

**TKR COLLEGE OF ENGINEERING & TECHNOLOGY
(Autonomous)**

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

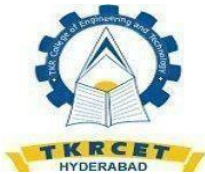
COURSE STRUCTURE & SYLLABUS

B.Tech. III Semester

S.No.	Course Code	Course Title	L	T	P	Credits
1.	BBSM3	Probability & Statistics	3	0	0	3
2.	B53PC2	Data Structures	3	0	0	3
3.	B53PC3	Digital Logic Design	3	0	0	3
4.	B53PC4	Object Oriented Programming	3	0	0	3
5.	B53PC5	Digital Logic Design lab	0	0	3	1.5
6.	B53PC6	Data Structures Lab	0	0	3	1.5
7.	BE23	Advanced Communication Skills Lab	0	0	3	1.5
8.	B53PC7	Object Oriented Programming Lab	0	0	3	1.5
9.	BITW1	IT Workshop	0	0	3	1.5
Total Credits						19.5

B.Tech. IV Semester

S.No.	Course Code	Course Title	L	T	P	Credits
1.	BBSM5	Discrete Mathematics	3	0	0	3
2.	B54PC2	Computer Organization	3	0	0	3
3.	B54PC3	Python Programming	3	0	0	3
4.	B54PC4	Database Management System	3	0	0	3
5.	B54PC5	Software Engineering	3	0	0	3
6.	B54PC6	Python Programming Lab	0	0	3	1.5
7.	B54PC7	Database Management System Lab	0	0	3	1.5
8.	B54PC8	Software Engineering lab	0	0	3	1.5
Total Credits						19.5



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

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

PROBABILITY & STATISTICS - BBSM3

B.Tech. III Semester

L/T/P C

3/0/0 3

COURSE OBJECTIVES:

1. The objective of this course is to familiarize the prospective engineers with statistical techniques
2. It aims to equip the students to deal with advanced level of mathematics and application that would be essential for their disciplines.
3. Random variables that describe randomness of an uncertainty in certain realistic situations.
4. Sampling Distributions of means- variance- Point Estimation and Interval Estimation.
5. The testing of Hypothesis.

Course Outcomes:

The students will learn:

- | | |
|--|----|
| 1. Solve the problems using the ideas of probability and random variables and various discrete and continuous probability distribution and their properties. | L3 |
| 2. Solve the problems using the basic ideas of statistics including correlation and regression. | L3 |
| 3. Analyze the statistical methods of studying data sample. | L4 |
| 4. Differentiate among many random variables involving the probability models which are quite useful for all branches of engineering. | L4 |
| 5. To evaluate Mean and Proportion and to make impotent decisions from few samples which are taken from unmanageable huge populations. | L5 |

UNIT-I:

Random variables:

Discrete and continuous Random variables- properties- Expectation of discrete and continuous Random variables and Variance of a sum.

UNIT-II:

Probability Distributions:

Binomial- poisson and Normal-evaluation of statistical parameters for these three distributions and problems.

UNIT-III:

Basic statistics:

Correlation and regression- Rank correlation- Curve fitting by the method of least squares- fitting of straight lines- second degree parabolas- power and exponential curves.

UNIT-IV:**Testing of hypothesis – I**

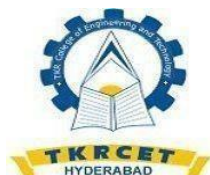
Null and Alternative hypothesis- critical region and types of errors- Test of significance- Large sample test for single proportion- difference of proportions- single mean- difference of means.

UNIT-V:**Testing of hypothesis – II**

Small samples Test for single mean- difference of two means- test for ratio of variances- Chi-square test for goodness of fit and independence of attributes.

Reference Books

1. Erwin Kreyszig- Advanced Engineering Mathematics- 9th Edition- John Wiley & Sons-2006.
2. N.P. Bali and Manish Goyal- A text book of Engineering Mathematics- Laxmi Publications-Reprint-2010.
3. B.S. Grewal- Higher Engineering Mathematics- Khanna Publishers- 35th Edition- 2000.
4. S.Ross- A First Course in Probability- 6th Ed.- Pearson Education in India- 2002.
5. W.Feller- An introduction to probability theory and its applications- Vol.1- 3rd edition Wiley-1968.



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

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

DATA STRUCTURES - B53PC2

B.Tech. III Semester

L/T/P C

3/0/0 3

Course Objective:

Understand the significance of data structures and apply them in building efficient algorithms.

Course Outcomes:

Upon completion of the course the student will be able to,

1. Apply knowledge, on how to compute time and space complexity for a given algorithm using asymptotic notations. Differentiate linear and nonlinear data structures. L3
2. Identify appropriate data structure for a given problem scenario. Illustrate array and linked representations of stack and queues. L3
3. Apply the knowledge of binary tree to implement tree traversal techniques, demonstrate how priority queues are implemented. Perform operations of max heap and disjoint set ADT with an example. L3
4. Compare searching, sorting and external sorting methods. L4
5. Make use graph search methods and perform operations on various search trees. L3

UNIT-I:

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, Circular linked lists-Operations for Circular 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, 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:

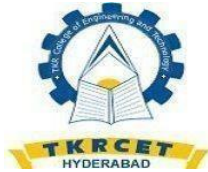
Binary Search Tree ADT, Definition, Operations- Searching, Insertion and Deletion, Balanced search trees-AVL Trees-Definition and Examples only, B-Trees-Definition and Examples only, Red-Black Trees-Definitions and Examples only, k-d trees, Comparison of Search Trees.

Text Books:

1. Data structures, Algorithms and Applications in C++, 2nd Edition, Sartaj Sahni, Universities Press.
2. Data structures and Algorithms in C++, Adam Drozdek, 4th 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++, 3rd edition, M. A. Weiss, Pearson.
4. Classic Data Structures, D. Samanta, 2nd edition, PHI.



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

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

DIGITAL LOGIC DESIGN - B53PC3

B.Tech. III Semester

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

Course Objective:

Understand the significance of converting from mechanical era to electronic era and to learn fundamentals of assembly language.

Course Outcomes:

Upon completion of the course the student will be able to:

- | | |
|---|----|
| 1. Analyze digital systems, by applying the knowledge of Boolean algebra and logic gates. Learn how to perform number base conversions and implement basic logic gates using universal gates. | L4 |
| 2. Apply the knowledge of K-Map method to simplify logic expressions and design combinational circuits for various arithmetic operations, code converters. | L3 |
| 3. Design registers, counters, flip flops and sequential circuits, evaluate the performance of flip flops, sequential circuits, and analyze the asynchronous sequential circuits for reliable operations. | L6 |
| 4. Develop assembly language programming skills by using 8086 instruction set, address modes, and assembler directives. | L6 |
| 5. Explore the memory organizations, decoding techniques, and programmable logic devices to understand memory organization in digital systems. | L3 |

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 (3, 4, 5, Variables) sum of products, product of sums simplification, don't care conditions.

