

MAHAKAUSHAL

University



**Faculty of & Computer Science
Syllabus & Prescribed Books
Subjest-M.Sc.IT**

**M.Sc. IT Semester Examination
2022-2023
I, II,III& IV Semester**

MSc Information Technology 1st semester**MMIT0101-T Computer and Communication Fundamentals**

Unit No.	Topics
Unit 1	Computer Organization: Digital and Analog computers, Major components of a digital computer, Memory addressing capability of a CPU, Word length of a computer, Processing speed of a CPU, Definitions of Hardware, Software and Firmware., Smart and Intelligent terminals. Binary Systems: Digital Systems, Binary Numbers, Number Base Conversions, Octal and Hexadecimal Numbers, Complements, Signed Binary Numbers, Binary Codes: BCD code, Gray Code, ASCII code, Excess 3 Code, Error detecting Code. Computer Arithmetic: Binary representation of Negative Integers using 2's complement and Signed magnitude representation, Fixed point Arithmetic operations on Positive and Signed (Negative) Integers like addition, subtraction, multiplication, Booth algorithm for multiplication, and bit pair (fast) multiplication. Division of positive and negative binary numbers.
Unit 2	Boolean Algebra and Logic Gates: Basic Definitions, Basic Theorems and properties of Boolean Algebra, Boolean Functions, Canonical and standard forms, Other Logic operations, Digital Logic gates, Integrated Circuits. Gate-Level Minimization: The K-Map Method, 3 and 4 variable K-Map, Product of sums simplification, Sum of Products simplification, Don't care conditions, NAND and NOR implementations, Exclusive-OR function.
Unit 3	Combinational Logic: Combinational Circuits, Analysis Procedure, Design Procedure, Binary half adder, binary full adder, binary full subtractor, binary parallel adder, carry propagation delay and Propagation delay calculation of various digital circuits. Fast adder, Decimal Adder, Binary multiplier, magnitude comparator, Parity generator, seven segment display, BCD to excess three code converter, Decoders, Encoders, Multiplexes, and Demultiplexers Synchronous Sequential logic: Sequential circuits, Latches, Flip Flops: SR, D, JK, T. Master Slave JK Flip flop. Characteristic equations and Excitation tables of flip-flops. Analysis of clocked sequential circuits: State diagrams, State equations for D, JK and T Flip flops. State reduction methods using all Flip Flops. Mealy and Moore Models.
Unit 4	Shift Registers- Serial in Serial out, Serial in Parallel out, Parallel in Serial out and Parallel in Parallel out. Designing of Asynchronous (Ripple) Counters, Design of Synchronous Counters. Signal characteristics: frequency, phase and amplitude, Types of communication with and among computers, Characteristics of communication channels, allocation of channel, signal power, amplitude and frequency modulation, space division multiplexing, frequency division multiplexing, Time division multiplexing.
Unit 5	Physical communication media: bounded media and unbounded media, Infrared technology, public switched telephone networks, digital modulation: ASK, PSK and FSK, half duplex and full duplex transmission, asynchronous and synchronous transmission. Need for computer communication networks, internet and world wide web: E-mail, File transfer, Remote login. Networking topologies. LAN, MAN, WAN, Ethernet LAN. Layer 2 and layer 3 switches, Wireless LAN. Switch hub and router. Wireless LAN, interconnecting networks, IP address and IP data grams, internet and intranet; inter security, virtual private networks, future of internet technology.

1. Text Book :

Digital Design by M. Morris Mano.

