T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

COURSE STRUCTURE & SYLLABUS

III SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1.	**BBSM3**	Probability & Statistics	**3**	**0**	**0**	**3**
2.	**B63PC1**	Data Structures	**3**	**0**	**0**	**3**
3.	**B63PC2**	Digital Logic Design	**3**	**0**	**0**	**3**
4.	**B63PC3**	Object Oriented Programming	**3**	**0**	**0**	**3**
5.	**B63PC4**	Introduction to Analytics	**3**	**0**	**0**	**3**
6.	**B63PC5**	Data Structures Lab	**0**	**0**	**3**	**1.5**
7.	**B63PC6**	Object Oriented Programming Lab	**0**	**0**	**3**	**1.5**
8.	**B63PC7**	Digital Logic Design Lab	**0**	**0**	**3**	**1.5**
					Total	**19.5**

IV SEMESTER

S.No	Course Code	Course Title	L	T	P	Credits
1.	**BBSM5**	Discrete Mathematics	**3**	**0**	**0**	3
2.	**B64PC2**	Computer Organization	**3**	**0**	**0**	3
3.	**B64PC3**	Database Management Systems	**3**	**0**	**0**	3
4.	**B64PC4**	Software Engineering	**3**	**0**	**0**	3
5.	**B64PC5**	Formal Language & Automata Theory	**3**	**0**	**0**	3
6.	**B64PC6**	Database Management System Lab	**0**	**0**	**3**	**1.5**
7.	**B64PC7**	Software Engineering Lab	**0**	**0**	**3**	**1.5**
8.	**BITW1**	IT Workshop	**0**	**0**	**3**	**1.5**
Total						**19.5**

T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -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:

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

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- poison 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.

SUGGESTED TEXT/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- 3 edition Wiley-1968.

T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY- R18

DATA STRUCTURES - B63PC1

B.Tech. III Semester **L/T/P/C**
3/0/0/3

COURSE OBJECTIVES:

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

COURSE OUTCOMES:

1. Understand the concept of ADT.
2. Ability to choose appropriate data structures to represent data items in real world problems.
3. Ability to analyses the time and space complexities of algorithms.
4. 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.
5. Able to analyze 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, class templates, Inheritance and Polymorphism.

Basic Concepts - Data objects and Structures, Algorithm Specification-Introduction, Data Abstraction, Performance analysis- time complexity and space complexity, Asymptotic Notation-Big O, Omega and Theta notations, 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, 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.

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.

UNIT-V:

Graphs–Definitions, Terminology, Applications and more definitions, Properties, Graph ADT, Graph Representations- Adjacency matrix, Adjacency lists, Graph Search methods - DFS and BFS. 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**,** 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)

DEPARTMENT OF INFORMATION TECHNOLOGY- R18

DIGITAL LOGIC DESIGN - B63PC2

B.Tech. III Semester **L/T/P/C**
3/0/0/3

COURSE OBJECTIVES:

1. To understand basic number systems, codes and logic gates.
2. To understand the concepts of Boolean algebra.
3. To understand the minimization logic to solve Boolean logic expressions.
4. To understand the design of combinational and sequential circuits.
5. To understand the state reduction methods for Sequential circuits.
6. To understand the basics of various types of memories.

COURSE OUTCOMES

1. Ability to do conversions in Number Systems and implement logic gates.
2. Students can apply Minimization methods to solve Boolean Expressions.
3. Ability to design sequential and combinational circuits.
4. Able to apply State Reduction methods to solve sequential circuits.

UNIT-I:

Digital Systems, Binary Numbers, Number base conversions, Octal, Hexadecimal and other base numbers, complements, signed binary numbers, Floating point number representation, binary codes, Error detection and correction, 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.

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 eifferent code converters and other problems, Binary Adder-Sub tractor, Decimal Adder, Binary Multiplier, Magnitude Comparator, Decoders, Encoders, Multiplexers, Demultiplexers.

UNIT-IV:

Synchronous Sequential Circuits, Latches, Flip-flops, analysis of clocked sequential circuits, Registers, Shift registers, Ripple counters, Synchronous counters, other counters.

UNIT-V:

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

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, 7th 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)

DEPARTMENT OF INFORMATION TECHNOLOGY- R18

OBJECT ORIENTED PROGRAMMING - B63PC3

B.Tech. III Semester **L/T/P/C**
3/0/0/3

COURSE OBJECTIVES:

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

COURSE OUTCOMES

1. Able to understand the use of abstract classes.
2. Able to solve problems using java collection framework and I/O classes.
3. Able to develop multithreaded applications with synchronization.
4. Able to develop applets for web applications.
5. Able to design GUI based applications.