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-III:

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.

Sequential Circuits: Circuits with latches, Design procedure, Reduction of state and follow tables, Race- free state assignment, Hazards.

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.

UNIT-IV:

8086 Instruction Set and Assembler Directives:

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

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.

UNIT-V:

Memory:

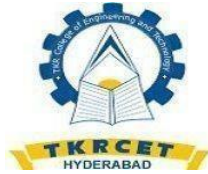
Introduction, Random-Access memory, Memory decoding, ROM, Programmable Logic Array, Programmable Array Logic, Sequential Programmable Devices.

Text Books:

1. Digital Design, M. Morris Mano, M.D.Ciletti, 5th edition, Pearson.
2. Computer System Architecture, M.Morris Mano, 3rd edition, Pearson.
3. Advanced Microprocessors and Peripherals, K M Bhurchandi, A.K Ray, 3rd edition, McGraw Hill India Education Private Ltd.

Reference Books:

1. Fundamentals of Logic Design, C. H. Roth, L. L. Kinney, 7th edition, Cengage Learning.
2. Fundamentals of Digital Logic & Micro Computer Design, 5TH Edition, M. Rafiquzzaman, John Wiley.
3. R.P.Jain “Modern Digital Electronics” Tata McGraw Hill, 4th edition 2009



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

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

OBJECT ORIENTED PROGRAMMING - B53PC4

B.Tech. III Semester

L/T/P C

3/0/0 3

Course Objective:

Understand the features of object oriented paradigm using JAVA programming.

Course Outcomes:

Upon completion of the course the student will be able to:

- | | |
|---|----|
| 1. Identify and apply various forms of inheritance, including specialization, specification, construction, extension, limitation, and combination. | L3 |
| 2. Make use of the knowledge of interfaces to develop the instances of multiple inheritances in Java. | L3 |
| 3. Identify and resolve the problem of deadlocks in multithreaded applications. | L3 |
| 4. Apply various utility classes for specialized task. | L3 |
| 5. Identify and utilize different layouts, event sources and listeners in the context of the delegation event model for effective event-driven programming and create simple applets on a web page. | L3 |

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 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 presumptive 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.

UNIT-V:**GUI Programming:**

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

Event Handling:

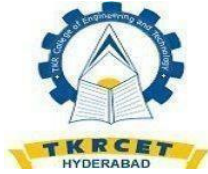
The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouseand keyboard events, Adapter classes, Inner classes, Anonymous Inner classes.

Text Books:

1. Java The complete reference, 9th 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, JohnWiley &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, 2nd edition, Oxford Univ. Press.



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

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

DIGITAL LOGIC DESIGN LAB - B53PC5

B.Tech. III Semester

L/T/P C

0/0/3 1.5

Course Objective:

Understand and implement the logic gates, combinational and sequential circuits using hardware components and to write 8086 assembly level language programs.

Course Outcomes:

After learning the contents of this course, the student will be able to:

1. Demonstrate proficiency in applying the basic principles of digital logic gates to design both combinational and sequential circuits, showcasing a foundational understanding. L3
2. Employ advanced analytical skills to critically assess and interpret outcomes from logic gates and diverse combinational as well as sequential circuits, achieving a higher-level understanding. L4
3. Exhibit a high level of competence by formulating algorithms and creating assembly language programs for varied problem scenarios using the 8086 Assembly Language Programming (8086ALP) architecture, demonstrating advanced skills and achieving proficiency. L5

Exercises in Digital Logic Design:

Design Logic gates using minimum number of Universal (NAND and NOR) gates.

1. Identify the logic gates required to design a Full Adder, and Design it using them.
2. Analyze and formulate the relationship between input lines and control lines in Multiplexers. Design and implement the 4:1 MUX, 8:1 MUX using gates/ICs.
3. Design and Implement a 3 to 8 decoder using gates
4. Design a 4 bit comparator using gates/IC
5. Design and Implement a 4 bit shift register using Flip flops and draw the timing diagrams
6. Design and Implement a Decade counter and draw the timing diagram.

Experiment # 01

Realization of Combinational circuits

Description: Five binary inputs of a digital logic circuit are designated as A, B, C, D and E. The circuit has three outputs, namely X, Y and Z. X should output 0, only if A is 0 and C and E both are 1. In all other cases X must remain 1. Y should output 0, if both B and C are 0 and D and E are 1. In all other cases Y must remain as 1. Z goes low if A, D and E are 1 and B and C are 0. Otherwise Z remains high. Prepare a truth table for the logic and then design a suitable circuit to implement it.

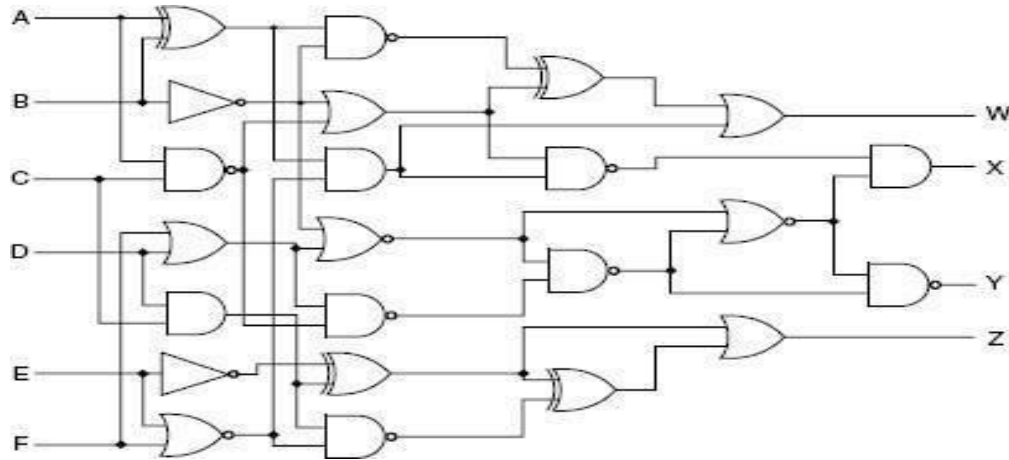
Task1: Identify the Logic gates required with their functionalities to implement the given scenario

Task2: Apply different inputs to the Logic gates and check the functionality

Task3: Prepare a Complete Truth table for the scenario

Task4: Design a suitable circuit to implement it.