Publication: PHI Eastern economy edition (Year-2001)

2. Reference books :

1. Data communications and networking By A. Forouzan

Publication: TMH Third edition (Year- 2004)

2. Computer Fundamentals – Architecture and Organization By B. Ram.

3. Computer networks by Andrew Tanenbaum

Publication: PHI Fourth edition (Year- 2003)

4. Computer organization and architecture by William Stallings.

5. Publication : PHI Fifth edition (Year – 1999)



MAHAKAUSHAL UNIVERSITY
JABALPUR

ज्ञानं महाकौशलम्

Class Assignments:

1. Write the various units of CPU. Describe the working of each unit.
2. Write the definition of hardware, software and firmware.
3. What you mean by word length of a computer.
4. If address bus for a CPU is 20 bits wide, then calculate the size of memory that can be attached to it.
5. If data bus for a CPU is 16 bit then what will be its word length.
6. What are the major factors that decide the processing speed of a CPU?
7. Write the first 20 decimal digits in base 4.
8. Write the first 20 decimal digits in base 3.
9. Add and multiply the following numbers in the given base without converting to decimal.
 - I. $(1230)_4$ and $(33)_4$
 - II. $(130)_5$ and $(34)_5$
 - III. $(230)_6$ and $(54)_6$
 - IV. $(130.4)_5$ and $(34.4)_5$
10. Write the first 100 decimal digits into binary.
11. Convert the following numbers into binary.
 - I. 123.56 II. 456.75
 - III. 345.9 IV. 890.9
 - V. 567.9 VI. 668.7
12. Converts the following numbers into decimal.
 - I. 10101010 II. 101010110011
 - III. 10110101.1111 IV. 101010111.1101
13. Perform the following conversion , without converting into decimal :
 - I. $(3674)_8$ to $()_{16}$ to $()_2$
 - II. $(1001010101010)_2$ to $()_{16}$ to $()_8$
 - III. $(AC4)_{16}$ to $()_2$ to $()_8$
14. Represent the following numbers into sign magnitude representation.
 - I. -11 II. 15 III. -15 IV. 7 V. -32
15. Represent the following numbers into sign two's complement representation.
 - I. -11 II. 15 III. -15 IV. -32
16. Perform the M-N and N-M using two's complement method.
 - I. M= 101100101 and N = 111000110
 - II. M =101 and N = 110110
 - III. M= 45 and N =90
17. Perform the multiplication of following numbers using two's complement method
 - I. $16 * -6$
 - II. $23 * -9$
 - III. $-4 * -6$
18. Perform the multiplication of following numbers using booth algorithm.
 - I. $12 * 5$ II. $30 * -5$ III. $-13 * 25$
19. Simplify the following Boolean functions to minimum number of literals
 - I. $AB+AB'+C'+ABC$
 - II. $ABC+ABC'+AB$
 - III. $AC+BC+ABC+BC'$
 - IV. $A'CD + A'C'D' + A'B'C'D + ABC' + ABCD + A'B'C'D'$
20. Simplify the following functions using karnaugh map.

- I. $F = (1, 4, 7, 8)$
 - II. $F = (3, 4, 7)$
 - III. $F = (0, 1, 2, 4, 7, 8, 10, 15)$
 - IV. $F = (1, 4, 7, 8, 10)$ and $D = (2, 11, 12)$
 - V. $F = \square(1, 2, 3, 4, 7, 8, 9, 11, 12, 15)$
21. Implement the following function using AND and OR gate.
- I. $F = \square(1, 4, 7, 8)$
 - II. $F = \square(3, 4, 7)$
 - III. $F = \square(0, 1, 2, 4, 7, 8, 10, 15)$
 - IV. $F = \square(1, 2, 3, 4, 7, 8, 9, 11, 12, 15)$
22. Implement the following function using only NOR gate.
- I. $AB + AB' + C' + ABC$
 - II. $ABC + ABC' + AB$
 - III. $AC + BC + ABC + BC'$
 - IV. $ABC' + BC + AB$
23. Implement the following function using only NOR gate.
- I. $AB + AB' + C' + ABC$
 - II. $ABC + ABC' + AB$
 - III. $AC + BC + ABC + BC'$
 - IV. $ABC' + BC + AB$
24. Design a combinational circuit that accepts a three-bit number and generates an output binary number equal to the square of the input number.
25. Design a combinational circuit that accept BCD values and generate cube of that number.
26. Design a combinational circuit that generates 540321 weighted code for a BCD input.
27. Design a combinational circuit with four input lines that represent a decimal digit in BCD and four output lines that generate the 9's complement of the input number.
28. Design a 4-bit circuit that generates booth multiplier for given multiplier.
29. A combinational circuit is defined by the following two functions.
- I. $F_1 = x'y' + xyz'$
 - II. $F_2 = x' + y$
 - III. $F_3 = xy + x'y'$
30. Design the circuit with a decoder and external gates.
31. Design an even parity generator.
32. Design 3 to 8 decoder by using two 2 to 4 decoder.
33. Design 4 to 16 decoder by using two 3 to 8 decoder.
34. Design 5 to 32 decoder by using four 3 to 8 decoder.
35. Derive the excitation table of all flip-flops with their truth table.
36. Design a synchronous down counter.

