relevance and importance of the theory in solving practical problems in the real world.

Course Outcomes (CO):

After completion of the course, students will be able to

- Y Install and use R for simple programming tasks.
- Y Extend the functionality of R by using add-on packages
- Y Extract data from files and other sources and perform various data manipulation tasks on them.
- Y Explore statistical functions in R.
- Y Use R Graphics and Tables to visualize results of various statistical operations on data.
- Y Apply the knowledge of R gained to data Analytics for real-life applications.

List of Experiments:

1: INTRODUCTION TO COMPUTING

- a. Installation of R
- b. The basics of R syntax, workspace
- c. Matrices and lists
- d. Subsetting
- e. System-defined functions; the help system
- f. Errors and warnings; coherence of the workspace

2: GETTING USED TO R: DESCRIBING DATA

- a. Viewing and manipulating Data
- b. Plotting data
- c. Reading the data from console, file (.csv) local disk and web
- d. Working with larger datasets

3: SHAPE OF DATA AND DESCRIBING RELATIONSHIPS

- a. Tables, charts and plots.
- b. Univariate data, measures of central tendency, frequency distributions, variation, and Shape.
- c. Multivariate data, relationships between a categorical and a continuous variable,
- d. Relationship between two continuous variables – covariance, correlation coefficients, comparing multiple correlations.
- e. Visualization methods – categorical and continuous variables, two categorical variables, two continuous variables.

4: PROBABILITY DISTRIBUTIONS

- a. Sampling from distributions – Binomial distribution, normal distribution
- b. tTest, zTest, Chi Square test
- c. Density functions
- d. Data Visualization using ggplot – Box plot, histograms, scatter plotter, line chart, bar chart, heat maps



Computer Science & Engineering

5: EXPLORATORY DATA ANALYSIS Demonstrate the range, summary, mean, variance, median, standard deviation, histogram, box plot, scatter plot using population dataset.

6: TESTING HYPOTHESES

- a. Null hypothesis significance testing
- b. Testing the mean of one sample
- c. Testing two means

7: PREDICTING CONTINUOUS VARIABLES

- a. Linear models
- b. Simple linear regression
- c. Multiple regression
- d. Bias-variance trade-off – cross-validation

8: CORRELATION

- a. How to calculate the correlation between two variables.
- b. How to make scatter plots.
- c. Use the scatter plot to investigate the relationship between two variables

9: TESTS OF HYPOTHESES

- a. Perform tests of hypotheses about the mean when the variance is known.
- b. Compute the p-value.
- c. Explore the connection between the critical region, the test statistic, and the p-value

10: ESTIMATING A LINEAR RELATIONSHIP Demonstration on a Statistical Model for a Linear Relationship

- a. Least Squares Estimates
- b. The R Function lm
- c. Scrutinizing the Residuals

11: APPLY-TYPE FUNCTIONS

- a. Defining user defined classes and operations, Models and methods in R
- b. Customizing the user's environment
- c. Conditional statements
- d. Loops and iterations

12: STATISTICAL FUNCTIONS IN R

- a. Write Demonstrate Statistical functions in R
- b. Statistical inference, contingency tables, chi-square goodness of fit, regression, generalized linear models, advanced modeling methods.

References:

1. SandipRakshit, “Statistics with R Programming”, McGraw Hill Education, 2018.
2. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, “AN Introduction to Statistical Learning: with Applications in R”, Springer Texts in Statistics, 2017.
3. Joseph Schmuller, “Statistical Analysis with R for Dummies”, Wiley, 2017.
4. K G Srinivasa, G M Siddesh, ChetanShetty, Sowmya B J, “Statistical Programming in R”, Oxford Higher Education, 2017.

Online Learning Resources/Virtual Labs:

1. www.oikostat.ch
2. <https://learningstatisticswithr.com/>
3. <https://www.coursera.org/learn/probability-intro#syllabus>
4. <https://www.isibang.ac.in/~athreya/psweur/>

Note: The list of experiments not be restricted to the above list. Detailed list of programming exercises can be prepared by the concerned faculty members.