1. Generate the truth table of the following circuit with six inputs A, B, C, D, E and F and four outputs W, X, Y and Z.



- 1) Study the Truth tables for different Logic gates
- 2) Apply different inputs to the Logic gates and check the functionality
- 3) Analyse the outputs of the Logic gates
- 4) Generate the truth table for the circuit.

- 1 b) A circuit has five inputs as A, B, C, D and E. Its six outputs are U, V, W, X, Y and Z. Design the circuit defined by the following truth table.

A	B	C	D	E	U	V	W	X	Y	Z
1	0	1	X	x	0	1	x	X	x	x
X	x	0	1	0	x	x	1	1	x	x
1	x	X	X	1	1	x	x	x	0	1
X	1	X	0	x	x	0	0	0	1	0

- 1) Study and Analyze different input combinations for the given table.
- 2) Develop the Truth table from the given table.
- 3) Design the circuit from the truth table.
- 4) Verify the output.

- 1 c) Design a 4-bit combinational circuit to increment (A circuit that adds one to a 4-bit binary number) using 4 Half-adders.
- 1) Understand the combinational circuit design
- 2) Analyze the functionality of an increment circuit.
- 3) Design the circuit using half adders

Experiment # 02
Design of ALU with two select-lines

Description: Design a 4-bit ALU with inputs A0–A3, B0–B3 and Cin. It is to have 4-bit output Y0–Y3 and Cout. The ALU would have two select lines to implement any one of the following four functions as per the pattern of select lines.

Select lines	Function	Remarks
00	$A + B + C_{in}$	Add with carry
01	$A - B - C_{in}$	Subtract with borrow
10	$A \text{ AND } B$	Logical AND
11	$A \text{ OR } B$	Logical OR

Task 1: Identify the Logic gates required to design an ALU.

Task 2: Design the Adder / subtractor circuit of ALU with the given condition.

Task 3: Design the Logical AND/Logical OR circuit of ALU with the given condition.

Task 4: Test the circuit to get the required output.

Experiment # 03
Design of ALU with two states

Description: Design an ALU capable of performing multiplication of two 4-bit numbers by producing 8-bit result, using Booth's algorithm. The unit may be hardware controlled in which case the complete hardware design is to be implemented. Alternately the unit may be micro-coded, in which case all signals and micro-steps are to be specified.

Task1: Identify the Logic gates required to design an ALU

Task2: Design the ALU circuitry for Multiplication

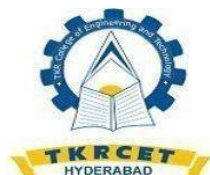
Task3: Test the circuit with different inputs for multiplication

Task4: Verify and record the output.

Exercises in Micro Processor programming:

Write assembly language programs for the following using GNU Assembler. Write assembly language programs to evaluate the expressions:

1. $a = b + c - d * e$ 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 inconsecutive 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 arrange in ascending and descending order.
3. Write an ALP of 8086 to take N numbers as input. Considering 8-bit, 16 bit binary numbers and 2 digit, 4 digit and 8 digit BCD numbers.
 - a) Find max and minimum
 - b) Find average Display the results by using “int xx” of 8086 and validate the program for the boundary conditions.
4. Write an ALP of 8086 to take a string as input (in ‘C’ format)
 - a) Find the length
 - b) Find is it Palindrome or not
5. Write an ALP of 8086 to take a string as input (in ‘C’ format) find whether given string is a substring or not.
6. Write an ALP of 8086 to take a string as input (in ‘C’ format)
 - a) Find the given number is an Armstrong number or not
 - b) Find the Fibonacci series for n numbers Display the results by using “int xx” of 8086.
7. Write the ALP to implement the above operations as procedure 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 to display the result.



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

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

DATA STRUCTURES LAB - B53PC6

B.Tech. III Semester

**L/T/P C
0/0/3 1.5**

Course Objective:

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

Course Outcomes:

The student will be able to do:

- | | |
|--|----|
| 1. Apply single linked list, doubly linked list, stack and double ended queue data structures for the given problem scenario and demonstrate the insertion, deletion and display operations. | L3 |
| 2. Apply binary search tree algorithms to the given problem scenario and demonstrate the in order, preorder and post order traversals. | L3 |
| 3. Apply various sorting algorithms for the given problem scenario. | L3 |
| 4. Apply binary search tree, hashing and extendible hashing techniques for the given problem scenario. | L3 |

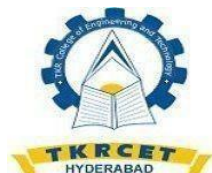
C++ Programming Concepts:

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 specifies, friend functions, constructors and destructor, Operator overloading, class templates, Inheritance and Polymorphism.

List of Programmes to be performed during the Course:

- Write a C++ program that uses functions to perform the following:
 - Create a singly linked list of integers.
 - Delete a given integer from the above linked list.
 - Display the contents of the above list after deletion.
- Write a template based C++ program that uses functions to perform the following:
 - Create a doubly linked list of elements.
 - Delete a given element from the above doubly linked list.
 - Display the contents of the above list after deletion.
- Write a C++ program that uses stack operations to convert a given infix expression into its postfix equivalent, Implement the stack using an array.
- Write a C++ program to implement a double ended queue ADT using an array, using a doubly linked list.
- Write a C++ program that uses functions to perform the following:
 - Create a binary search tree of characters.
 - Traverse the above Binary search tree recursively in preorder, in order and postorder.
- Write a C++ program that uses function templates to perform the following:

- 1) Search for a key element in a list of elements using linear search.
- 2) Search for a key element in a list of sorted elements using binarysearch.
7. Write a C++ program that implements Insertion sort algorithm to arrange a list of integers in elements in descending order.
8. Write a template based C++ program that implements Quick sort algorithm to arrange a list of elements in ascending order.
9. Write a C++ program that implements Heap sort algorithm for sorting a list of integers in ascending order.
10. Write a C++ program that implements Merge sort algorithm for sorting a list of integers in ascending order.
11. Write a C++ program to implement all the functions of a dictionary (ADT) using hashing.
12. Write a C++ program that implements Radix sort algorithm for sorting a list of integers in ascending order.
13. Write a C++ program that uses functions to perform the following:
 - 1) Create a binary search tree of integers.
 - 2) Traverse the above Binary search tree non-recursively in ignored.
14. Write a C++ program that uses functions to perform the following:
 - 3) Create a binary search tree of integers.
 - 4) Search for an integer key in the above binary search tree non-recursively.
 - 5) Search for an integer key in the above binary search tree recursively.
15. Write a C++ program to implement hashing using any hash function.
16. Write a C++ program to implement extendible hashing.