MSc Information Technology 1st semester
MMIT0102-T Programming and Problem Solving Using C

Unit No.	Topics
Unit 1	Introduction to Computer based Problem Solving, Classification of programming languages, Programming Environment {Assemblers, compilers, interpreters, linkers, and loaders}. Introduction to Computer based Problem Solving, Features of good programs – Structured program, modular program, fundamentals of programming. Classification of programming languages: low-level language, middle-level language, high -level language along with examples and differences. Programming Concepts with Flowcharts and algorithms, How to make flowchart: Introduction of the symbols of flowchart and algorithm. Importance of Flowchart and algorithms. Solving and discussing problems with flowchart and algorithm. Class assignment on flowchart. Developing and debugging flowcharts for Programming Problem.
Unit 2	Overview of C: C character set, identifiers and keywords. Data types (description of different types of data types along with examples.), Coding style (first program of C). Format Specifiers associated with each data type. Constants and Variables: Some examples of constants and variables. Expressions and Operators: Instructions, declaration, arithmetic, unary, relational, logical, assignment and conditional operators. Basic Input/ Output functions like scanf() and printf() along with examples. Introduction of Formatting Characters and how to use them along with examples. Sequence Control Structures, Decision Control Structures (if-else and nested if-else) along with examples. Some exercises based on decision control statements. Loop Control Structures: Introduction of loops like for loop and some examples of —for loop. Examples of do-while loop, while loop, break, continue. Explanation and examples of Case Control Structures. Nesting of loops i.e. loops in a loop along with examples. Introduction of Arrays: Array initialization, bound checking. Programs using 1D Array. Advantages and disadvantages of Arrays. 2D and multidimensional array and programs based on it.
Unit 3	Introduction of Pointers: Pointers declaration, pointer arithmetic and operation. Some programs based on pointers. Concept of Functions: prototype and parameter passing and some programs based on it. Advantages/pitfalls of pointers. Dynamic Memory allocation: malloc() and calloc() functions. Pointers and their Applications: String Handling, Matrices Operations. Call by reference and Call by value. String Handling: Pointers and strings, standard library string functions, pointer to array of characters vs. string and its limitation. Matrices Operations along with examples. Introduction of Storage classes (register, static, auto and extern) and their differences. Introduction of Structures: Structure declaration, program and application. Introduction of Union: Union declaration, program and application. Difference between Structure and Union. Some standard Functions like gets(), goto(), puts(), getchar(), putchar(), getch() etc.
Unit 4	Introduction of user defined functions along with examples. Functions with arguments, without arguments along with the examples. Communication between functions (how one function calls the other functions). Some exercise based on functions. Arguments to main: Introduction of argc and argv. Enumerations and bit fields and program based on it. Introduction of bit wise operators along with examples. Introduction of Recursion and program based on it. Concept of typedef and its programming.
Unit 5	Concept of Pre-Processors: macro expansion, file inclusion Conditional compilation. Macros with argument and macro versus function. Concept of local and global variables. Header file creation. Introduction to file handling: Different operations on file like read, write and append. File Creation and programs based on file creation, reading and merging. Advanced pointer. Other Programming Techniques.

1. Text Book :

Let us C, Yashwant Kanetkar, BPB Publications.

