



# T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

## B.TECH. INFORMATION TECHNOLOGY

### COURSE STRUCTURE & SYLLABUS

#### III SEMESTER

S. No	Course Code	Course Title	L	T	P	Credits
1	A63HS1	Mathematics – IV	4	1	0	4
2	A63PC2	Data Structures through C++	4	0	0	4
3	A63PC3	Mathematical Foundations of Computer Science	4	0	0	4
4	A63PC4	Digital Logic Design	3	0	0	3
5	A63PC5	Object Oriented Programming through Java	3	0	0	3
6	A63PC6	Data Structures through C++ Lab	0	0	3	2
7	A63PC7	IT Workshop	0	0	3	2
8	A63PC8	Java Programming Lab	0	0	3	2
9	A63MC3	*Environmental Science and Technology	3	0	0	0
		<b>Total Credits</b>	<b>21</b>	<b>1</b>	<b>9</b>	<b>24</b>

#### IV SEMESTER

S. No	Course Code	Course Title	L	T	P	Credits
1	A64PC1	Computer Organization	4	0	0	4
2	A64PC2	Database Management Systems	4	0	0	4
3	A64PC3	Operating Systems	4	0	0	4
4	A64PC4	Formal Languages and Automata Theory	3	0	0	3
5	A64HS5	Business Economics and Financial Analysis	3	0	0	3
6	A64PC6	Computer Organization Lab	0	0	3	2
7	A64PC7	Database Management Systems Lab	0	0	3	2
8	A64PC8	Operating Systems Lab	0	0	3	2
9	A64MC4	*Gender Sensitization Lab	0	0	3	0
		<b>Total Credits</b>	<b>18</b>	<b>0</b>	<b>12</b>	<b>24</b>

\* Satisfactory/Unsatisfactory



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech III Semester**

**L/T/P/C  
4/1/0/4**

**MATHEMATICS- IV (A63HS1)**

**Course Objectives:**

To learn

1. Differentiation and integration of complex valued functions
2. Evaluation of integrals using Cauchy's integral formula and residue theorem.
3. Laurent's series expansion of complex functions
4. Express a periodic function by Fourier series
5. Express a non-periodic function by Fourier transform

**Course Outcomes:**

After learning the contents of this paper the student must be able to

1. Analyse the complex functions with reference to their analyticity, integration using Cauchy's integral theorem
2. Find the Taylor's and Laurent's series expansion of complex functions
3. The bilinear transformation
4. Finding any periodic function in term of sine's and cosines
5. Finding a non-periodic function as integral representation

**UNIT-I**

**Functions of a complex variable:**

Introduction, Continuity, Differentiability, Analyticity, properties, Cauchy, Riemann equations in Cartesian and polar coordinates. Harmonic and conjugate harmonic functions-Milne-Thompson method.

**Bilinear transformation-** fixed point-cross ratio-properties-invariance of circles

**UNIT-II**

**Complex integration:**

Line integral, Cauchy's integral theorem, Cauchy's integral formula, and Generalized Cauchy's integral formula, Power series: Taylor's series- Laurent series

**UNIT-III**

**Evaluation of integrals:**

Singular points, isolated singular points, pole of order m – essential singularity, Residue, Cauchy Residue theorem (Without proof).

Types of real integrals:

- a) Improper real integrals  $\int_{-\infty}^{\infty} f(x)dx$  b)  $\int_c^{c+2\pi} f(\cos\theta, \sin\theta)dx$

**UNIT-IV**

**Fourier series:**

Introduction, Periodic functions, Fourier series of periodic function, Dirichlet's conditions, Even and odd functions, Change of interval, Half range sine and cosine series.

## **UNIT-V**

### **Fourier transforms**

Fourier integral theorem (without proof), Fourier sine and cosine integrals, sine and cosine, transforms, properties, inverse transforms, Finite Fourier transforms

### **Applications Fourier Transforms: Heat Equation and Wave Equation**

#### **Text Books:**

1. A first course in Complex Analysis with applications by Dennis G. Zill and Patrick Shanahan, Johns and Bartlett Publishers.
2. Higher Engineering Mathematics by Dr. B. S. Grewal, Khanna Publishers.
3. Advanced Engineering Mathematics with MATLAB by Dean G. Duffy

#### **Reference Books:**

1. Fundamentals of Complex Analysis by Saff, E. B. and A. D. Snider, Pearson.
2. Advanced Engineering Mathematics by Louis C. Barrett, McGraw Hill.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech III Semester**

**L/T/P/C  
4/0/0/4**

**DATA STRUCTURES THROUGH C++ (A63PC2)**

**Course Objectives:**

- To understand the basic concepts such as Abstract Data Types, Linear and Non Linear Data structures.
- To understand the notations used to analyze the Performance of algorithms.
- To understand the behavior of data structures such as stacks, queues, trees, hash tables, search trees, Graphs and their representations.
- To choose an appropriate data structure for a specified application.
- To understand and analyze various searching and sorting algorithms.
- To learn to implement ADTs such as lists, stacks, queues, trees, graphs, search trees in C++ to solve problems.

**Course Outcomes:**

- Understand the concept of ADT.
- Ability to choose appropriate data structures to represent data items in real world problems.
- Ability to analyse the time and space complexities of algorithms.
- Ability to design programs using a variety of data structures such as stacks, queues, hash tables, binary trees, search trees, heaps, graphs, and B-trees.
- Able to analyse and implement various kinds of searching and sorting techniques.

**UNIT - I**

**C++ Programming Concepts**

Review of C, input and output in C++, functions in C++-value parameters, reference parameters, Parameter passing, function overloading, function templates, Exceptions-throwing an exception and handling an exception, arrays, pointers, new and delete operators, class and object, access specifiers , friend functions, constructors and destructor, Operator overloading, class templates, Inheritance and Polymorphism.

**Basic Concepts**

Data objects and Structures, Algorithm Specification-Introduction, Recursive algorithms, Data abstraction, Performance analysis- time complexity and space complexity, Asymptotic Notation-Big O, Omega and Theta notations, Complexity Analysis Examples, Introduction to Linear and Non Linear data structures.

## **UNIT - II**

Representation of single, two dimensional arrays, sparse matrices-array and linked representations. Linear list ADT-array representation and linked representation, Singly Linked Lists-Operations-Insertion, Deletion, Circularly linked lists-Operations for Circularly linked lists, Doubly Linked Lists- Operations- Insertion, Deletion. Stack ADT, definition, array and linked list implementations, applications-infix to postfix conversion, Postfix expression evaluation, recursion implementation, Queue ADT, definition, array and linked Implementations, Circular queues-Insertion and deletion operations, Polynomial.

## **UNIT – III**

### **Trees**

Definition, terminology, Binary trees-definition, Properties of Binary Trees, Binary Tree ADT, representation of Binary Trees-array and linked representations, Binary Tree traversals, threaded binary trees, Priority Queues –Definition and applications, Max Priority Queue ADT-implementation-Max Heap-Definition, Insertion into a Max Heap, Deletion from a Max Heap. Disjoint set ADT - Equivalence relations, the dynamic equivalence problem, Basic data structure, Smart union algorithms, Path compression, and worst case for union by rank and path compression, and an application - generation of mazes.

## **UNIT - IV**

### **Searching**

Linear Search, Binary Search, Hashing-Introduction, hash tables, hash functions, Overflow Handling, Comparison of Searching methods.

### **Sorting**

Insertion Sort, Selection Sort, Radix Sort, Quick sort, Heap Sort, Merge sort, Comparison of Sorting methods.

### **External sorting**

Model for external sorting, basic external sorting algorithm, multi-way merge, poly-phase merge, replacement selection.

## **UNIT - V**

### **Graphs**

Definitions, Terminology, Applications and more definitions, Properties, Graph ADT, Graph Representations- Adjacency matrix, Adjacency lists, Graph Search methods - DFS and BFS, Complexity analysis.

### **Search Trees**

Binary Search Tree ADT, Definition, Operations- Searching, Insertion and Deletion, Balanced search trees-AVL Trees-Definition and Examples only, B-Trees-Definition and Examples only, Red-Black Trees-Definitions and Examples only, k-d trees, Comparison of Search Trees.

### **TEXT BOOKS:**

1. Data structures, Algorithms and Applications in C++, 2<sup>nd</sup> Edition, Sartaj Sahni, Universities Press.
2. Data structures and Algorithms in C++, Adam Drozdek, 4<sup>th</sup> edition, Cengage learning.

### **REFERENCE BOOKS:**

1. Data structures with C++, J. Hubbard, Schaum's outlines, TMH.
2. Data structures and Algorithms in C++, M.T. Goodrich, R. Tamassia and D. Mount, Wiley India.
3. Data structures and Algorithm Analysis in C++, 3<sup>rd</sup> edition, M. A. Weiss, Pearson.
4. Classic Data Structures, D. Samanta, 2<sup>nd</sup> edition, PHI.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech III Semester**

**L/T/P/C  
4/0/0/4**

**MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE (A63PC3)**

**Course Objectives:**

To learn

1. To relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context.
2. To describe the importance and limitations of predicate logic.
3. To describe the importance of Permutations and combinations in real life.
4. To relate the ideas of mathematical induction to recursion and recursively defined structures.
5. To use Graph Theory for solving problems.