**TK R COLLEGE OF ENGINEERING & TECHNOLOGY
(Autonomous)**

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

ADVANCED COMMUNICATION SKILLS LAB - BE23

B.Tech. III Semester

**L/T/P C
0/0/3 1.5**

Introduction:

A course on Advanced Communication Skills (ACS) Lab is considered essential at the third year level of B.Tech and B.Pharmacy courses. At this stage, the students need to prepare themselves for their career which requires them to listen to, read, speak and write in English both for their professional and interpersonal communication. The main purpose of this course is to prepare the students of Engineering for their placements.

Course Objectives:

1. To improve students' fluency in spoken English.
2. To enable them to listen to English spoken at normal conversational speed To the students develop their vocabulary.
3. To read and comprehend texts in different contexts.
4. To communicate their ideas relevantly and coherently in writing To make students industry-ready.
5. To help students acquire behavioral skills for their personal and professional life To respond appropriately in different socio-cultural and professional contexts To sensitize the importance of Soft Skills and people skills.

Course Outcomes:

Students will be able to:

- | | |
|---|----|
| 1. Develop their interpersonal communication skills by building vocabulary and using appropriate body language. | L5 |
| 2. Distinguish the meaning of different tests like reading for facts; guessing meanings from context and improve proficiency in academic reading. | L4 |
| 3. Illustrate writing skills like appropriately drafting letters; compiling their resumes; doing e-correspondence and mastering technical report writing. | L3 |
| 4. Defend themselves while participating in group discussion and plan for effective presentations in JAM sessions/seminars/workshops etc. | L5 |
| 5. Defend themselves while participating in group discussion and plan for effective presentations in JAM sessions/seminars/workshops etc. | L4 |

Syllabus:

The following course activities will be conducted as part of the Advanced Communication Skills(ACS)Lab

UNIT-I:

Inter-personal Communication and Vocabulary Building – Starting a Conversation – Responding Appropriately and Relevantly – Using Appropriate Body Language – Role Play in Different Situations

– Synonyms and Antonyms, One-word Substitutes, Prefixes and Suffixes, Idioms and Phrases and Collocations.

UNIT-II:

Reading Comprehension–General Vs Local Comprehension, Reading for Facts, Guessing Meanings from Context, Skimming, Scanning, Inferring Meaning and practice with different texts.

UNIT-III:

Writing Skills – Structure and Presentation of Different Types of Writing – Letter writing / Resume Writing/ e-correspondence/statement of purpose/ Technical Report Writing/Styles-Types- Report in Manuscript format.

UNIT-IV:

Group Discussion & Presentation Skills:

Group Discussions-Dynamics of Group Discussion, Intervention, Summarizing, Modulation of Voice, Body Language, Relevance, Fluency and Organization of Ideas and Rubrics of Evaluation- Concept and Process Presentation Skills – Oral Presentations (individual or group) through JAM Sessions/ Seminars/ PPTs and Written Presentations through Posters/ Projects/ Reports/ emails/ Assignment.

UNIT-V:

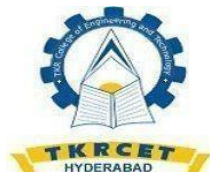
Interview Skills – Pre-interview Planning, Opening Strategies, Answering Strategies, Interview through Tele-conference & Video-conference and Mock Interviews.

Minimum Hardware Requirement:

1. Advanced English Communication Skills (ACS) Laboratory shall have the following infrastructural facilities to accommodate at least 35 students in the lab.
2. Spacious room with appropriate acoustics.
3. Eight round tables with five movable chairs for each table. Audio-visual aids.
4. LCD Projector.
5. Public Address system.
6. Computer with suitable configuration.
7. Suggested Software: The software consisting of the prescribed topics elaborated above should be procured and used.
8. Oxford Advanced Learner's Compass, 8th Edition.
9. DELTA's key to the Next Generation TOEFL Test: Advanced Skill Practice.

Reference Books:

1. Rizvi, M Ashraf. Effective Technical Communication. Mc Graw – Hill
2. Kumar, Sanjay and Pushp Lata. English for Effective Communication, OUP, 2015
3. Konar, Nira. English Language Laboratories – A Comprehensive Manual, PHI Learning Pvt Ltd, 2011.
4. Shiv Khera, You can Win, Macmillan Books, New York, 2003.
5. Jeff Butterfield, Soft Skills for Everyone, Cengage Learning, 2015.



**TK R COLLEGE OF ENGINEERING & TECHNOLOGY
(Autonomous)**

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

OBJECT ORIENTED PROGRAMMING LAB - B53PC7

B.Tech. III Semester

**L/T/P C
0/0/3 1.5**

Course Objective:

Write programs for solving real world problems using java language.

Course Outcomes:

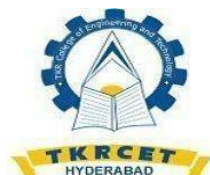
After learning the contents of this course, the student must be able to:

1. Implement and analyze complex algorithms that leverage Java collections for solving real-world problems. L4
2. Apply design principles to create abstract classes that model real-world entities with abstraction. L3
3. Utilize advanced features of the Java language, such as generics, lambda expressions, and streams, in the context of object-oriented design. L3
4. Identify and avoid common pitfalls leading to deadlocks in multithreaded programs. L3

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, adds 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. 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.
4. 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.
5. Write a Java program for the following:
 - 1) Create a doubly linked list of elements.
 - 2) Delete a given element from the above list.
 - 3) Display the contents of the list after deletion.
6. 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.

7. 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.
8. 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.
9. 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).
10. 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).
11. Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.
12. Write a Java program to list all the files in a directory including the files present in all its subdirectories.
13. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order.
14. 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.
15. 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).
16. 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).
17. Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.
18. Write a Java program to list all the files in a directory including the files present in all its subdirectories.
19. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order.
20. 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.



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

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

IT-WORKSHOP - BITW1

B.Tech. III Semester

**L/T/P C
0/0/3 1.5**

Course Objectives:

Training on PC Hardware, assembling, software installation, Internet, World Wide Web, and usage of productivity tools for documentation, Spreadsheet computations and Presentations.

Course Outcomes:

After learning the contents of this course, the student will be able to:

- | | |
|--|----|
| 1. Apply knowledge for computer assembling and software installation. | L3 |
| 2. Solve the trouble shooting problems. | L3 |
| 3. Apply the tools for preparation of PPT, Documentation and budget sheet etc. | L3 |
| 4. Create standard documents and research documents using Latex. | L6 |
| 5. Create project plans. | L6 |

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 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:**

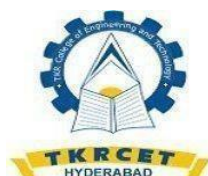
Cell Referencing, Formulae in spreadsheet – average, std. deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function, Sorting, Conditional formatting.