UNIT-I:

Object-oriented thinking-. 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, forms of inheritance- specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance, super uses, using final with inheritance, method overriding, abstract classes, Object class, Polymorphism-ad hoc polymorphism, pure polymorphism.

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, The Console class, Serialization, Enumerations, auto boxing.

UNIT-III:

Exception handling - Fundamentals of exception handling, Exception types, 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, and inter thread communication

UNIT-IV:

The Collections Framework (java.util)- Collections overview, Collection Interfaces, The Collection classes- Array List, Linked List, HashSet, TreeSet, Priority Queue, Accessing a Collection via an Iterator, Using an Iterator, The For-Each alternative, Map Interfaces and Classes, Comparators, Arrays, The Legacy Classes and Interfaces-Dictionary, Hashtable ,Properties, Stack, Vector, String Tokenizer.

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.

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, JTabbed Pane, JScroll Pane, JList, JCombo Box, Swing Menus, Dialogs.

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.
3.

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, Universities Press.
4. Programming in Java, S. Malhotra, S. Chudhary, 2nd edition, Univ. Press., Oxford
5. Java Programming and Object oriented Application Development, R. A. Johnson, Cengage Learning.

T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

INTRODUCTION TO ANALYTICS - B63PC4

B.Tech. III Semester **L/T/P/C**
3/0/0/3

COURSE OBJECTIVES:

1. To introduce the terminology, technology and itsapplications
2. To introduce the concept of Analytics for Business
3. To introduce the tools, technologies & programming languages which is used in day to day analytics cycle.

COURSE OUTCOMES:

1. Able to work on R- language and prioritize the work according to the requirements.
2. Able to summarize the data
3. Able to implement database connectivity.
4. Able to analyze the various regressions.

UNIT-I:
Introduction to Analytics and R programming (NOS 2101):
Introduction to R, RStudio (GUI): R Windows Environment, introduction to various data types, Numeric, Character, date, data frame, array, matrix etc., Reading Datasets, Working with different file types .txt,.csv etc. Outliers, Combining Datasets, R Functions and loops.
Manage your work to meet requirements (NOS 9001)
Understanding Learning objectives, Introduction to work & meeting requirements, Time Management, Work management & prioritization, Quality & Standards Adherence.

UNIT-II:
Summarizing Data & Revisiting Probability (NOS 2101)
Summary Statistics - Summarizing data with R, Probability, Expected, Random, Bivariate Random variables, Probability distribution. Central Limit Theorem etc.
Work effectively with Colleagues (NOS 9002)
Introduction to work effectively, Team Work, Professionalism, Effective Communication skills, etc.

UNIT-III:
SQL using R Introduction to NoSQL, Connecting R to NoSQL databases.Excel and R integration with R connector.

UNIT-IV:
Correlation and Regression Analysis (NOS 9001)
Regression Analysis, Assumptions of OLS Regression, Regression Modelling.Correlation, ANOVA, Forecasting, Heteroscedasticity, Autocorrelation, Introduction to Multiple Regression etc.

UNIT-V:
Understand the Verticals - Engineering, Financial and others (NOS 9002) Understanding systems viz. Engineering Design, Manufacturing, Smart Utilities, Production lines, Automotive, Technology etc.
Understanding Business problems related to various businesses
Requirements Gathering

Gathering all the data related to Business objective

TEXT BOOKS

1. Student’s Handbook for Associate Analytics.

REFERENCE BOOKS

1. Introduction to Probability and Statistics Using R, ISBN: 978-0-557-24979-4, is a textbook written for an undergraduate course in probability and statistics.
2. An Introduction to R, by V enables and Smith and the R Development Core Team. This may be downloaded for free from the R Project website (http://www.r-project.org/, see Manuals). There are plenty of other free references available from the R Project website.
3. Montgomery, Douglas C., and George C. Runger, Applied statistics and probability for engineers. John Wiley & Sons, 2010 http://anson.ucdavis.edu/~azari/sta137/AuNotes.pdf
4. Time Series Analysis and Mining with R,Yanchang Zhao.