**Course Outcomes:**

After learning the contents of this paper the student must be able to

1. Ability to illustrate by examples the basic terminology of functions, relations, and sets and demonstrate knowledge of their associated operations.
2. Ability to demonstrate in practical applications the use of basic counting principles of permutations, combinations, inclusion/exclusion principle and the pigeonhole methodology.
3. Ability to represent and Apply Graph theory in solving computer science problems and ability to write programs with graph theory.
4. Ability to solve problems using Predicate and Mathematical logic.
5. Ability to solve the recurrence relations.

**UNIT-I:**

**Mathematical Logic:**

Statements and notations, Connectives, well-formed formulas, truth tables, tautology, equivalence implication normal forms, quantifiers, universal quantifiers.

Predicates: Predicative logic, free and bounded variables, rules of inference, consistency, proof of contradiction.

**UNIT-II:**

**Elementary combinatorics:**

Basics of counting, combinations and permutations, with repetitions, constrained repetitions, binomial coefficients, binomial multi nominal theorem, Pigeon hole principles and its application.

**Recurrence Relation**

Generating functions, Recurrence relations, solving recurrence relations by substitution and generating function, characteristic roots, solution of homogeneous and non-homogeneous recurrence relation

### **UNIT-III:**

#### **Relations**

Properties of Binary relations, Equivalence, transitive closure, compatibility and partial ordering relations, lattices & properties, Hasse diagram

#### **Functions**

Inverse function, composition of functions, recursive functions.

### **UNIT-IV:**

#### **Groups**

Algebraic structures, examples and general properties, Semi groups and monoids, Groups and sub groups, cosets and Lagranges theorem, normal subgroups, homomorphism, isomorphism of groups, permutation groups, cyclic groups

### **UNIT-V:**

#### **Graph Theory**

Representation of graph, planar graphs. Graph theory and applications, basic concepts- isomorphism and sub graphs, multi graphs and Euler's circuits, Hamiltonian graphs, chromatic number.

#### **Text Books:**

1. Discrete Mathematics for Computer scientists & Mathematicians, J. L. Mott, A. Kandel, T.P.Baker.
2. Discrete mathematics and its Applications, Kenneth H.Rosen, fifth edition. TMH

#### **Reference Books:**

1. Elements of Discrete mathematics, C.L.Liu, D.P.Mohapatra, 4<sup>th</sup> edition, McGraw Hill education (India) Private Limited
2. Discrete mathematical structures theory and applications- malik & Sen Cengage
3. Discrete mathematics with applications, Thomas Koshy, Elsevier
4. Logic and Discrete mathematics, grass Man & Trembley, Pearson Education



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech III Semester**

**L/T/P/C  
3/0/0/3**

**DIGITAL LOGIC DESIGN (A63PC4)**

**Course Objectives**

- To understand basic number systems, codes and logical gates.
- To understand the concepts of Boolean algebra.
- To understand the use of minimization logic to solve the Boolean logic expressions.
- To understand the design of combinational and sequential circuits.
- To understand the state reduction methods for Sequential circuits.
- To understand the basics of various types of memories.

**Course Outcomes**

- Able to understand number systems and codes.
- Able to solve Boolean expressions using Minimization methods.
- Able to design the sequential and combinational circuits.
- Able to apply state reduction methods to solve sequential circuits.
- Understand about memory, RAM and ROM.

**UNIT - I**

Digital Systems, Binary Numbers, Number base conversions, Octal, Hexadecimal and other base numbers, Excess-3 code, Gray code, complements, signed binary numbers, binary codes, binary storage and registers, binary logic, Boolean algebra and logic gates, Basic theorems and properties of Boolean Algebra, Boolean functions, canonical and standard forms, Digital Logic Gates, Implementation of basic gates using universal gates.

**UNIT - II**

**Gate -Level Minimization**

The K-Map Method, Three-Variable Map, Four-Variable Map, Five-Variable Map , sum of products , product of sums simplification, Don't care conditions, NAND and NOR implementation and other two level implementations, Exclusive-OR function.

**UNIT - III**

**Combinational Circuits (CC)**

Analysis procedure, Design Procedure, Combinational circuit for different code converters and other problems, Binary Adder-Subtractor, Decimal Adder, Binary Multiplier, Magnitude Comparator, Decoders, Encoders, Multiplexers, De-multiplexers.

**UNIT - IV**

**Flip-Flops**

Basic Latch, SR and D latches, Master Slave edge triggered D Flip-flop, T Flip-Flop, and JK Flip Flops, Analysis of clocked sequential circuits. State Reduction and assignment, Flip-Flop Excitation tables, Design procedure. Registers, Shift registers, Ripple counters, Synchronous counters, other counters. Asynchronous Sequential Circuits -Introduction, Analysis procedure, Circuits with latches, Design procedure, Reduction of state and follow tables, Race- free state assignment, Hazards.

## **UNIT - V**

### **Memory**

Introduction, Random-Access memory, Memory decoding, ROM, Programmable Logic Array, Programmable Array Logic, Sequential programmable devices.

### **Register Transfer and Micro-operations**

Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro-operations, Logic Micro-operations, Shift Micro-operations, Arithmetic Logic Shift Unit.

### **TEXT BOOKS**

1. Digital Design, M. Morris Mano, M.D.Ciletti, 5th edition, Pearson.(Units I, II, III, IV, Part of Unit V)
2. Computer System Architecture, M.Morris Mano, 3rd edition, Pearson.(Part of Unit V)

### **REFERENCE BOOKS**

1. Switching and Finite Automata Theory, Z. Kohavi, Tata McGraw Hill.
2. Fundamentals of Logic Design, C. H. Roth, L. L. Kinney, 7<sup>th</sup> edition, Cengage Learning.
3. Fundamentals of Digital Logic & Micro Computer Design, 5TH Edition, M. Rafiquzzaman, John Wiley.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech III Semester**

**L/T/P/C  
3/0/0/3**

**OBJECT ORIENTED PROGRAMMING THROUGH JAVA (A63PC5)**

**Course Objectives:**

- To introduce the object oriented programming concepts.
- To understand object oriented programming concepts, and apply them in solving problems.
- To introduce the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes
- To introduce the implementation of packages and interfaces
- To introduce the concepts of exception handling and multithreading.
- To introduce the design of Graphical User Interface using applets and swing controls.

**Course Outcomes**

- Able to understand the use of abstract classes.
- Able to solve problems using java collection framework and I/O classes.
- Able to develop multithreaded applications with synchronization.
- Able to develop applets for web applications.
- Able to design GUI based applications.
- Write Java programs to solve real world problems.

**UNIT - I**

**Object-oriented thinking**

A way of viewing world – Agents and Communities, messages and methods, Responsibilities, Classes and Instances, Class Hierarchies- Inheritance, Method binding, Overriding and Exceptions, Summary of Object-Oriented concepts. History of object-oriented programming, overview of java, Object oriented design, Structure of java program, Java buzzwords, Data types, Variables and Arrays, operators, expressions, control statements, Introducing classes, Methods and Classes, String handling.

**Inheritance**

Inheritance concept, Inheritance basics, Member access, Constructors, Creating Multilevel hierarchy, super uses, using final with inheritance, Polymorphism-ad hoc polymorphism, pure polymorphism, method overriding, abstract classes, Object class, forms of inheritance-specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance.

**UNIT - II**

**Packages**

Defining a Package, CLASSPATH, Access protection, importing packages.

**Interfaces**

Defining an interface, implementing interfaces, Nested interfaces, applying interfaces, variables in interfaces and extending interfaces.

**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, auto boxing, generics.

## **UNIT - III**

### **Exception handling**

Fundamentals of exception handling, Exception types, Termination or resumptive models, Uncaught exceptions, using try and catch, multiple catch clauses, nested try statements, throw, throws and finally, built- in exceptions, creating own exception sub classes.

### **Multithreading**

Differences between thread-based multitasking and process-based multitasking, Java thread model, creating threads, thread priorities, creating multiple threads, synchronizing threads, inter thread communication, dead lock, Suspending, resuming, and stopping threads.

## **UNIT - IV**

### **The Collections Framework (java.util)**

Collections overview, Collection Interfaces, The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Accessing a Collection via an Iterator, Using an Iterator, The For-Each alternative, Map Interfaces and Classes, Comparators, Collection algorithms, Arrays, The Legacy Classes and Interfaces- Dictionary, Hash table ,Properties, Stack, Vector, More Utility classes, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter, Scanner.

## **UNIT - V**

### **GUI Programming with Swing**

Introduction, limitations of AWT, MVC architecture, components, containers. Understanding Layout Managers, Flow Layout, Border Layout, Grid Layout, Card Layout, Grid Bag Layout.

### **Event Handling**

The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes. A Simple Swing Application.