2. Reference books :

1. B.W. Kernighan & D.M. Ritchie, "The C Programming Language", Prentice Hall of India.
2. Ashok N. Kamthane, —Programming with ANSI and Turbo Cl, Pearson Education
3. Ashok N. Kamthane et. al., Computer Programming and IT (for RTU), Pearson Education, 2011

Class Assignments:-

1. Write a C program to display "Hello Computer" on the screen.
2. Write a C program to display Your Name, Address and City in different lines.
3. Write a C program to find the area of a circle using the formula: $\text{Area} = \text{PI} * r^2$
4. Write a C program to find the area and volume of sphere. Formulas are: $\text{Area} = 4 * \text{PI} * R * R$
 $\text{Volume} = 4/3 * \text{PI} * R * R * R$.
5. Write a C program to print the multiply value of two accepted numbers.
6. Write a C program to read in a three digit number produce following output (assuming that the input is 347)
3 hundreds 4 tens 7 units
7. Write a C program to read in two integer values and display one as a percentage of the other. Typically your output should look like 20 is 50.00% of 40 assuming that the input numbers were 20 and 40. Display the percentage correct to 2 decimal places.
8. Write a C program to find out whether the character presses through the keyboard is a digit or not (using conditional operator).
9. Write a C program to swap variable values of i and j.
10. Write a C program to find the maximum from given three nos.
11. Write a C program to find that the accepted no. is Negative, Positive or Zero.
12. Write a program which reads two integer values. If the first is lesser print the message up. If the second is lesser, print the message *down* if they are equal, print the message *equal* if there is an error reading the data, print a message containing the word *Error*.
13. Write a C program that prints the given three integers in ascending order using if –else.
14. Given as input three integers representing a date as day, month, year, print the number day, month and year for the next day's date. Typical input: —28 2 1992| Typical output: —Date following 28:02:1992 is 29:02:1992|.
15. Write a C program for calculator designing using switch /case loop.
16. Write a C program to convert decimal to binary.
17. Write a C program to convert decimal to hexadecimal.
18. Write a C program to display first 25 Fibonacci nos.
19. Write a C program to display first 100 prime nos.
20. Write a C program to find factorial of accepted no.
21. Write a C program to find the sum of digits of accepted no.
22. Write a C program to print the accepted no and its reverse no.
23. Write a C program to print all the factors of accepted no.
24. Write a C program to find HCF of two given numbers.
25. Write a C program to find LCM of two given numbers.
26. Write a C program to find all the prime numbers between two given numbers.
27. Write C programs to print the terms of each of the following series:
 - i. $\text{Sin}(x)$ ii. $\text{Cos}(x)$ iii. $\text{Log}(1+x)$ iv. $\text{log}(1-x)$ v. e^x vi. e^{-x}

28. Write a C program to print the sum of series.(will be given in class)

29. Display the following output on screen (assuming the value for input parameter n=5) :

a. * ** *** **** *****	b. 1 12 123 1234 12345	c. AA B ABC ABCD ABCDE	d. 1 23 345 4567 56789	e. 1 23 456 78910 101112131415
f. ***** **** *** ** *	g. ABCDE ABCD ABC AB A	h. * *** **** ***** *****	i. 1 123 12345 1234567 123456789	j. 1 121 12321 1234321 123454321
k. 1 01 101 0101 10101	l. 1 22 333 4444 55555	m. A AB ABC AB A	n. ABCDEDCBA ABCD DCBA ABC CBA AB BA A A	o. 1 121 12321 1234321 123454321

MMIT0102-P Programming and Problem Solving Using C LAB

1. Linked List Implementation:

- Implement a singly linked list in C.
- Write functions to insert elements at the beginning, end, and at a specific position in the linked list.
- Implement functions to delete elements from the beginning, end, and at a specific position.
- Write a function to display the contents of the linked list.

2. Stack Implementation:

- Implement a stack data structure using an array or linked list.
- Write functions to perform stack operations like push (insert), pop (delete), and display.
- Implement functions to check if the stack is empty or full.
- Demonstrate the application of a stack in solving problems such as parentheses matching or infix to postfix conversion.