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. COMPUTER SCIENCE & ENGINEERING - R18

DISCRETE MATHEMATICS - BBSM5

B.Tech. IV Semester

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

Course Objectives:

1. Throughout the course students will be expected to demonstrate their understanding of Discrete Mathematics by being able to do each of the following.
2. Use mathematically correct terminology and notation
3. Construct correct direct and indirect proofs.
4. Use division into cases in a proof.
5. Use counter examples.
6. Apply logical reasoning to solve a variety of problems.

Course Outcomes:

1. Demonstrate a given logic sentence express it in terms of predicates- quantifiers and logical connectives. L3
2. For a given a problem- derive the solution using deductive logic and prove the solution based on logical inference. L5
3. For a given mathematical problem classify its algebraic structure. L4
4. Evaluate Boolean function s and simplify expressions using the properties of Boolean algebra. L5
5. Develop the given problems graph network and solve with techniques of graph theory. L5

UNIT-I

Sets & Relations:

Introduction of sets –Relations & functions – Types of Relations –Partial ordering-Total Ordering – Posets – Hasse Diagram.

UNIT-II

Counting Principles:

Principles of mathematical induction: The well ordering principle- recursive definition- division algorithm prime numbers-greatest common divisor Euclidean algorithm- fundamental theorem of arithmetic.

UNIT-III:

Mathematical Logic:

Propositional logic – Syntax semantics – validity and satisfiability – basic connectives and truth tables

– logical equivalence: the laws of logic – logical implication – rules of inference – use of quantifiers.

UNIT-IV:**Group Theory:**

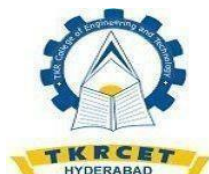
Algebraic structures – group – sub group – normal sub groups – cyclic groups – homomorphism – isomorphism.

UNIT-V:**Graph Theory:**

Graphs – Degree – Path – Subgraphs – isomorphism – eulerian and Hamiltonian graphs – planar graphs – Graph coloring.

Reference Books:

1. Graph theory by Narsing deo.
2. Discrete Mathematics by Joe L.Mott, Kandall
3. Kenneth H. Rosen- Discrete mathematics and its applications- Tata McGraw-Hill
4. Susanna S. Epp- Discrete Mathematics with applications 4th edition- Wadsworth publishing Co. Inc.
5. CLLIU and DPMohapatra- elements of discrete mathematics A Computer oriented Approach 3rd edition by- Tata McGraw-Hill.
6. J.P. Tremblay and R. Manohar- Discrete mathematical structure and its application to computer science” TMG edition Tata McGraw-Hill.
7. Norman L. Biggs- Discrete mathematics 2nd edition Oxford University Press Schaum's outlines series Seymour Lipchitz- Marc Lipson.



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

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

COMPUTER ORGANIZATION - B54PC2

B.Tech. IV Semester

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

Course Objective:

Understand the detailed computer architecture and organization, hardware operation of digital computer.

Course Outcomes:

Upon completion of the course the student will be able to:

- | | |
|---|----|
| 1. Make use of the principles to understand the organization of a computer. | L3 |
| 2. Proficiently exhibit the knowledge gained, on the study of the 8086 instruction set and assembler directives. | L3 |
| 3. Evaluate the organizational aspects of computer arithmetic and input/output (I/O) systems. | L5 |
| 4. Apply the concepts of memory hierarchy and Analyze its influence on the cost/performance. | L4 |
| 5. Analyze how the principles and mechanisms of concurrent processing are useful for processing computer systems. | L4 |

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, Micro program 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.

UNIT-III:

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-IV:**Memory Organization:**

Memory Hierarchy, Auxiliary memory, Associate Memory, Cache Memory, Virtual Memory, Memory Management Hardware.

UNIT-V:**Pipeline and Vector Processing:**

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

Multi Processors:

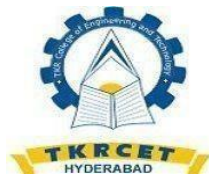
Characteristics of Multiprocessors, Interconnection Structures, Inter processor arbitration, Inter processor communication, and synchronization.

Text Books:

1. Computer System Architecture, M. Moris Mano, Third Edition, Pearson.
2. Advanced Microprocessors and Peripherals, K M Bhurchandi, A.K Ray, 3rd edition, McGrawHill India Education Private Ltd.

Reference Books:

1. Microprocessors and Interfacing, D V Hall, SSSP Rao, 3rd edition, McGraw Hill India Education Private Ltd.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky: Computer Organization, 5th 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, 4th Edition, Elsevier, 2009.



**TK R COLLEGE OF ENGINEERING & TECHNOLOGY
(Autonomous)**

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

PYTHON PROGRAMMING - B54PC3

B.Tech. IV Semester

L/T/P C

3/0/0 3

Course Objectives:

Learn the concepts of basic python programming used for developing the real time applications.

Course Outcomes:

Upon completion of the course the student will be able to:

1. Setup Python environment and execute Python scripts effectively. L3
2. Learn how to Solve problems by evaluating conditional and looping control structures. L6
3. Apply, the function definitions, calling methods, with various arguments. L3
4. Apply modules, packages, and implement them to write Python programs effectively. L3
5. Analyze and differentiate between errors and exceptions, implementing appropriate exception. L4

UNIT-I:

INTRODUCTION:

Introduction to Python, History, Need of Python Programming, features Applications, python environment setup, Basic syntax, Variables

Data Types: Lists, Tuples, dictionaries, sets, sequences, comprehensions. Keywords, Input-Output, Indentation, script structure, Running Python Scripts.

UNIT-II:

OPERATORS:

Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations, Conditional statements if, if-else Looping Control Structures for, while Control Statements: Break, Continue, Pass.

UNIT-III:

FUNCTIONS:

Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Anonymous Functions, Fruitful Functions (Function Returning Values), Scope of the Variables in a Function - Global and Local Variables.

File Handling Utilities: Threads, process, IPC (Inter Process Communication), Serialization & De-serialization, Exception Handling, Handling Multiple exception.

UNIT-IV:

FILES:

File input/output, Text processing file functions.

MODULES:

Introduction to modules, working with modules, Name spacing, Packages, using package implementing packages: numpy, iterator tools, scipy, matplotlib.

UNIT-V:**OBJECT ORIENTED PROGRAMMING IN PYTHON:**

Classes, 'self-variable', Methods, Constructor Method, Inheritance, Overriding Methods, Data hiding.