### **Applets**

Applets and HTML, Security Issues, Applets and Applications, passing parameters to applets. Creating a Swing Applet, Painting in Swing, A Paint example, Exploring Swing Controls- JLabel and Image Icon, JText Field, The Swing Buttons- JButton, JToggle Button, JCheck Box, JRadio Button, Tabbed Pane, JScroll Pane, JList, JCombo Box, Swing Menus, Dialogs.

## **TEXT BOOKS**

1. Java The complete reference, 9<sup>th</sup> edition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.

## **REFERENCE BOOKS**

1. An Introduction to programming and OO design using Java, J. Nino and F.A. Hosch, John Wiley & sons.
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
3. Object Oriented Programming through Java, P. Radha Krishna, and Universities Press.
4. Programming in Java, S. Malhotra, S. Chudhary, 2<sup>nd</sup> edition, Oxford Univ. Press.
5. Java Programming and Object oriented Application Development, R. A. Johnson, Cengage Learning.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech III Semester**

**L/T/P/C  
0/0/3/2**

**DATA STRUCTURES THROUGH C++ LAB (A63PC6)**

**Course Objectives**

- To write and execute programs in C++ to solve problems using data structures such as arrays, linked lists, stacks, queues, trees, graphs, hash tables and search trees.
- To learn to write C++ programs to implement various sorting and searching algorithms.

**Course Outcomes**

- Able to identify the appropriate data structures and algorithms for solving real world problems.
- Able to implement various kinds of searching and sorting techniques.
- Able to implement data structures such as stacks, queues, Search trees, and hash tables to solve various computing problems.
- Able to implement different disjoint set operations and k-d trees.

**List of Programmes to be performed during the Course**

1. Write a C++ program that uses functions to perform the following:
  - a) Create a singly linked list of integers.
  - b) Delete a given integer from the above linked list.
  - c) Display the contents of the above list after deletion.
2. Write a template based C++ program that uses functions to perform the following:
  - a) Create a doubly linked list of elements.
  - b) Delete a given element from the above doubly linked list.
  - c) Display the contents of the above list after deletion.
3. Write a C++ program that uses stack operations to convert a given infix expression into its postfix equivalent, Implement the stack using an array.
4. Write a C++ program to implement a double ended queue ADT using an array, using a doubly linked list.
5. Write a C++ program that uses functions to perform the following:
  - a) Create a binary search tree of characters.
  - b) Traverse the above Binary search tree recursively in preorder, in order and post order,
6. Write a C++ program that uses function templates to perform the following:
  - a) Search for a key element in a list of elements using linear search.
  - b) Search for a key element in a list of sorted elements using binary search.
7. Write a C++ program that implements Insertion sort algorithm to arrange a list of integers in ascending order.
8. Write a template based C++ program that implements selection sort algorithm to arrange a list of elements in descending order.
9. Write a template based C++ program that implements Quick sort algorithm to arrange a list of elements in ascending order.
10. Write a C++ program that implements Heap sort algorithm for sorting a list of integers in ascending order.

11. Write a C++ program that implements Merge sort algorithm for sorting a list of integers in ascending order.
12. Write a C++ program to implement all the functions of a dictionary (ADT) using hashing.
13. Write a C++ program that implements Radix sort algorithm for sorting a list of integers in ascending order.
14. Write a C++ program that uses functions to perform the following:
  - a) Create a binary search tree of integers.
  - b) Traverse the above Binary search tree non-recursively in ignored.
15. Write a C++ program that uses functions to perform the following:
  - a) Create a binary search tree of integers.
  - b) Search for an integer key in the above binary search tree non-recursively.
  - c) Search for an integer key in the above binary search tree recursively.
16. Write a C++ program to implement hashing using any hash function.
17. Write a C++ program to implement extendible hashing.

## **REFERENCE BOOKS**

1. Data Structures using C++, D. S. Malik, 2<sup>nd</sup> edition, Cengage learning.
2. Data Structures using C++, V. Patil, Oxford University Press.
3. Fundamentals of Data structures in C++, 2<sup>nd</sup> edition, E. Horowitz, S. Sahni and D. Mehta, Universities Press.
4. C++ Plus Data Structures, 4<sup>th</sup> edition, Nell Dale, Jones and Bartlett student edition.



# TKRC COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

## B.TECH. INFORMATION TECHNOLOGY

B.Tech III Semester

L/T/P/C  
0/0/3/2

### IT WORKSHOP (A63PC7)

#### Course Objectives

- The IT Workshop is a training lab course to get training on PC Hardware, Internet & World Wide Web, and Productivity tools for documentation, Spreadsheet computations, and Presentation.
- To introduce to a personal computer and its basic peripherals, the process of assembling a personal computer, installation of system software like MS Windows, Linux and the required device drivers, hardware and software level troubleshooting process.
- To introduce connecting the PC on to the internet from home and workplace and effectively usage of the internet, Usage of web browsers, email, newsgroups and discussion forums. To get knowledge in awareness of cyber hygiene, i.e., protecting the personal computer from getting infected with the viruses, worms and other cyber-attacks.
- To introduce the usage of Productivity tools in crafting professional word documents, excel spreadsheets and power point presentations using open office tools and LaTeX.

#### Course Outcomes:

- Apply knowledge for computer assembling and software installation.
- Ability how to solve the trouble shooting problems.
- Apply the tools for preparation of PPT, Documentation and budget sheet etc.
- Create standard documents and research documents using Latex.
- Able to create project plans.

#### PC Hardware

The students should work on working PC to disassemble and assemble to working condition and install operating system like Linux or any other on the same PC. Students are suggested to work similar tasks in the Laptop scenario wherever possible.

#### Problem 1

Every student should identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor. Every student should disassemble and assemble the PC back to working condition.

#### Problem 2

Every student should individually install operating system like Linux or MS windows on the personal computer. The system should be configured as dual boot with both windows and Linux.

### **Problem 3**

Hardware Troubleshooting: Students have to be given a PC which does not boot due to improper assembly or defective peripherals. They should identify the problem and fix it to get the computer back to working condition.

### **Problem 4**

Software Troubleshooting: Students have to be given a malfunctioning CPU due to system software problems. They should identify the problem and fix it to get the computer back to working condition.

## **Internet & World Wide Web.**

### **Problem 5**

Orientation & Connectivity Boot Camp: Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should demonstrate how to access the websites and email.

### **Problem 6**

Web Browsers, Surfing the Web: Students customize their web browsers with the LAN proxy settings, bookmarks, search toolbars and pop up blockers. Also, plug-ins like Macromedia Flash and JRE for applets should be configured.

### **Problem 7**

Search Engines & Netiquette: Students should know what search engines are and how to use the search engines. Usage of search engines like Google, Yahoo, ask.com and others should be demonstrated by student.

### **Problem 8**

Cyber Hygiene: Students should learn about viruses on the internet and install antivirus software. Student should learn to customize the browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

### **Problem 9**

Develop home page: Student should learn to develop his/her home page using HTML consisting of his/her photo, name, address and education details as a table and his/her skill set as a list.

## **Productivity tools**

### **LaTeX and Word**

#### **Word Orientation**

An overview of LaTeX and Microsoft (MS) office / equivalent (FOSS) tool word should be learned: Importance of LaTeX and MS office / equivalent (FOSS) tool Word as word Processors, Details of the three tasks and features that should be covered in each, using LaTeX and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter.

### **Problem 10**

#### **Using LaTeX and Word**

To create project certificate. Features to be covered:-Formatting Fonts in word, Drop Cap in word, Applying Text effects, Using Character Spacing, Borders and Colours, Inserting Header and Footer, Using Date and Time option in both LaTeX and Word.

**Problem 11****Creating project abstract Features to be covered**

Formatting Styles, Inserting table, Bullets and Numbering, Changing Text Direction, Cell alignment, Footnote, Hyperlink, Symbols, Spell Check, Track Changes.

**Problem 12****Creating a Newsletter Features to be covered**

Table of Content, Newspaper columns, Images from files and clipart, Drawing toolbar and Word Art, Formatting Images, Textboxes, Paragraphs in word.

**Problem 13****Spreadsheet Orientation**

Accessing, overview of toolbars, saving spreadsheet files, Using help and resources.

**Creating a Scheduler**

Gridlines, Format Cells, Summation, auto fill, Formatting Text

**Problem 14****Calculating GPA Features to be covered**

Cell Referencing, Formulae in spreadsheet – average, std. deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function, Sorting, Conditional formatting.

**Problem 15****Creating Power Point**

Student should work on basic power point utilities and tools in Latex and Ms Office/equivalent (FOSS) which help them create basic power point presentation. PPT Orientation, Slide Layouts, Inserting Text, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows, Hyperlinks, Inserting Images, Tables and Charts.

**REFERENCE BOOKS**

1. Introduction to Information Technology, ITL Education Solutions limited, Pearson Education.
2. LaTeX Companion – Leslie Lamport, PHI/Pearson.
3. Comdex Information Technology course tool kit Vikas Gupta, WILEY Dreamtech.
4. IT Essentials PC Hardware and Software Companion Guide Third Edition by David Anfinson and Ken Quamme. – CISCO Press, Pearson Education.
5. PC Hardware and A+ Handbook – Kate J. Chase PHI (Microsoft).