3. Queue Implementation:

- Implement a queue data structure using an array or linked list.
- Write functions to perform queue operations like enqueue (insert) and dequeue (delete).
- Implement functions to check if the queue is empty or full.
- Use queues to solve problems like breadth-first search (BFS) traversal of a graph or simulating a printer queue.

4. Binary Search Tree (BST) Implementation:

- Implement a binary search tree in C.
- Write functions to perform BST operations like insertion, deletion, and searching.
- Implement functions to perform traversals: inorder, preorder, and postorder.
- Demonstrate the application of BST in operations like searching, sorting, or finding the minimum/maximum element.

5. Hash Table Implementation:

- a. Implement a hash table data structure in C using separate chaining or open addressing.
- b. Write functions to perform hash table operations like insertion, deletion, and searching.
- c. Implement collision resolution techniques like linear probing or chaining.
- d. Show the application of hash tables in tasks like dictionary implementation or duplicate detection.

6. Graph Representation and Traversal:

- a. Implement graph representation using adjacency matrix or adjacency list.
- b. Write functions to perform graph traversal algorithms like depth-first search (DFS) and breadth-first search (BFS).
- c. Demonstrate the application of graph traversal in tasks like finding connected components or shortest path.

7. Dynamic Programming Problems:

- a. Solve classic dynamic programming problems like Fibonacci series, coin change problem, or knapsack problem using C.
- b. Implement memoization or tabulation techniques for optimizing solutions.
- c. Analyze and compare the time complexity of recursive vs. dynamic programming solutions.

8. Sorting Algorithms Implementation:

- a. Implement various sorting algorithms such as bubble sort, selection sort, insertion sort, merge sort, and quick sort in C.
- b. Compare the performance of these sorting algorithms for different input sizes and input types.
- c. Analyze and understand the time complexity and stability of each sorting algorithm.

9. String Manipulation:

- a. Write C programs to perform string operations like string concatenation, substring extraction, string reversal, and string comparison.
- b. Implement string searching algorithms like brute force and Knuth-Morris-Pratt (KMP) algorithms.
- c. Solve problems related to pattern matching or text processing using string manipulation techniques.

10. File Handling and I/O Operations:

- a. Write C programs to perform file handling operations like reading from and writing to files.
- b. Implement programs to manipulate files, such as copying files, merging files, or searching for specific content within files.
- c. Demonstrate error handling and robust file I/O techniques.

MSc Information Technology 1st semester
MMIT0103-T Operating Systems

Unit No.	Topics
Unit 1	Introduction: Evolution of operating systems, operating system concepts Processor Management: Concepts, Algorithms for batch processing Processor Management Algorithms for time sharing operating systems, Introduction to real time systems.
Unit 2	Memory Management: Concepts, single user memory management. Partition memory allocation Virtual memory management using paging and segmentation techniques.
Unit 3	Concurrent Processes: Mutual exclusion and synchronization Techniques of inter process Deadlock handling.
Unit 4	File Management: Operations on a file, structure of a file System Free block list, keeping track of blocks allocated to a file, directory structure, sharing and protection of files, file system Reliability, Unix file system Device Management: Goals of input/output software design, Structure of device hardware and software.
Unit 5	Layers of I/O software, structure of device drivers, Disk driver, disk arm scheduling algorithms, terminal driver, clock driver etc. Introduction to network and distributed operating systems.

1. Text Book :

Operating System Concepts, Addison Wesley, 4th Edition, A. Silberschatz and P. Galvin. 1994. Reference books :

2. Reference books :

1. Design of Unix operating system, Bach M., Pearson Education.
2. Operating systems, 4th Edition, William Stallings, Pearson Education, 2003.

MMIT0103-P Operating Systems LAB

Operating Systems lab:-

1. Setup Environment:

- Set up a simulated environment where processes can be created, scheduled, and executed.
- Define parameters such as CPU burst time, arrival time, priority, etc., for each process.