T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY- R18

DATA STRUCTURES LAB - B63PC5

B.Tech. III Semester **L/T/P/C**
0/0/3/1.5

COURSE OBJECTIVES:

1. 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.
2. To learn to write C++programs to implement various sorting and searching algorithms

COURSE OUTCOMES:

1. Able to identify the appropriate data structures and algorithms for solving real world problems.
2. Able to implement various kinds of searching and sorting techniques.
3. Able to implement data structures such as stacks, queues, Search trees, and hash tables to solve various computing problems.

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 inorder.

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.

REFERENCE BOOKS

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

T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY- R18

OBJECT ORIENTED PROGRAMMING LAB - B63PC6

B.Tech. III Semester **L/T/P/C**
0/0/3/1.5

COURSE OBJECTIVES:

1. To write programs using abstract classes.
2. To write programs for solving real world problems using java collection frame work.
3. To write multithreaded programs.
4. To write GUI programs using swing controls in Java.
5. To introduce java compiler and eclipse platform.

COURSE OUTCOMES:

1. Able to write programs for solving real world problems using java collection frame work.
2. Able to write programs using abstract classes.
3. Design and develop programs using objects and inheritance in Java language.
4. Able to write multithreaded programs.
5. 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.

Programs:

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 Num1and 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 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. a) Create an applet program for Menu demonstration. Menu bar should containFile, Edit, View and its submenus.
    b) Create an applet program for key events it should recognize normal as well as special keys & should be displayed on the panel.

REFERENCE BOOKS

1. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition, Pearson education.
2. Thinking in Java, Bruce Eckel, Pearson Education.
3. Java Programming, D. S. Malik and P. S. Nair, Cengage Learning.

T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

DIGITAL LOGIC DESIGN LAB - B63PC7

B.Tech. III Semester **L/T/P/C**
0/0/3/1.5

COURSE OBJECTIVES:

1. This course provides the basic knowledge of digital logic levels and application of knowledge to understand digital electronics circuits.
2. To prepare students to perform the analysis and design of various digital electronic circuits.

COURSE OUTCOMES:

1. To understand the logic gates.
2. Able to design adders and subtractions using logic gates
3. Able to design various combinational circuits using logic gates
4. Understand the concepts of asynchronous and synchronous counter.

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

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.

T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY - R18

DISCRETE MATHEMATICS - BBSM5

B.Tech. IV Semester **L/T/P/C**
3/0/0/3

COURSE OBJECTIVES:

Throughout the course students will be expected to demonstrate their understanding of Discrete Mathematics by being able to do each of the following.

1. Use mathematically correct terminology and notation
2. Construct correct direct and indirect proofs.
3. Use division into cases in a proof.
4. Use counter examples.
5. Apply logical reasoning to solve a variety of problems.

COURSE OUTCOMES:

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

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 colouring.

TEXT/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 DP Mohapatra- elements of discrete mathematics A Computer oriented approach 3rdedition by- Tata McGraw-Hill.
6. J.P. Tremblay and R. Manohar- Discrete mathematical structure and it's application to computer science" TMG editiontata mcgaw-Hill.
7. Norman L. Biggs- Discrete mathematics 2nd edition oxford university press schaum's outlines series Seymour Lipchitz- Marc Lipson- Discrete Mathematics- Tata McGraw-Hill.
8. Discrete Mathematics – TataMcGraw-Hill.

T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

COMPUTER ORGANIZATION - B64PC2

B.Tech. IV Semester

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

COURSE OBJECTIVES:

1. To understand basic components of computers.
2. To understand the I/O organization.
3. To understand the Memory Organization.
4. To understand the Architecture of 8086.
5. To understand the instruction formats and representation of data at the machine level and how computations are performed.

COURSE OUTCOMES:

1. Able to understand the basic components and the design of CPU,ALU and Control Unit.
2. Ability to understand Memory hierarchy and its impact on computer cost/performance.
3. Ability to use instruction sets and formats of 8086.
4. Able to write assembly language programs to solve problems.
5. Ability to understand the Design of a processor.

UNIT-I:

Basic Computer Organization - Functions of CPU, I/O Units, Memory Instruction: Instruction Formats - One address, two addresses, zero addresses and three addresses and comparison; addressing modes with numeric examples: Program Control - Status bit conditions, conditional branch instructions, Program Interrupts: Types of Interrupts.

UNIT-II:

Input-Output Organizations - I/O Interface, I/O Bus and Interface modules: I/O Vs memory Bus, Isolated Vs Memory-Mapped I/O, Asynchronous data Transfer- Strobe Control, Hand Shaking: Asynchronous Serial transfer- Asynchronous Communication interface, Modes of transfer programmed I/O, Interrupt Initiated I/O, DMA; DMA Controller, DMA Transfer, IOP-CPU-IOP Communication, Intel *).