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech III Semester**

**L/T/P/C  
0/0/3/2**

**OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB (A63PC8)**

**Course Objectives**

- To write programs using abstract classes.
- To write programs for solving real world problems using java collection frame work.
- To write multithreaded programs.
- To write GUI programs using swing controls in Java.
- To introduce java compiler and eclipse platform.
- To impart hands on experience with java programming.

**Course Outcomes**

- Able to write programs for solving real world problems using java collection frame work.
- Able to write programs using abstract classes.
- Design and develop programs using objects and inheritance in Java language.
- Able to write multithreaded programs.
- Able to write GUI programs using swing controls in Java.

**Note:**

1. Use Linux and MySQL for the Lab Experiments. Though not mandatory, encourage the use of Eclipse platform.
2. The list suggests the minimum program set. Hence, the concerned staff is requested to add more problems to the list as needed.

**List of Experiments to be performed during the Course**

1. Use Eclipse or Net bean 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 a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
2. Write a Java program 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. Handle any possible exceptions like divided by zero.
3. a) Develop an applet in Java that displays a simple message.  
b) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.
4. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, 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.

5. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
6. Write a Java program for the following:
  - i) Create a doubly linked list of elements.
  - ii) Delete a given element from the above list.
  - iii) Display the contents of the list after deletion.
7. Write a Java program that simulates a traffic light. The program lets the user select one of 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 selected color. Initially, there is no message shown.
8. 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.
9. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.
10. Write a Java program 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).
11. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).
12. Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication.
13. Write a Java program to list all the files in a directory including the files present in all its subdirectories.
14. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order.
15. Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.

## REFERENCE BOOKS

1. Java for Programmers, P. J. Deitel and H. M. Deitel, 10<sup>th</sup> Edition Pearson education.
2. Thinking in Java, Bruce Eckel, Pearson Education.
3. Java Programming, D. S. Malik and P. S. Nair, Cengage Learning.
4. Core Java, Volume 1, 9<sup>th</sup> edition, Cay S. Horstmann and G Cornell, Pearson.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech III Semester**

**L/T/P/C**

**3/0/0/0**

**ENVIRONMENTAL SCIENCE AND TECHNOLOGY (A63MC3)**

**Course Objectives**

- Understanding the importance of ecological balance for sustainable development.
- Understanding the impacts of developmental activities and mitigation measures.
- Understanding the environmental policies and regulations

**Course Outcomes**

Based on this course, the Engineering graduate will understand /evaluate / develop technologies on the basis of ecological principles and environmental regulations which in turn helps in sustainable development.

**UNIT - I**

**Ecosystems**

Definition, Scope and Importance of ecosystem. Classification, structure, and function of an ecosystem, Food chains, food webs, and ecological pyramids. Flow of energy, Biogeochemical cycles, Bioaccumulation, Bio-magnification, ecosystem value, services and carrying capacity, Field visits.

**UNIT - II**

**Natural Resources:**

Classification of Resources, Living and Non-Living resources.

**Water resources**

Use and over utilization of surface and ground water, floods and droughts, Dams: benefits and problems.

**Mineral resources**

Use and exploitation, environmental effects of extracting and using mineral resources.

**Land resources**

Forest resources.

**Energy resources**

Growing energy needs, renewable and non-renewable energy sources, use of alternate energy source, case studies.

**UNIT - III**

**Biodiversity and Biotic Resources**

Introduction, Definition, genetic, species and ecosystem diversity. Value of biodiversity; consumptive use, productive use, social, ethical, aesthetic and optional values. India as a mega diversity nation, Hot spots of biodiversity. Field visit. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts; conservation of biodiversity: In-Situ and Ex-situ conservation. National Biodiversity act.

**UNIT - IV**

**Environmental Pollution and Control Technologies: Environmental Pollution:**

Classification of pollution.

**Air Pollution**

Primary and secondary pollutants, Automobile and Industrial pollution, ambient air quality standards.

**Water pollution**

Sources and types of pollution, drinking water quality standards.

**Soil Pollution:**

Sources and types, Impacts of modern agriculture, degradation of soil.

**Noise Pollution:**

Sources and Health hazards, standards.

**Solid waste**

Municipal Solid Waste management, composition and characteristics of e-Waste and its management.

**Pollution control technologies:**

Wastewater Treatment methods: Primary, secondary and Tertiary. Overview of air pollution control technologies, Concepts of bioremediation.

**Global Environmental Problems and Global Effort**

Climate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). Deforestation and desertification. International conventions / Protocols: Earth summit, Kyoto protocol, and Montréal Protocol.

**UNIT - V****Environmental Policy, Legislation & EIA**

Environmental Protection act, Legal aspects Air Act – 1981, Water Act, Forest Act, Wild life Municipal Solid waste Management and handling rules, biomedical waste management and handling rules, hazardous waste management and handling rules. EIA: EIA structure, methods of baseline data acquisition. Overview on Impacts of air, water, biological and Socio-economic aspects. Strategies for risk assessment, Concepts of Environmental Management Plan (EMP).

**Towards Sustainable Future**

Concept of Sustainable Development, Population and its explosion, Crazy Consumerism, Environmental Education, Urban Sprawl, Human health, Environmental Ethics, Concept of Green Building, Ecological Foot Print, Life Cycle assessment (LCA), Low carbon life style.

**TEXT BOOKS**

1. Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha for University Grants Commission.
2. Environmental Studies by R. Rajagopalan, Oxford University Press.

**REFERENCE BOOKS:**

1. Environmental Science: towards a sustainable future by Richard T. Wright. 2008 PHL Learning Private Ltd. New Delhi.
2. Environmental Engineering and science by Gilbert M. Masters and Wendell P. Ela. 2008 PHI Learning Pvt. Ltd.
3. Environmental Science by Daniel B. Botkin & Edward A. Keller, Wiley INDIA edition.
4. Environmental Studies by Anubha Kaushik, 4<sup>th</sup> Edition, New age international publishers.
5. Text book of Environmental Science and Technology - Dr. M. Anji Reddy 2007, BS Publications.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech IV Semester**

**L/T/P/C  
4/0/0/4**

**COMPUTER ORGANIZATION (A64PC1)**

**Course Objectives:**

- To understand basic components of computers.
- To understand the architecture of 8086 processor.
- To understand the instruction sets, instruction formats and various addressing modes of 8086.
- To understand the representation of data at the machine level and how computations are performed at machine level.
- To understand the memory organization and I/O organization.
- To understand the parallelism both in terms of single and multiple processors.

**Course Outcomes:**

- Able to understand the basic components and the design of CPU, ALU and Control Unit.
- Design the hardwired and micro-programmed control units.
- Ability to understand memory hierarchy and its impact on computer cost/performance.
- Ability to understand the advantage of instruction level parallelism and pipelining for high performance Processor design.
- Ability to understand the instruction set, instruction formats and addressing modes of 8086.
- Ability to write assembly language programs to solve problems.

**UNIT - I**

**Digital Computers**

Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture.

**Basic Computer Organization and Design**

Instruction codes, Computer Registers, Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt, Complete Computer Description.

**Hardwired control unit**

Control unit inputs, Control unit logic.

**Micro Programmed Control**

Control memory, Address sequencing, micro program example, design of control unit.

**UNIT - II**

**Central Processing Unit**

Processor Organization, Register Organization, Instruction cycle, The 8086 Processor Architecture, Register organization, Physical memory organization, General Bus Operation, I/O Addressing Capability, Special Processor Activities, Minimum and Maximum mode system and timings.

**8086 Instruction Set and Assembler Directives**

Machine language instruction formats, Addressing modes, Instruction set of 8086, Assembler directives and operators.

### **UNIT - III**

#### **Assembly Language Programming with 8086**

Machine level programs, Machine instruction characteristics, types of operands, types of operations, Machine coding the programs, Programming with an assembler, Assembly Language example programs. Stack structure of 8086, Interrupts and Interrupt service routines, Interrupt cycle of 8086, Interrupt programming, Passing parameters to procedures, Macros, Timings and Delays.

### **UNIT - IV**

#### **Computer Arithmetic**

Introduction, The arithmetic logic unit, Integer representation, Integer arithmetic, Floating point arithmetic, Floating point representation, Addition and Subtraction, Multiplication Algorithms, Division Algorithms, Floating - point Arithmetic operations.

#### **Input-Output Organization**

Peripheral Devices, Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt, Direct memory Access, Input –Output Processor (IOP), Intel 8089 IOP.

### **UNIT - V**

#### **Memory Organization**

Memory Hierarchy, Auxiliary memory, Associate Memory, Cache Memory.

#### **Pipeline and Vector Processing**

Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processors.

#### **Multi Processors**

Characteristics of Multiprocessors, Interconnection Structures, Inter processor arbitration, Inter processor communication, and synchronization.