2. Implement Scheduling Algorithms:

- Implement various process scheduling algorithms such as:
 - First Come First Serve (FCFS)
 - Shortest Job First (SJF) - both preemptive and non-preemptive
 - Round Robin (RR) with varying time slices
 - Priority Scheduling - both preemptive and non-preemptive
 - Multilevel Queue Scheduling
 - Multilevel Feedback Queue Scheduling

3. Simulation Execution:

- Simulate the execution of a set of processes using each scheduling algorithm.
- Keep track of metrics such as average waiting time, turnaround time, response time, etc., for each algorithm.

4. Visualization:

- Visualize the scheduling process using Gantt charts for each algorithm.
- Display the timeline of when processes are scheduled and executed.

5. Performance Analysis:

- Compare the performance of different scheduling algorithms based on metrics like CPU utilization, throughput, and fairness.
- Analyze how different parameters (e.g., quantum time in RR, priority levels in priority scheduling) affect performance.

6. Interactive Simulation:

- Provide an interactive interface where users can input process details and choose scheduling algorithms to observe their behavior.
- Allow users to experiment with different parameters and observe the impact on scheduling decisions.

7. Real-time Scheduling:

- Implement real-time scheduling algorithms like Rate Monotonic Scheduling (RMS) or Earliest Deadline First (EDF).
- Simulate real-time task execution and analyze their deadlines and completion times.

8. Advanced Features:

- Introduce additional features like I/O bursts, blocking, and context switching overhead into the simulation.
- Implement algorithms that handle these features effectively (e.g., Shortest Remaining Time First for processes with I/O bursts).

9. Optimization and Heuristics:

- Experiment with optimization techniques or heuristics to improve scheduling efficiency.
- Implement algorithms that dynamically adjust scheduling parameters based on workload characteristics.

10. Documentation and Analysis:

- Document the design, implementation, and results of each scheduling algorithm.
- Provide analysis and insights into the strengths and weaknesses of each algorithm under different scenarios.

MSc Information Technology 1st semester
MMIT104-T Discrete Structures

Unit No.	.Topics
Unit 1	<p>The Foundations: Logic, Sets and Functions: Introduction to set theory, mathematical logic, prepositions, propositional equivalences, predicates and quantifiers. Importance of Quantifiers. The Foundations: Logic, Sets and Functions: Sets, set operations, fuzzy sets, functions, functions for computer science, sequences and summations.</p> <p>Mathematical reasoning: Introduction to Methods of proof, mathematical induction. Use of mathematical induction to solve different problems. Importance of recursions in computer science, scope of recursions, Recursive definitions, recursive algorithms.</p>
Unit 2	<p>Combinatorics: The basics of counting, The sum rule, The product rule, The Pigeonhole Principle, Permutations with repetitions, Permutations without repetitions, Circular Permutations. Applications of combinations. Applications of Combinatorics to solve CoMMIT0tee problems, word problems, puzzle problems etc. Applications of Combinatorics to understand Telephone numbering plan, understanding internet addresses, Advanced counting techniques, recurrence relations, solving recurrence relations, algorithm design, Basic understanding of complexities, basic problems of complexity of algorithms.</p>
Unit 3	<p>Relations: Relation definition , Importance of relations in computer science, Relations and their properties, Unary relations , Binary relations, Ternary relations , n-ary relations and their applications, closures of relations, equivalence relations, partial ordering. Representing relations, relation matrix, relation graph, composite relation. Operations on relations – union, intersection and join. Concepts of least upper bond, Greatest lower bond, maximal element, minimal element, Greatest element, Least element of a partially ordered set, lattices, sub lattices, chains and antichains.</p>
Unit 4	<p>Graphs: Introduction to Graphs, Importance of graph theory in computer science, Graph terminology, representing graphs, graph types, graph models, and graph isomorphism. Connectivity, Euler and Hamiltonian Paths, shortest path problems, planar graphs, graph colouring, chromatic number, Euler’s formula, kuratowski’s theorem. The four colour problem, Applications of Graph Colouring, Introduction to Trees, applications of trees, tree traversal, trees and sorting, Spanning trees, minimum spanning trees.</p>
Unit 5	<p>Languages and Grammars: Introduction to Languages and Grammars, solving problems for validity of statements according to the grammar. Importance of Language theory in Computer Science, Importance of Derivation trees, solving problems of Derivation trees, Importance of Parsing, Phrase-Structure Grammars, Types of Phrase structure grammars.</p>