UNIT-III:

Memory Organizations: Memory hierarchy, Main Memory, RAM, ROM Chips, Memory Address Map, Memory Connection to CPU, associate memory, Cache Memory, Data Cache, Instruction cache, Miss and Hit ratio, Access time associative, set associative, mapping, waiting into cache, Introduction to virtual memory.

UNIT-IV:

8086 CPU Pin Diagram- Special functions of general purpose registers. Segment register, 8086 Flag register, Addressing modes of 8086.

UNIT-V:

8086-Instruction formats: assembly Language Programs involving branch & Call instructions, sorting, evaluation of arithmetic expressions.

TEXT BOOKS

1. Computer System Architecture: Moris Mano (UNIT - 1, 2, 3).
2. Advanced Micro Processor and Peripherals - Hall/ A K Ray (UNIT - 4, 5).

REFERENCE BOOKS

1. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.
2. Structured Computer Organization and Design - Andrew S. Tanenbaum, 4th Edition PHI/Pearson.
3. Fundamentals or Computer Organization and Design - Sivaraama Dandamudi Springer Int. Edition.

T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

DATABASE MANAGEMENT SYSTEMS - B64PC3

B.Tech. IV Semester **L/T/P/C**
3/0/0/3

COURSE OBJECTIVES:

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

COURSE OUTCOMES:

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

UNIT-I:

Introduction: 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 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.

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 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.

UNIT-IV:

Transaction Management: 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, Multisession Schemes.
Recovery System-Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm, Buffer Management, Failure with loss of nonvolatile storage, Early Lock Release and Logical Undo Operations, Remote Backup systems.

UNIT-V:

Storage and Indexing: Overview of Storage and Indexing: Data on External Storage, File Organization and Indexing, Index Data Structures, 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.

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 1, 6th edition.

REFERENCE BOOKS

1. Database Systems, 6th edition, R Elmasri, ShamkantB.Navathe, Pearson Eduction.
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)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

SOFTWARE ENGINEERING - B64PC4

B.Tech. IV Semester **L/T/P/C**
3/0/0/3

COURSE OBJECTIVES:

1. To understanding of software process models such as waterfall and evolutionary models.
2. To understanding of software requirements and SRS document.
3. To understanding of different software architectural styles.
4. To understanding of software testing approaches such as unit testing and integration testing
5. To understanding on quality control and how to ensure good quality software.

COURSE OUTCOMES:

1. Ability to identify the minimum requirements for the development of application.
2. Ability to develop, maintain, efficient, reliable and cost effective software solutions
3. Ability to critically thinking and evaluate assumptions and arguments.

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: perspective and specialized process models, COCOMO model

UNIT-II:

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, 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, data dictionary.

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, mapping data flow into a software architecture.

Modeling component-level design: Designing class-based components,
Designing conventional components.

Performing User interface design: Golden rules, interface analysis, interface design steps.

UNIT-IV:

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

Product metrics: Software Quality, Framework for Product metrics, Metrics 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 vs Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

TEXT BOOKS:

1. Software engineering 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, PankajJalote, Wiley India, 2010.
2. SoftwareEngineering:APrimer,WamanSJawadekar,TataMcGraw-Hill,2008
3. FundamentalsofSoftwareEngineering,RajibMall,PHI,2005
4. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
5. Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.
6. Software Engineering2: Specification of systems and languages, Diner Bjorner, Springer International edition 2006.
7. Software Engineering Foundations, Yingxu Wang, Auerbach rd Publications, 2008.
8. Software Engineering Principles and Practice, Hans Van Vliet, 3 edition, John Wiley& Sons Ltd.
9. Software Engineering3: Domains, Requirements, and Software Design, D. Bjorner, Springer International Edition.

T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

FORMAL LANGUAGES & AUTOMATA THEORY - B64PC5

B.Tech. IV Semester **L/T/P/C**
3/0/0/3

COURSE OBJECTIVES:

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

COURSE OUTCOMES:

1. Able to understand the concept of abstract machines and their power to recognize the languages.
2. Able to employ finite state machines for modeling and solving computing problems.
3. Able to design context free grammars for formal languages.
4. Able to design Turing Machines.
5. Able to distinguish between decidability and un-decidability.
6. 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.

UNIT-II:

Regular Expressions, Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, 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.