### **TEXT BOOKS**

1. Computer System Architecture, M. Moris Mano, Third Edition, Pearson. (**UNIST-I, IV, V**)
2. Advanced Microprocessors and Peripherals, K M Bhurchandi, A.K Ray ,3<sup>rd</sup> edition, McGraw Hill India Education Private Ltd. (**UNITS - II, III**).

### **REFERENCE BOOKS**

1. Microprocessors and Interfacing, D V Hall, SSSP Rao, 3<sup>rd</sup> edition, McGraw Hill India Education Private Ltd.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky: Computer Organization, 5<sup>th</sup> Edition, Tata McGraw Hill, 2002
3. Computer Organization and Architecture, William Stallings, 9th Edition, Pearson.
4. David A. Patterson, John L. Hennessy: Computer Organization and Design – The Hardware / Software Interface ARM Edition, 4<sup>th</sup> Edition, Elsevier, 2009.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech IV Semester**

**L/T/P/C  
4/0/0/4**

**DATABASE MANAGEMENT SYSTEMS (A64PC2)**

**Course Objectives**

- To understand the basic concepts and the applications of database systems. To master the basics of SQL and construct queries using SQL.
- To understand the relational database design principles.
- To become familiar with the basic issues of transaction processing and concurrency control.
- To become familiar with database storage structures and access techniques.

**Course Outcomes**

- Demonstrate the basic elements of a relational database management system.
- Ability to identify the data models for relevant problems.
- Ability to design entity relationship model and convert entity relationship diagrams into RDBMS and formulate SQL queries on the data.
- Apply normalization for the development of application software.
- Understand transaction processing, concurrency control and recovery techniques.

**UNIT - I**

**Introduction**

Introduction and basic concepts, File organization for conventional data management system, Higher-level file organization for DBMSs, Database System Applications, Purpose of Database Systems, View of Data, Database Languages – DDL, DML, Relational Databases, Database Design, Data Storage and Querying, Transaction Management, Database Architecture, Data Mining and Information Retrieval, Specialty Databases, Database Users and Administrators, History of Database Systems.

**Introduction to Data base design**

Database Design and ER diagrams, Entities, Attributes and Entity sets, Relationships and Relationship sets, Additional features of ER Model, Conceptual Design with the ER Model, Conceptual Design for Large enterprises.

**Relational Model**

Introduction to the Relational Model, Integrity Constraints over Relations, Enforcing Integrity constraints, Querying relational data, Logical data base Design: ER to Relational, Introduction to Views, Destroying /Altering Tables and Views.

**UNIT - II**

Relational Algebra and Calculus: Preliminaries, Relational Algebra, Relational calculus Tuple relational Calculus, Domain relational calculus, Expressive Power of Algebra and calculus.

**SQL**

SQL data definition and Data types, Schema and catalog concepts in SQL, Queries, Constraints, Triggers: Form of Basic SQL Query, UNION, INTERSECT, and EXCEPT, Nested Queries, Aggregate Operators, NULL values Complex Integrity Constraints in SQL, Triggers and Active Data bases, Designing Active Databases.

## **UNIT - III**

### **Schema Refinement and Normal Forms**

Introduction to Schema Refinement, Functional Dependencies - Reasoning about FDs, Normal Forms - 1NF, 2NF, 3NF, Properties of Decompositions, Normalization, Schema Refinement in Database Design, Other Kinds of Dependencies.

### **Database design**

Overview of the design process, Case study of a design process, mapping the conceptual database model into Conceptual schemas.

## **UNIT - IV**

### **Transaction Management**

Transactions, Transaction Concept, A Simple Transaction Model, Storage Structure, Transaction Atomicity and Durability, Transaction Isolation, serializability, Transaction Isolation and Atomicity Transaction Isolation Levels, Implementation of Isolation Levels.

### **Concurrency Control**

Lock-Based Protocols, Multiple Granularity, Timestamp-Based Protocols, Validation-Based Protocols, multi-version Schemes. Recovery System-Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm, Buffer Management, Failure with loss of non-volatile storage, Early Lock Release and Logical Undo Operations, Remote Backup systems.

## **UNIT - V**

### **Storage and Indexing**

Overview of Storage and Indexing: Data on External Storage, File Organization and Indexing, Index Data Structures, and Comparison of File Organizations. Tree-Structured Indexing: Intuition for tree Indexes, Indexed Sequential Access Method (ISAM), B+ Trees: A Dynamic Index Structure, Search, Insert, Delete.

### **Hash- Based Indexing**

Static Hashing, Extendible hashing, Linear Hashing, Extendible vs. Linear Hashing.

### **Emerging Technologies**

Introduction to Web Database programming using PHP, XML, Data mining concepts, Overview of data warehousing and PLAP.

## **TEXT BOOKS**

1. Data base Management Systems, Raghu Ramakrishnan, Johannes Gehrke, McGraw Hill Education (India) Private Limited, 3<sup>rd</sup> Edition. (**Part of UNIT-I, UNIT-II, UNIT-III, UNIT-V**)
2. Data base System Concepts, A. Silberschatz, Henry. F. Korth, S. Sudarshan, McGraw Hill Education(India) Private Limited l, 6<sup>th</sup> edition.( **Part of UNIT-I, UNIT-IV**).

## **REFERENCE BOOKS:**

1. Database Systems, 6<sup>th</sup> edition, R Elmasri, Shamkant B.Navathe, Pearson Education.
2. Database System Concepts, Peter Rob & Carlos Coronel, Cengage Learning.
3. Introduction to Database Management, M. L. Gillenson and others, Wiley Student Edition.
4. Database Development and Management, Lee Chao, Auerbach publications, Taylor & Francis Group.
5. Introduction to Database Systems, C. J. Date, Pearson Education.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech IV Semester**

**L/T/P/C  
4/0/0/4**

**OPERATING SYSTEMS (A64PC3)**

**Course Objectives**

- To understand the OS role in the overall computer system
- To study the operations performed by OS as a resource manager
- To understand the scheduling policies of OS
- To understand the different memory management techniques
- To understand process concurrency and synchronization
- To understand the concepts of input/output, storage and file management
- To understand the goals and principles of protection
- Introduce system call interface for file and process management.
- To study different OS and compare their features.

**Course Outcomes**

- Ability to design and solve synchronization problems.
- Learn about minimization of turnaround time, waiting time and response time and also maximization of throughput by keeping CPU as busy as possible.
- Understand the process scheduling algorithms.
- Ability to change access controls to protect files.
- Ability to compare the different operating systems.
- Understands the windows vista operating system design.

**UNIT - I**

**Overview**

Introduction-Operating system objectives, User view, System view, Operating system definition, Computer System Organization, Computer System Architecture, OS Structure, OS Operations, Process Management, Memory Management, Storage Management, Protection and Security, Computing Environments. Operating System services, User and OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, OS Structure.

**UNIT - II**

**CPU Scheduling Process**

concepts-The Process, Process State, Process Control Block, Threads, Process Scheduling-Scheduling Queues, Schedulers, Context Switch, Operations on Processes, System calls-fork(),exec(),wait(),exit(), Inter-process communication-ordinary pipes and named pipes, message queues, shared memory, in Unix.

**Process Scheduling**

Basic concepts, Scheduling Criteria, Scheduling algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Thread scheduling, Linux scheduling and Windows scheduling. Process Synchronization, Background, The Critical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization in Linux and Windows.

### **UNIT - III**

#### **Memory Management and Virtual Memory**

Memory Management Strategies- Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of Page Table, IA-32 Segmentation, IA-32 Paging.

#### **Virtual Memory Management**

Background, Demand Paging, Copy-on-Write, Page Replacement, Page Replacement Algorithms, Allocation of Frames, Thrashing, Virtual memory in Windows.

### **UNIT - IV**

#### **Storage Management**

File System- Concept of a File, System calls for file operations - open (), read (), write (), close (), seek (), unlink (), Access methods, Directory and Disk Structure, File System Mounting, File Sharing, Protection.

#### **File System Implementation:**

File System Structure, File System Implementation, Directory Implementation, Allocation methods, Free-space Management, Efficiency, and Performance. Mass Storage Structure – Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap space Management

### **UNIT - V**

#### **Deadlocks**

System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.

#### **Protection**

System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection.

#### **CASE STUDY on WINDOWS VISTA**

History of windows vista, Programming windows vista, System structure, Process and threads in windows vista, memory management, Caching in windows vista, input/output in windows vista, the Windows NT file system, Security in windows vista.

### **TEXT BOOKS**

1. Operating System Concepts , Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 9th Edition, Wiley, 2016 India Edition
2. Operating Systems – Internals and Design Principles, W. Stallings, 7th Edition, Pearson.