1. Text Book :

Kenneth H. Rosen —Discrete Mathematics and its Applications, 5th edition , Tata McGraw-Hill Edition. Let us C, Yashavant Kanetkar, BPB Publications.

2. Reference books :

1. Kolman, Busby & Ross "Discrete Mathematical Structures" ,5th edition Pearson Education
2. Narsingh Deo "Graph Theory with Applications to Engineering. & Computer Science", 4th edition, Prentice Hall of India .
3. Discrete Structures, Logic and Computability by James L. Hein, 2nd edition, Narosa Publishing House.
4. Discrete and Combinatorial Mathematics, 5th Edition by Ralph P. Grimaldi, Addison-Wesley publication.

Class Assignments:-

1. Find a formula for the sum of the first n even positive integers.
2. Use mathematical induction to prove the formula that you found in Exercise 1.
3. Use mathematical induction to prove that $3 + 3 \cdot 5 +$
4. $5_2 + \dots + 3 \cdot 5^n = 3(5^{n+1} - 1)/4$ whenever n is a nonnegative integer.
5. Use mathematical induction to prove that $2 - 2 \cdot 7 + 2 \cdot 7^2 - \dots + 2(-7)^n = (1 - (-7)^{n+1})/4$ whenever n is a nonnegative integer.
6. Find a formula for $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots + \frac{1}{2^n}$ by examining the values of this expression for small values of n . Use mathematical induction to prove your result.
7. Show that $1^2 + 2^2 + \dots + n^2 = n(n+1)(2n+1)/6$ whenever n is a positive integer.
8. Show that $1^3 + 2^3 + \dots + n^3 = [n(n+1)/2]^2$ whenever n is a positive integer.
9. Prove that $1^2 + 3^2 + 5^2 + \dots + (2n+1)^2 = (n+1)(n+1)(2n+3)/3$ whenever n is a nonnegative integer.
10. Prove that $1 \cdot 1! + 2 \cdot 2! + \dots + n \cdot n! = (n+1)! - 1$ whenever n is a positive integer.
11. Show by mathematical induction that if $h > -1$, then $1 + nh \leq (1+h)^n$ for all nonnegative integers n . This is called **Bernoulli's inequality**.
12. Prove that $3^n < n!$ whenever n is a positive integer greater than 6.
13. Show that $2^n > n^2$ whenever n is an integer greater than 4.
14. Use mathematical induction to prove that $n! < n^n$ whenever n is a positive integer greater than 1.
15. Prove using mathematical induction that $1 - 2 + 2^2 - 3 + \dots + n(n+1) = n(n+1)(n+2)/3$
16. Whenever n is a positive integer.
17. Use mathematical induction to show that 3 divides $n^3 + 2n$ whenever n is a nonnegative integer.
18. Use mathematical induction to show that 5 divides $n^5 - n$ whenever n is a nonnegative integer.
19. Use mathematical induction to show that 6 divides $n^3 - n$ whenever n is a nonnegative integer.
20. Use mathematical induction to show that $n^2 - 1$ is divisible by 8 whenever n is an odd positive integer.
21. Use mathematical induction to show that $n^2 - 7n + 12$ is nonnegative if n is an integer greater than 3.