ERROR AND EXCEPTIONS:

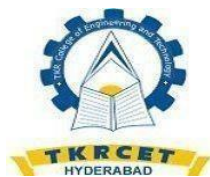
Difference between an error and Exception, Handling Exception, try except block, Raising Exceptions, User Defined Exceptions.

Text Books:

1. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
2. Wesley J. Chun “Core Python Programming”, Second Edition, Prentice Hall.

Reference Books:

1. Allen Downey, “Think Python”, Second Edition , Green Tea Press
2. Introduction to Computation & Programming Using Python, Spring 2013 Edition, By John V.Guttag.
3. Programming in Python 3: A Complete Introduction to the Python Language (Developer'sLibrary),by Mark Summerfield, 2nd Edition.



**TK R COLLEGE OF ENGINEERING & TECHNOLOGY
(Autonomous)**

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

DATABASE MANAGEMENT SYSTEM - B54PC4

B.Tech. IV Semester

L/T/P C

3/0/0 3

Course Objective:

Learn the approaches associated in building and implementing a Relational data model.

Course Outcomes:

Upon completing of the course the student will be able to:

- | | |
|--|----|
| 1. Develop a foundational understanding of database management systems (DBMS), their evolution, importance, and the fundamental concepts that govern their design and operation. | L3 |
| 2. Compare and evaluate the expressive power, capabilities, advantages, and Limitations of relational algebra and calculus in querying relational databases. | L5 |
| 3. Develop expertise in formulating complex SQL queries, using constraints, conditions, and expressions to retrieve, manipulate, and manage data based on specific criteria, requirements, and business rules. | L3 |
| 4. Develop a foundational understanding of transactions, their definition, properties, and significance in ensuring data consistency, integrity, and reliability in database systems. | L3 |
| 5. Compare and contrast index data structures with traditional file organizations. | L4 |

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.

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.

Introduction to Database 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.

UNIT-III:

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.

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.

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:

Indexing:

Index Data Structures, and Comparison with 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

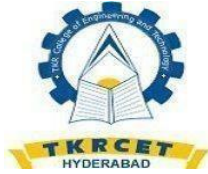
Static Hashing, Extendible hashing, Linear Hashing, Extendible vs. Linear Hashing.

Text Books:

1. Data base Management Systems, Raghu Ramakrishnan, Johannes Gehrke, McGraw Hill Education (India) Private Limited, 3rd Edition.
2. Data base System Concepts, A. Silberschatz, Henry. F. Korth, S. Sudarshan, McGraw Hill Education (India) Private Limited, 6th edition.

Reference Books:

1. Database Systems, 6th 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. COMPUTER SCIENCE & ENGINEERING - R18

SOFTWARE ENGINEERING - B54PC5

B.Tech. IV Semester

L/T/P C

3/0/0 3

Course Objective:

Learn about the software process models, SRS document, architectural styles, verification and validation of developed code and application prototype.

Course Outcomes:

After learning the contents of this course, the student will be able to:

- | | |
|--|----|
| 1. Identify common patterns in software development processes and apply different process models for software engineering. | L3 |
| 2. Classify various requirements and develop techniques for gathering and analyzing requirements. | L4 |
| 3. Apply fundamental design concepts in designing a software architecture based on given requirements. | L3 |
| 4. Estimate the overall quality of software through various testing methodologies. | L4 |
| 5. Identify potential risks associated with software development. | L3 |

UNIT-I:

Introduction to Software Engineering:

The evolving role of software, Changing Nature of Software, legacy software, software myths.

A Generic View of Process:

Software engineering- A layered technology, a process framework, the capability Maturity Model Integration (CMMI), process patterns, process assessment, personal and team process models.

Process Models:

The Waterfall Model, Incremental Process Model, Evolutionary Process models, specialized process models, unified process.

UNIT-II:

Software Requirements:

Functional and Non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

Requirements Engineering Process:

Feasibility studies, Requirements elicitation and analysis, requirements validation, requirements management.

System Models:

Context Models, Behavioral Models, Data Models, Object Models, structured methods.

UNIT-III:

Design Engineering:

Design Process and Design quality, Design concepts, the design model, pattern based software design

Creating an Architectural Design:

Software architecture, Data Design, Architectural styles and patterns. Architectural Design, assessing alternative architectural designs, mapping data flow into a software architecture.

Modelling Component-Level Design:

Design class-based components, conducting component-level design, object constraint language, design conventional components.

UNIT-IV:**Testing Strategies:**

A strategic approach to software testing, testing strategies for conventional software, Black-Box and White-Box testing. Validation testing, system testing, the art of debugging.

Product Metrics:

Software Quality, Frame work for Product metrics, Metric for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance.

Metrics For Process And Products:

Software Measurement, Metrics for Software Quality.

UNIT-V:**Risk Management:**

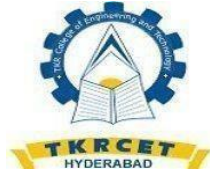
Reactive versus Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement. RMMM, RMMM plan.

Text Books:

1. Software engineering A Practitioner's approach, Roger S Pressman, Sixth Edition McGraw Hill International Edition.
2. Software Engineering: Ian Sommerville, Seventh Edition, Pearson Education.

Reference Books:

1. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India 2010.
2. Software Engineering: A Primer, Waman S. Jawadekar, Tata McGraw Hill, 2008.
3. Software Engineering Foundations, Yingxu Wang, Auerbach Publications 2008.



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

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

PYTHON PROGRAMMING LAB - B54PC6

B.Tech. IV Semester

**L/T/P C
0/0/3 1.5**

Course Objectives:

Write and execute the programs based on operators, functions, simple data structures, basic packages using pythonprogramming constructs.