### **REFERENCE BOOKS:**

1. Modern Operating Systems, Andrew S Tanenbaum, 3rd Edition, PHI
2. Operating Systems: A concept-based Approach, 2nd Edition, D.M. Dhamdhere, TMH.
3. Principles of Operating Systems, B. L. Stuart, Cengage learning, India Edition.
4. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
5. Principles of Operating systems, Naresh Chauhan, Oxford University Press.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech IV Semester**

**L/T/P/C  
3/0/0/3**

**FORMAL LANGUAGES AND AUTOMATA THEORY (A64PC4)**

**Course Objectives**

- To provide introduction to some of the central ideas of theoretical computer science from the perspective of formal languages.
- To introduce the fundamental concepts of formal languages, grammars and automata theory.
- Classify machines by their power to recognize languages.
- Employ finite state machines to solve problems in computing.
- To understand deterministic and non-deterministic machines.
- To understand the differences between decidability and un-decidability.

**Course Outcomes**

- Able to understand the concept of abstract machines and their power to recognize the languages.
- Able to employ finite state machines for modelling and solving computing problems.
- Able to design context free grammars for formal languages.
- Design Turing Machines.
- Able to distinguish between decidability and un-decidability.
- Able to gain proficiency with mathematical tools and formal methods.

**UNIT - I**

**Introduction**

To Finite Automata, Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems. Deterministic Finite Automata, Nondeterministic Finite Automata, an application: Text Search, Finite Automata with Epsilon-Transitions, Finite automata with output - Mealy and Moore machines, Equivalence of Mealy and Moore machines.

**UNIT - II**

**Regular Expressions**

Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Automata and Regular expressions, Converting DFA's to Regular Expressions, Converting Regular Expressions to DFA, Properties of Regular Languages-Pumping Lemma for Regular Languages, Applications of the Pumping Lemma, Closure Properties of Regular Languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

**UNIT - III**

**Context-Free Grammars**

Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.

## **Push Down Automata**

Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Deterministic Pushdown Automata, non-deterministic pushdown automata, power of Deterministic Pushdown Automata and Non-Deterministic Pushdown Automata.

## **UNIT - IV**

### **Normal Forms for Context- Free Grammars**

The Pumping Lemma for Context-Free Languages, Closure Properties of Context-Free Languages. Decision Properties of CFL's - Complexity of Converting among CFG's and PDA's, Running time of conversions to Chomsky Normal Form.

### **Introduction to Turing Machines**

Problems That Computers Cannot Solve, The Turing Machine, Programming Techniques for Turing Machines, Extensions to the basic Turing machine, Restricted Turing Machines, Turing Machines, and Computers.

## **UNIT - V**

### **Undecidability**

A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines, Post's Correspondence Problem, Other Undecidable Problems,

### **Intractable Problems**

Polynomial time and space, Some NP-complete problems, The Classes P and NP, An NP-Complete Problem.

## **TEXT BOOKS:**

1. Introduction to Automata Theory, Languages, and Computation, 3<sup>rd</sup> Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Introduction to the Theory of Computation, Michael Sipser, 3<sup>rd</sup> edition, Cengage Learning.

## **REFERENCE BOOKS:**

1. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
2. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
3. A Text book on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge University Press.
4. Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R, Pearson.
5. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekar, 2nd edition, PHI.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech IV Semester**

**L/T/P/C  
3/0/0/3**

**BUSINESS ECONOMICS AND FINANCIAL ANALYSIS (A64HS5)**

**Course Objectives:**

- To learn the basic Business types, impact of the Economy on Business and Firms specifically.
- To analyze the Business from the Financial Perspective.

**Course Outcomes:**

- The students will understand the various Forms of Business and the impact of economic variables on the Business.
- The Demand, Supply, Production, Cost, Market Structure, Pricing aspects are learnt.
- The Students can study the firm's financial position by analysing the Financial Statements of a Company.

**UNIT – I**

**Introduction to Business and Economics:**

**Business**

Structure of Business Firm, Theory of Firm, Types of Business Entities, Limited Liability Companies, Sources of Capital for a Company, Non-Conventional Sources of Finance.

**Economics**

Significance of Economics, Micro and Macro Economic Concepts, Concepts and Importance of National Income, Inflation, Money Supply in Inflation, Business Cycle, Features and Phases of Business Cycle. Nature and Scope of Business Economics, Role of Business Economist, Multidisciplinary nature of Business Economics.

**UNIT – II**

**Demand and Supply Analysis:**

**Elasticity of Demand**

Elasticity, Types of Elasticity, Law of Demand, Measurement and Significance of Elasticity of Demand, Factors affecting Elasticity of Demand, Elasticity of Demand in decision making, Demand Forecasting: Characteristics of Good Demand Forecasting, Steps in Demand Forecasting, Methods of Demand Forecasting.

**Supply Analysis**

Determinants of Supply, Supply Function & Law of Supply.

**UNIT- III**

**Production, Cost, Market Structures & Pricing**

**Production Analysis**

Factors of Production, Production Function, Production Function with one variable input, two variable inputs, Returns to Scale, Different Types of Production Functions.

**Cost analysis**

Types of Costs, Short run and Long run Cost Functions.

**Market Structures**

Nature of Competition, Features of Perfect competition, Monopoly, Oligopoly, and Monopolistic Competition.

## **Pricing**

Types of Pricing, Product Life Cycle based Pricing, Break Even Analysis, and Cost Volume Profit Analysis.

## **UNIT - IV**

### **Financial Accounting**

Accounting concepts and Conventions, Accounting Equation, Double-Entry system of Accounting, Rules for maintaining Books of Accounts, Journal, Posting to Ledger, Preparation of Trial Balance, Elements of Financial Statements, and Preparation of Final Accounts.

## **UNIT - V**

### **Financial Analysis through Ratios**

Concept of Ratio Analysis, Liquidity Ratios, Turnover Ratios, Profitability Ratios, Proprietary Ratios, Solvency, Leverage Ratios (simple problems). Introduction to Fund Flow and Cash Flow Analysis (simple problems).

## **TEXT BOOKS**

1. D. D. Chaturvedi, S. L. Gupta, Business Economics - Theory and Applications, International Book House Pvt. Ltd. 2013.
2. Dhanesh K Khatri, Financial Accounting, Tata McGraw Hill, 2011.
3. Geethika Ghosh, Piyali Gosh, Purba Roy Choudhury, Managerial Economics, 2e, Tata McGraw Hill Education Pvt. Ltd. 2012.

## **REFERENCES:**

1. Paresh Shah, Financial Accounting for Management 2e, Oxford Press, 2015.
2. S. N. Maheshwari, Sunil K Maheshwari, Sharad K Maheshwari, Financial Accounting, 5e, Vikas Publications, 2013.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech IV Semester**

**L/T/P/C  
0/0/3/2**

**COMPUTER ORGANIZATION LAB (A64PC6)**

**Exercises in Digital Logic Design:**

1. Implement Logic gates using NAND and NOR gates
2. Design a Full adder using gates
3. Design and implement the 4:1 MUX, 8:1 MUX using gates /ICs.
4. Design and Implement a 3 to 8 decoder using gates
5. Design a 4 bit comparator using gates/IC
6. Design and Implement a 4 bit shift register using Flip flops
7. Design and Implement a Decade counter

**Exercises in Micro Processor programming:**

Write assembly language programs for the following using GNU Assembler.

1. Write assembly language programs to evaluate the expressions:
  - i)  $a = b + c - d * e$
  - ii)  $z = x * y + w - v + u / k$
  - a. Considering 8-bit, 16 bit and 32 bit binary numbers as b, c, d, e.
  - b. Considering 2 digit, 4 digit and 8 digit BCD numbers. Take the input in consecutive memory locations and also Display the results by using “int xx” of 8086. Validate program for the boundary conditions.
2. Write an ALP of 8086 to take N numbers as input. And do the following operations on them.
  - a. Arrange in ascending and descending order.
3. Write an ALP of 8086 to take N numbers as input. And do the following operations on them.
  - a. Find max and minimum
  - b. Find averageConsidering 8-bit, 16 bit binary numbers and 2 digit, 4digit and 8 digit BCD numbers. Display the results by using “int xx” of 8086. Validate program for the boundary conditions.
4. Write an ALP of 8086 to take a string of as input (in ‘C’ format)and do the following Operations on it.
  - a. Find the length
  - b. Find it is Palindrome or n.Considering 8-bit, 16 bit binary numbers and 2 digit, 4digit and 8 digit BCD numbers. Display the results by using “int xx” of 8086. Validate program for the boundary conditions.
5. Write an ALP of 8086 to take a string of as input (in ‘C’ format) and do the following Operations on it.
  - a. Find whether given string substring or not.

6. Write an ALP of 8086 to take a string of as input (in 'C' format) and do the following Operations on it
  - a. Find the Armstrong number
  - b. Find the Fibonacci series for n numbersDisplay the results by using "int xx" of 8086.
7. Write the ALP to implement the above operations as procedures and call from the main procedure.
8. Write an ALP of 8086 to find the factorial of a given number as a Procedure and call from the main program which display the result.