UNIT-IV:

Normal Forms for Context- Free Grammars - Chomsky normal form, 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 **Introduction to Turing Machines-**Problems 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**:**, Some NP- complete problems, The Classes P and NP, An NP- Complete Problem.

TEXT BOOKS:

1. Introduction to Automata Theory, Languages, and Computation, 3rd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Introduction to the Theory of Computation, Michael Sipser, 3rd 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 Chandrashekaran, 2nd edition, PHI.

T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

DATABASE MANAGEMENT SYSTEM LAB - B64PC6

B.Tech. IV Semester **L/T/P/C**
0/0/3/1.5

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:

1. Formulate queries using SQL DML/DDL/DCL commands.
2. Analyze the normalization techniques
3. Design and implement a database schema for given problem.
4. Develop 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

Analyze 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. Passenger_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, Passenger_ID. 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, Passenger_ID, Ticket_ID.You can do the second and third normal forms if required.

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:

(UNDERSTANDING) SIMPLE

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,’hyderabd’,’Banglore’); Insert into Bus values (23,’hyderabd’,’Kolkata’); Insert into Bus values (45,’Tirupathi,’Banglore’); Insert into Bus values (34,’hyderabd’,’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 thepassengers.
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: Aggregate functions

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.

Experiment 9: Querying using clauses

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. Create a PL/SQL block for Addition of Two Numbers
2. Develop a PL/SQL block for IF Condition
3. Design a PL/SQL block for IF and elsecondition
4. Develop a PL/SQL block for greatest of three numbers using IF AND ELSEIF
5. Construct 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_ q +
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, Pearsoneducation.
2. Oracle PL/SQL, B. Rosenzweig and E. Silvestrova, Pearson education.
3. S740QL & 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)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

SOFTWARE ENGINEERING LAB - B64PC7

B.Tech. IV Semester **L/T/P/C**
0/0/3/1.5

COURSE OBJECTIVES:

To understand the software engineering methodologies involved in the phases for project development.

1. To gain knowledge about open source tools used for implementing software engineering methods.
2. To exercise developing product-startups implementing software engineering methods.
3. Open source Tools: StarUML / UMLGraph / Topcased

COURSE OUTCOMES:

1. Ability to translate end user requirements using tools into system and software requirements
2. Ability to generate a high level design of the system from the software requirements
3. Ability to experience and/or awareness of testing problems and will be able to develop a simple testing report.

Prepare the following documents and develop the software project startup, prototype model, using software engineering methodology for atleast 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. Proto type model– Develop the proto type of the product.

The SRS and proto type model should be submitted for end semester Examination.

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. ACMS 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 An area for student posting of papers and other assignments

1. A grade book where faculty can record grades and each student can view his or her grades

2. An integrated email tool allowing participants to send announcement email messages to the entire class or to a subset of the entire class

4. A chat tool allowing synchronous communication among class participants

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 generator set 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 In a college; it could be Lecturer/ Professor/ Head of the Department/ Dean/ Principal etc.

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

1. Login to the system through the first page of theapplication
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 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 see his/her current leave applications and the leave applications that are submitted to him/her for approval or cancellation approve/reject the leave applications that are submitted to him/her withdraw his/her leave application(which has not Been approved yet)
6. Cancel his/her leave (which has been already approved). This will need to be approved by his/her Superior
7. Get help about the leave system on how to use the different features of the system
8. As soon as a leave application/cancellation request/withdrawal/approval/rejection
9. password-change is made by the person ,an automatic email should be sent to the person and his superior giving details about theaction
10. 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
11. 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 a symmetric. 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 are natively 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

To generate the quick reports

To make accuracy and efficient calculations to provide proper information briefly

To provide data security

To provide huge maintenance of records Flexibility of transactions can be completed in time

4. Electronic Cash counter

This project tis mainly developed for the Account Division of a Banking sector to provide better interface of then tire banking transactions. This system is a imed 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.

5. It provides high level of security with different level of authentication.

T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY- R18

IT WORKSHOP - BITW1

B.Tech. IV Semester **L/T/P/C**
0/0/3/1.5

COURSE OBJECTIVES

1. 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.
2. 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.
3. 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.
4. 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:

1. Apply knowledge for computer assembling and software installation.
2. Ability how to solve the trouble shooting problems.
3. Apply the tools for preparation of PPT, Documentation and budget sheet etc.
4. Create standard documents and research documents using Latex.
5. 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.

Productivitytools 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).