Course Outcomes:

After completion of course, the student will be able to:

- | | |
|--|----|
| 1. Apply different operators to perform arithmetic, logical, and assignment operations in Python code. | L3 |
| 2. Apply predefined functions and build functions. | L3 |
| 3. Apply python modules and implement data structures to solve various computing problems. | L3 |
| 4. Analyze the data and derive insights. | L4 |

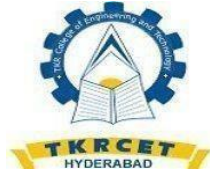
List of programs:

1. Write a python program to print “Hello World”.
2. Running instructions in Interactive interpreter and a Python Script.
3. Write a Python Programming to demonstrate the Indentation.
4. Write a Python program to calculate number of days between two dates.
5. Write a python program that takes 2 numbers as command line arguments and prints its product.
6. Write a Python program to test whether a given letter is a vowel or not.
7. Write a Python program to create a pattern.


```

      *
      * *
      * * *
      * * * *
      * * * * *
      * * * * *
      * * * *
      * * *
      * *
      *
      
```
8. Write a Python program to count the number 6 in a given list.
9. Write a python program to find the sum of the first n positive integers.
10. Write a Python program to calculate the sum of the digits in an integer
11. Write a Python program that prints all the numbers from 0 to 50 except multiples of 10 (10,20,30,40,50)
12. Write a Python program to check if a number is positive, negative or zero.
13. Write a Python program that will accept the base and height of a triangle and compute the area.
14. Write a Python program to compute the greatest common divisor (GCD) of two positive integers.
15. Write a Python program Make a Simple Calculator

16. Write a Python program to count the number of even and odd numbers from a series of numbers.
17. Write a Python function to calculate the factorial of a number (a non-negative integer). The function should accept the number as an argument.
18. Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.
19. Write a Python function that checks whether a passed string is palindrome or not.
20. Write a Python program to get the Fibonacci series between 0 to 50 using recursion.
21. Write a Python program to calculate the value of 'a' to the power 'b' using recursion.
22. Write a Python program to get the factorial of a non-negative integer using recursion.
23. Write a Python program to calculate the length of a string.
24. Write a Python program to count occurrences of a substring in a string.
25. Write a Python program to count and display the vowels of a given text.
26. Write a program to count the numbers of characters in the string and store them in a dictionary data structure.
27. Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.
28. Write a program combine lists that combines these lists into a dictionary.
29. Write a Python program for binary search.
30. Write a Python program to sort a list of elements using the bubble sort algorithm.
31. Write a Python program to sort a list of elements using the quick sort algorithm.
32. Write a Python program to count the frequency of words in a file.
33. Write a Python program to print last n lines of a file.
34. Write a Python program to combine each line from first file with the corresponding line in second file.
35. A string from the user and print string in upper case.
36. Write a Python class named Rectangle constructed by length and width and a method which will compute the area of a rectangle.
37. Solve the following linear equations using scipy library $X+3y+5z=10$, $x+5y+z=8$, $x+3y+8z=3$.
38. Find the determinant for a 2 * 2 matrix using scipy library module.
39. Find the mean and variance for the following data using scipy $[2, 23, 45, 56, 78, 89, 13, 33, 66, 89]$.
40. Draw a bar chart with the following data using matplotlib $\text{Men_mean}=[20, 35, 30, 35, 27]$
 $\text{Women_mean}=[25, 32, 34, 20, 25]$
 $\text{Men_std}=[2, 3, 4, 1, 2]$ $\text{Women_std}=[3, 5, 2, 3, 3]$.
41. Using matplotlib and scipy libraries, apply the following operations on an image.
 - a) Display the image
 - b) crop image
 - c) flip
 - d) rotate
 - e) display the statistical information of the image
 - f) turn upside down.
42. Using matplotlib package, draw histogram to show the results performance of a class in the subject C Programming. Students of first year written the exam c programming and obtained the results. Show their performance with the help of a histogram.



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

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

DATABASE MANAGEMENT SYSTEM LAB - B54PC7

B.Tech. IV Semester

**L/T/P C
0/0/3 1.5**

Course Objective:

Emphasize on designing, developing and querying a database in the context of example database “Roadway travels”.

Course Outcomes:

After learning the contents of this course, the student must be able to:

- | | |
|--|----|
| 1. Design and implement a database schema for given problem. | L3 |
| 2. Apply the normalization techniques for development of application software to Realistic Problems. | L3 |
| 3. Formulate queries using SQL DML/DDL/DCL commands. | L5 |
| 4. Develop application programs using PL/SQL. | L5 |

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)

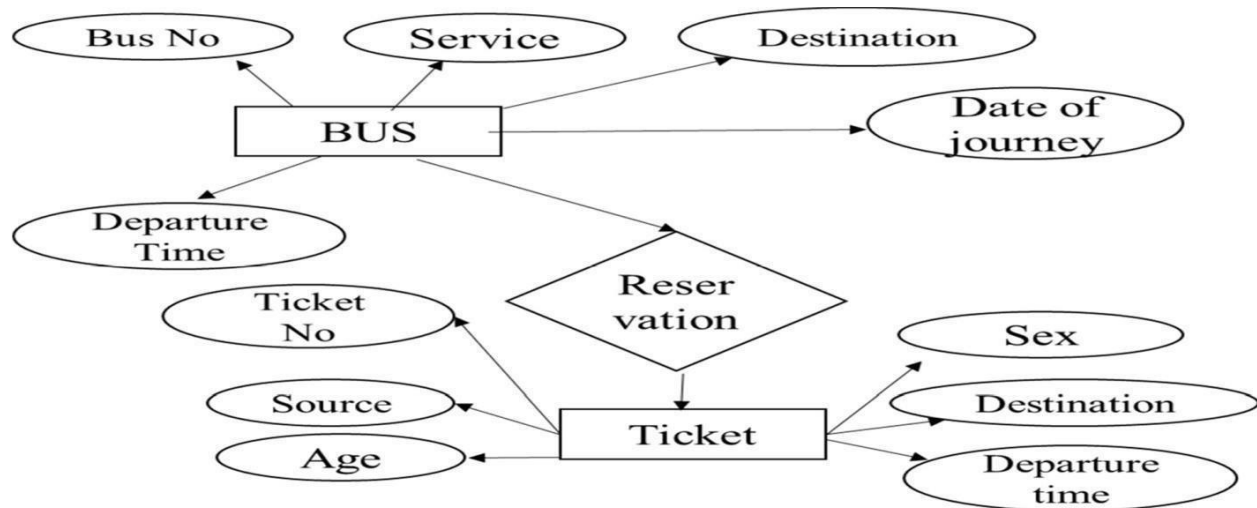
3. Bus_No. (Bus Entity)

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

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

**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	Passport ID	Ticket_ID

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.

Passe nger Nam e	A g e	S e x	Ad dre ss	Passp ortID

Passp ortID	Ticke t_ID

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

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, and Address VARCHAR (50) Not NULL); similarly create all other tables.

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.

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.

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

1. Display the number of days in a week on which the 9W01 bus is available.
2. Find number of tickets booked for each PNR_no using GROUP BY CLAUSE.
3. Find the distinct PNR numbers that are present.
4. 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.

5. 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 ANDELSEIF
5. Write a PL/SQL block for summation of odd numbers using for LOOP.