#### **REFERENCE BOOKS**

1. Switching theory and logic design –A. Anand Kumar PHI, 2013
2. Advanced microprocessor & Peripherals-A. K. Ray and K. M. Bherchandavi, TMH, 2nd edition.
3. Switching and Finite Automatic theory-Zvi Kohavi, Niraj K.Jha Cambridge, 3rd edition
4. Digital Design –Morris Mano, PHI, 3rd edition
5. Microprocessor and Interfacing –Douglas V. Hall, TMGH 2nd edition.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech IV Semester**

**L/T/P/C  
0/0/3/2**

**DATABASE MANAGEMENT SYSTEM LAB (A64PC7)**

**Course Objectives**

This lab enables the students to practice the concepts learnt in the subject DBMS by developing a database for an example company named “Roadway Travels” whose description is as follows. The student is expected to practice the designing, developing and querying a database in the context of example database “Roadway travels”. Students are expected to use “Mysql” database.

**Course Outcomes**

- Ability to design and implement a database schema for given problem.
- Apply the normalization techniques for development of application software to realistic problems.
- Ability to formulate queries using SQL DML/DDI/DCL commands.
- Develop application programs using PL/SQL

**Roadway Travels**

"**Roadway Travels**" is in business since 1997 with several buses connecting different places in India. Its main office is located in Hyderabad. The company wants to *computerize its operations* in the following areas:

**Reservations & Cancellation**

Reservations are directly handled by booking office. Reservations can be made 30 days in advance and tickets issued to passenger. One Passenger/person can book many tickets (to his/her family).

Cancellations are also directly handed at the booking office.

In the process of computerization of Roadway Travels you have to design and develop a Database which consists the data of Buses, Passengers, Tickets, and Reservation and cancellation details. You should also develop query's using SQL to retrieve the data from the database.

The above process involves many steps like 1. Analyzing the problem and identifying the Entities and Relationships, 2. E-R Model 3. Relational Model 4. Normalization 5. Creating the database 6. Querying. Students are supposed to work on these steps week wise and finally create a complete “Database System” to Roadway Travels. Examples are given at every experiment for guidance to students.

**Experiment 1 E-R Model**

Analyse the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc.

Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Example:

Entities: 1. Bus 2. Ticket 3. Passenger

Relationships 1. Reservation. 2. Cancellation.

Primary Key Attributes: 1. Ticket ID (Ticket Entity) 2. Passport ID (Passenger Entity)  
3. Bus\_No. (Bus Entity)

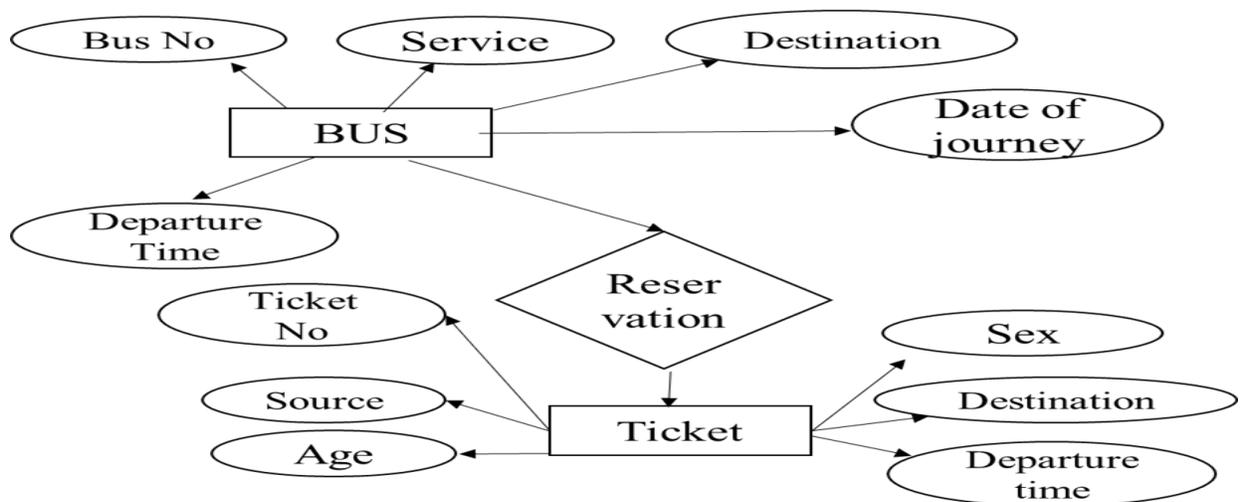
Apart from the above mentioned entities you can identify more. The above mentioned are few.

**Note:** The student is required to submit a document by writing the Entities and Keys to the lab teacher.

### Experiment 2 Concept design with E-R Model

Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc wherever required.

#### Example: E-R diagram for bus



**Note:** The student is required to submit a document by drawing the E-R Diagram to the lab teacher.

### Experiment 3 Relational Model

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality. Represent attributes as columns in tables or as tables based on the requirement. Different types of attributes (Composite, Multi-valued, and Derived) have different way of representation.

### Example

The passenger tables look as below. This is an example. You can add more attributes based on your E-R model. This is not a normalized table.

Passenger Name	Age	Sex	Address	PassportID	Ticket_ID

**Note:** *The student is required to submit a document by Represent relationships in a tabular fashion to the lab teacher.*

### Experiment 4 Normalization

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only. For the above table in the First normalization we can remove the multi valued attribute Ticket\_id and place it in another table along with the primary key of passenger.

**First Normal Form: The above table can be divided into two tables as shown below.**

Passenger Name	Age	Sex	Address	PassportID

PassportID	Ticket_ID

You can do the second and third normal forms if required. Any, how Normalized tables are given at the end.

### Experiment 5 Installation of MySQL and practicing DDL commands

Installation of MySQL. In this week you will learn Creating databases, How to create tables, altering the database, dropping tables and databases if not required. You will also try truncate, rename commands etc.

Example for creation of a normalized "Passenger" table.

```
CREATE TABLE Passenger ( Passport_id INTEGER PRIMARY KEY, Name VARCHAR (50) Not NULL, Age Integer Not NULL, Sex Char, Address VARCHAR (50) Not NULL);
```

Similarly create all other tables.

**Note: Detailed creation of tables is given at the end.**

### Experiment 6 Practicing DML commands

DML commands are used to for managing data within schema objects. Some examples:

SELECT - retrieve data from the a database

INSERT - insert data into a table

UPDATE - updates existing data within a table

DELETE - deletes all records from a table, the space for the records remain

### Inserting values into “Bus” table:

Insert into Bus values (1234,'hyderabad', 'tirupathi');

Insert into Bus values (2345,'hyderabad', 'Banglore');

Insert into Bus values (23,'hyderabad', 'Kolkata');

Insert into Bus values (45,'Tirupathi', 'Banglore');

Insert into Bus values (34,'hyderabad', 'Chennai');

### Inserting values into “Passenger” table:

Insert into Passenger values (1, 45,'ramesh', 45,'M', 'abc123');

Insert into Passenger values (2, 78,'geetha', 36,'F', 'abc124');

Insert into Passenger values (45, 90,'ram', 30,'M', 'abc12');

Insert into Passenger values (67, 89,'ravi', 50,'M', 'abc14');

Insert into Passenger values (56, 22,'seetha', 32,'F', 'abc55');

### Few more Examples of DML commands:

Select \* from Bus; (selects all the attributes and display)

UPDATE BUS SET Bus No = 1 WHERE BUS NO=2;

## Experiment 7 Querying

In this week you are going to practice queries (along with sub queries) using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

### Practice the following Queries

1. Display unique PNR\_No of all passengers.
2. Display all the names of male passengers.
3. Display the ticket numbers and names of all the passengers.
4. Find the ticket numbers of the passengers whose name start with 'r' and ends with 'h'.
5. Find the names of passengers whose age is between 30 and 45.
6. Display all the passengers names beginning with 'A'
7. Display the sorted list of passengers names

## Experiment 8 and Experiment 9: Querying (continued...)

You are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

1. Write a Query to display the Information present in the Passenger and cancellation tables.

**Hint:** Use UNION Operator.

2. Display the number of days in a week on which the 9W01 bus is available.
3. Find number of tickets booked for each PNR\_no using GROUP BY CLAUSE. **Hint:** Use GROUP BY on PNR\_No.
4. Find the distinct PNR numbers that are present.
5. Find the number of tickets booked by a passenger where the number of seats is greater than 1.

**Hint:** Use GROUP BY, WHERE and HAVING CLAUSES. 6. Find the total number of cancelled seats.

## Experiment 10 PL/SQL

1. Write a PL/SQL block for Addition of Two Numbers
2. Write a PL/SQL block for IF Condition
3. Write a PL/SQL block for IF and else condition
4. Write a PL/SQL block for greatest of three numbers using IF AND ELSEIF
5. Write a PL/SQL block for summation of odd numbers using for LOOP.

## Experiment 11 Triggers

In this week you are going to work on Triggers. Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database.

Eg: **CREATE TRIGGER updcheck BEFORE UPDATE ON passenger**

**FOR EACH ROW**

**BEGIN**

**IF NEW.TickentNO > 60 THEN SET New.Tickent no = Ticket no;**

```
ELSE
SET New.Ticketno = 0;
END IF; END;
```

### Experiment 12 Procedures

In this session you are going to learn Creation of stored procedure, Execution of procedure and modification of procedure. Practice procedures using the above database.

```
Eg: CREATE PROCEDURE myProc()
BEGIN
SELECT COUNT (Tickets) FROM Ticket WHERE age>=40;
End;
```

### Experiment 13 Cursors

In this week you need to do the following: Declare a cursor that defines a result set. Open the cursor to establish the result set. Fetch the data into local variables as needed from the cursor, one row at a time. Close the cursor when done.

```
CREATE PROCEDURE myProc(in_customer_id INT)
BEGIN
DECLARE v_id INT;
DECLARE v_name VARCHAR (30);
DECLARE c1 CURSOR FOR SELECT stdId,stdFirstname FROM students
WHERE stdId=in_customer_id;
OPEN c1;
FETCH c1 into v_id, v_name; Close c1;
END;
```

### Tables

#### BUS

Bus No: Varchar: PK (public key) Source : Varchar Destination : Varchar

#### Passenger

PPNO: Varchar (15)) : PK Name: Varchar (15) Age : int (4) Sex:Char (10) : Male / Female Address: VarChar (20)

#### Passenger\_Tickets

PPNO: Varchar (15)) : PK Ticket\_No: Numeric (9)

#### Reservation

PNR\_No: Numeric (9): FK Journey\_date: datetime (8) No\_of\_seats: int (8) Address: Varchar (50) Contact\_No: Numeric (9) --> Should not be less than 9 and Should not accept any other character other than Integer Status: Char (2) : Yes / No

#### Cancellation

PNR\_No: Numeric(9) : FK Journey\_date : datetime(8) No\_of\_seats : int (8) Address : Varchar (50) Contact\_No: Numeric (9) --> Should not be less than 9 and Should not accept any other character other than Integer Status: Char (2) : Yes / No

#### Ticket

Ticket\_No: Numeric (9): PK Journey\_date : datetime(8) Age : int (4) Sex:Char(10) : Male / Female Source : Varchar Destination : Varchar Dep\_time : Varchar

### REFERENCE BOOKS

1. Introduction to SQL, Rick F. Vander Lans, Pearson education.
2. Oracle PL/SQL, B. Rosenzweig and E. Silvestrova, Pearson education.
3. SQL & PL/SQL for Oracle 10 g, Black Book, Dr. P. S. Deshpande, Dream Tech.
4. Oracle Database 11 g PL/SQL Programming, M. Mc Laughlin, TMH.



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech IV Semester**

**L/T/P/C  
0/0/3/2**

**OPERATING SYSTEMS LAB (A64PC8)**

**Course Objectives**

- To write programs in Linux environment using system calls.
- To implement the scheduling algorithms.
- To implement page replacement algorithms
- To implement file allocation methods.
- To understand and implement ipc mechanism using named and unnamed pipes.
- To develop solutions for synchronization problems using semaphores.

**Course Outcomes**

- Ability to develop application programs using system calls in Unix.
- Ability to implement inter process communication between two processes.
- Ability to design and solve synchronization problems.
- Ability to simulate and implement operating system concepts such as scheduling, deadlock management, file management, and memory management.
- Ability to understand the inter process communication.

**Use Linux operating system and GNU C compiler.**

**List of Programs:**

1. Write C Programs to simulate the following CPU scheduling algorithms:  
a) Round Robin b) SJF
2. Write C Programs to simulate the following CPU Scheduling algorithms:  
a) FCFS b) Priority
3. Write C Programs to simulate the following File Organization techniques:  
a) Single level directory b) Two level c) Hierarchical
4. Write C Programs to simulate the following File Allocation Methods:  
a) Contiguous b) Linked c) Indexed
5. Write a C program to copy the contents of one file to another using system calls.
6. Write a C program to simulate Bankers Algorithm for Dead Lock Avoidance
7. Write a C program to simulate Bankers Algorithm for Dead Lock Prevention
8. Write C programs to simulate the following page replacement algorithms:  
a) FIFO b) LRU c) LFU
9. Write C programs to simulate the following techniques of memory management:  
a) Paging b) Segmentation
10. Write a C program to implement the `ls | sort` command. (Use unnamed Pipe)
11. Write a C program to solve the Dining- Philosopher problem using semaphores.
12. Write C programs to implement ipc between two unrelated processes using named pipe.
13. Write C programs to implement ipc between two processes using  
a) Message queues b) Shared memory.
14. Write a C program to simulate multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories – system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.

**REFERENCE BOOKS:**

1. An Introduction to Operating Systems, P.C.P Bhatt, 2<sup>nd</sup> edition, PHI.
2. Unix System Programming Using C++, Terrence Chan, PHI/Pearson.
3. Modern Operating Systems, Andrew S Tanenbaum, 3rd Edition, PHI



**T K R COLLEGE OF ENGINEERING & TECHNOLOGY  
(Autonomous)**

**B.TECH. INFORMATION TECHNOLOGY**

**B.Tech IV Semester**

**L/T/P/C  
0/0/3/2**

**GENDER SENSITIZATION LAB (A64MC2)**

**Course Objectives**

- To develop students' sensibility with regard to issues of gender in contemporary India.
- To provide a critical perspective on the socialization of men and women.
- To introduce students to information about some key biological aspects of genders. To expose the students to debates on the politics and economics of work.
- To help students reflect critically on gender violence.
- To expose students to more egalitarian interactions between men and women

**Course Outcomes:**

- Students will have developed a better understanding of important issues related to gender in contemporary India.
- Students will be sensitized to basic dimensions of the biological, sociological, psychological and legal aspects of gender. This will be achieved through discussion of materials derived from research, facts, everyday life, literature, and film.
- Students will attain a finer grasp of how gender discrimination works in our society and how to counter it.
- Students will acquire insight into the gendered division of labour and its relation to politics and economics.
- Men and women students and professionals will be better equipped to work and live together as equals.
- Students will develop a sense of appreciation of women in all walks of life.
- Through providing accounts of studies and movements as well as the new laws that provide protection and relief to women, the textbook will empower students to understand and respond to gender violence.

**UNIT - I**

**UNDERSTANDING GENDER**

**Gender:**

Why Should We Study It? (*Towards a World of Equals*: Unit -1)

**Socialization:**

Making Women, Making Men (*Towards a World of Equals*: Unit-2)

Introduction. Preparing for Womanhood. Growing up Male. First lessons in Caste. Different Masculinities.

**UNIT - II**

**GENDER AND BIOLOGY**

**Missing Women**

Sex Selection and Its Consequences (*Towards a World of Equals*: Unit -4) Declining Sex Ratio. Demographic Consequences.

**Gender Spectrum:**

Beyond the Binary (*Towards a World of Equals*: Unit -10) Two or Many? Struggles with Discrimination.

### **UNIT - III**

#### **GENDER AND LABOUR**

##### **Housework**

The Invisible Labour (*Towards a World of Equals*: Unit -3) “My Mother doesn’t Work.”  
“Share the Load.”

##### **Women’s Work:**

Its Politics and Economics (*Towards a World of Equals*: Unit -7) Fact and Fiction.  
Unrecognized and Unaccounted work. Additional Reading: Wages and Conditions of Work.

### **UNIT-IV**

#### **ISSUES OF VIOLENCE**

##### **Sexual Harassment**

Say No! (*Towards a World of Equals*: Unit -6) Sexual Harassment, not Eve-teasing- Coping with Everyday Harassment- Further Reading: “Chupulu”.

##### **Domestic Violence**

Speaking Out (*Towards a World of Equals*: Unit -8) Is Home a Safe Place? -When Women Unite [Film]. Rebuilding Lives. Additional Reading: New Forums for Justice. Thinking about Sexual Violence (*Towards a World of Equals*: Unit -11) Blaming the Victim-“I Fought for my Life....” - Additional Reading: The Caste Face of Violence.

### **UNIT - V**

#### **GENDER: CO - EXISTENCE**

##### **Just Relationships:**

Being Together as Equals (*Towards a World of Equals*: Unit -12) Mary Kom and Onler. Love and Acid just do not Mix. Love Letters. Mothers and Fathers. Additional Reading: Rosa Parks- The Brave Heart.

#### **TEXTBOOK**

All the five Units in the Textbook, “*Towards a World of Equals: A Bilingual Textbook on Gender*” written by A. Suneetha, Uma Bhrugubanda, Duggirala Vasanta, Rama Melkote, Vasudha Nagaraj, Asma Rasheed, Gogu Shyamala, Deepa Sreenivas and Susie Tharu and published by **Telugu Akademi, Hyderabad**, Telangana State in the year **2015**.

**Note:** *Since it is an Interdisciplinary Course, Resource Persons can be drawn from the fields of English Literature or Sociology or Political Science or any other qualified faculty who has expertise in this field from engineering departments.*

#### **REFERENCE BOOKS:**

1. Menon, Nivedita. Seeing like a Feminist. New Delhi: Zubaan-Penguin Books, 2012
2. Abdulali Sohaila. “*I Fought For My Life...and Won.*” Available online at: <http://www.thealternative.in/lifestyle/i-fought-for-my-lifeand-won-sohaila-abdulal/>