Experiment 11: 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.TicketNO > 60 THEN SET New.Ticket no = Ticket no; ELSE SET New.Ticketno = 0; END IF; END;

Experiment 12: Procedures:

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:

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 otherthan Integer Status: Char (2) : Yes / No

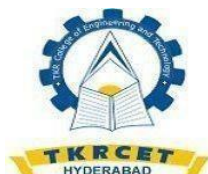
Cancellation

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 othercharacterother 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



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

B.TECH. COMPUTER SCIENCE & ENGINEERING - R18

SOFTWARE ENGINEERING LAB - B54PC8

B.Tech. IV Semester

**L/T/P C
0/0/3 1.5**

Course Objectives:

Understand the software engineering concepts and implementing prototype models.

Course Outcomes:

After learning the contents of this course, the student will be able to:

- | | |
|--|----|
| 1. Utilize software engineering methodologies across various phases of project development, such as requirements analysis, design, implementation, testing, deployment, and maintenance. | L3 |
| 2. Apply open source tools proficiently to implement software engineering methods. | L3 |
| 3. Develop innovative products within startup environments and Synthesize advanced knowledge in software engineering methodologies. | L6 |

Open source Tools: Star UML / UML Graph / Top cased:

Prepare the following documents and develop the software project startup, prototype model, using software engineering methodology for at least two real time scenarios or for the sample experiments.

1. Problem Analysis and Project Planning -Thorough study of the problem – Identify Project scope,Objectives and Infrastructure.
2. Software Requirement Analysis – Describe the individual Phases/modules of the project and Identify deliverables. Identify functional and non-functional requirements.
3. Data Modeling – Use work products – data dictionary.
4. Software Designing – Develop use case diagrams and activity diagrams, build and test class diagrams, sequence diagrams and add interface to class diagrams.
5. Prototype model – Develop the prototype of the product.

List of Sample Experiments:

1. Course Management System (CMS):

A course management system (CMS) is a collection of software tools providing an Online environment for course interactions. A CMS typically includes a variety of online tools and environments, such as:

- 1) An area for faculty posting of class materials such as course syllabus and handouts
- 2) An area for student posting of papers and other assignments
- 3) A grade book where faculty can record grades and each student can view his or her grades
- 4) An integrated email tool allowing participants to send announcement email messages to the entire class or to a subset of the entire class
- 5) A chat tool allowing synchronous communication among class participants
- 6) A threaded discussion board allowing asynchronous communication among participants

In addition, a CMS is typically integrated with other databases in the university so that students

enrolled in a particular course are automatically registered in the CMS as participants in that course. The Course Management System (CMS) is a web application for department Personnel, Academic Senate, and Registrar staff to view, enter, and manage course information formerly submitted via paper. Departments can use CMS to create new course proposals, submit changes for existing courses, and track the progress of proposals as they move through the stages of online approval.

2. Easy Leave:

This project is aimed at developing a web based Leave Management Tool, which is of importance to either an organization or a college.

The Easy Leave is an Intranet based application that can be accessed throughout the organization or a specified group/Dept. This system can be used to automate the workflow of leave applications and their approvals. The periodic crediting of leave is also automated. There are features like notifications, cancellation of leave, automatic approval of leave, report generators etc. in this Tool. Functional components of the project:

There are registered people in the system. Some are approvers. An approver can also be a requestor. In an organization, the hierarchy could be Engineers/Managers/Business Managers/Managing Director Etc. In a college, it could be Lecturer/Professor/Head of the Department/Dean/Principal etc.

Following is a list of functionalities of the system: A person should be able to:

1. login to the system through the first page of the application
2. change the password after logging into the system
3. see his/her eligibility details (like how many days of leave he/she is eligible for etc)
4. query the leave balance
5. see his/her leave history since the time he/she joined the company/college
6. apply for leave, specifying the from and to dates, reason for taking leave, address for communication while on leave and his/her superior's email id
7. see his/her current leave applications and the leave applications that are submitted to him/her for approval or cancellation
8. approve/reject the leave applications that are submitted to him/her
9. withdraw his/her leave application (which has not been approved yet)
10. Cancel his/her leave (which has been already approved). This will need to be approved by his/her Superior
11. get help about the leave system on how to use the different features of the system
12. As soon as a leave application /cancellation request / withdrawal /approval rejection /password - change is made by the person, an automatic email should be sent to the person and his superior giving details about the action
13. The number of days of leave (as per the assumed leave policy) should be automatically credited to everybody and a notification regarding the same be sent to them automatically
14. An automatic leave-approval facility for leave applications which are older than 2 weeks should be there. Notification about the automatic leave approval should be sent to the person as well as his superior

3. E-Bidding:

Auctions are among the latest economic institutions in place. They have been used since antiquity to sell a wide variety of goods, and their basic form has remained unchanged. In this dissertation, we

explore the efficiency of common auctions when values are interdependent the value to a particular bidder may depend on information available only to others-and asymmetric. In this setting, it is well known that sealed-bid auctions do not achieve efficient allocations in general since they do not allow the information held by different bidders to be shared.

Typically, in an auction, say of the kind used to sell art, the auctioneer sets a relatively low initial price. This price is then increased until only one bidder is willing to buy the object, and the exact manner in which this is done varies. In my model a bidder who drops out at some price can “reenter” at a higher price. With the invention of E-commerce technologies over the Internet the opportunity to bid from the comfort of one’s own home has seen a change like never seen before. Within the span of a few short years, what may have begun as an experimental idea has grown to an immensely popular hobby, and in some cases, a means of livelihood, the Auction Patrol gathers tremendous response every day, all day. With the point and click of the mouse, one may bid on an item they may need or just want, and in moments they find that either they are the top bidder or someone else wants it more, and you’re outbid!.

The excitement of an auction all from the comfort of home is a completely different experience. Society cannot seem to escape the criminal element in the physical world, and so it is the same with Auction Patrols. This is one area where in a question can be raised as to how safe Auction Patrols.

Proposed system:

1. To generate the quick reports
2. To make accuracy and efficient calculations
3. To provide proper information briefly
4. To provide data security
5. To provide huge maintenance of records, flexibility of transactions can be completed intime.

4. Electronic Cash counter:

This project is mainly developed for the Account Division of a Banking sector to provide better interface of the entire banking transactions. This system is aimed to give a better out look to the user interfaces and to implement all the banking transactions like:

1. Supply of Account Information
2. New Account Creations
3. Deposits
4. Withdraws
5. Cheque book issues
6. Stop payments
7. Transfer of accounts
8. Report Generations.

Proposed System:

The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach.

1. User friendliness is provided in the application with various controls.
2. The system makes the overall project management much easier and flexible.
3. Readily upload the latest updates, allows user to download the alerts by clicking the URL.
4. There is no risk of data mismanagement at any level while the project development is under process.