22. Use mathematical induction to prove that a set with n elements has $n(n - 1)/2$ subsets containing exactly two elements whenever n is an integer greater than or equal to 2.
23. Use mathematical induction to prove that a set with n elements has $n(n - 1)(n - 2)/6$ subsets containing exactly three elements whenever n is an integer greater than or equal to 3.
24. Write a small procedure for addition of a node in a binary search tree.
25. Define the following terms :-
 - a) Graph Isomorphism
 - b) Map coloring
 - c) Chromatic Polynomial
 - d) Phrase structure grammar
 - e) Euler graph.
 - f) Bipartite Graph
 - g) Hamiltonian circuits
26. Write a small C code to create a digraph.
27. Write a small C code for insertion of a node in binary search tree.
28. Implement Topological sorting algorithm in C language.
29. Write a C code to decide whether a graph is Bipartite.

MAHAKAUSHAL UNIVERSITY
JALALPUR

जाने महकाशालम्

MSc Information Technology 1st semester
MMIT0105-T Communication Skills

Unit No.	Topics
Unit 1	<p>Fundamentals of Communication (OHP & PPP): Definitions, importance, forms of communication, process of communication, channels, barriers and strategies to overcome barriers of communication.</p> <p>Listening (PPP): Def, Importance, Benefits, barriers, approaches, be a better listener, exercises and cases.</p>
Unit 2	<p>Advance Communication: Why communication? Art of communication, V3 communication, Key elements of IP communication, Quizzes, exercises and cases / incidents for practice.</p>
Unit 3	<p>Group Discussions:(PPP) Definitions, importance, process, points to be borne in mind while participating, Dos and Don'ts. Practice- if time permits or to be covered in PDP.</p> <p>Interview (PPP) Types of Interviews, Points to be borne in mind as an interviewer or an Interviewee, commonly asked questions, Dos and Don'ts. Practice- if time permits or to be covered in PDP.</p>
Unit 4	<p>Transactional Analysis: (PPP) Transactional analysis, Johari Window, FIRO-B (PPP)</p> <p>Written Communication: Cover letter, Report writing, documentation, business correspondence, preparation of manuals and project reports, Drafting emails.</p>
Unit 5	<p>Negotiation Skill: Basic principles, Building understanding, process of negotiation, essentials of negotiations. Contemporary Communication Styles, technology enabled communication</p>

1. Text Book :
 Chturvedi, P.D. and Chaturvedi Mukesh (2004), —Business Communication| Pearson Education, Singapore Pvt. Ltd.
2. Reference books :
 1. Business Communication by ICMR, Feb 2001.
 2. Toropov Brandon (2000), —Last Minute Interview Tips|, Jaico Publishing House, Mumbai.
 3. Heller Robert (1998), —Essential DK Managers: Communication Clearly|, Dorling Kindersley, London.

Class Assignments:

1. Making their Resumes.
2. Writing formal letters, Memos, drafting emails, notices.
3. Create a questionnaire, and meet new people they never met and know about the

MSc Information Technology 2nd semester
MMIT0201-T Data Structures using C++

Unit No.	Topics
Unit 1	<p>Introduction to C++: Structures, Variables in C++, References, Functions, Function Overloading, Default Values for Formal Arguments of Functions, Inline Functions.</p> <p>Class and Objects: Introduction to Classes and Objects Constructors, destructors, friend function, dynamic memory allocation, Inheritance, Overloading, Polymorphism, Templates.</p>
Unit 2	<p>Definition of data structures and abstract data types. Examples and real life applications Data Structures: Arrays, Stacks, Queues, Dequeues</p>
Unit 3	<p>Linked Lists, Singly and Doubly linked list Applications of Linked Lists, Linked implementation of queue and stacks Binary Search Tree : Definition and Implementation- preorder, post order, inorder traversal, Red Black Tree Definition and implementation</p>
Unit 4	<p>Graphs: Definition and implementation, Hash function, Collision Resolution Techniques, Hashing Applications, Time Complexity, Big – Oh - notation, Running Times, Best Case, Worst Case, Average Case, Factors depends on running time, Introduction to Recursion, Divide and Conquer Algorithm, Evaluating time Complexity</p>
Unit 5	<p>Straight Sequential Search, Array implementations, Linked implementation Binary Search, Interpolation Search. Introduction, Sorting by exchange selection, insertions, Bubble sort, Selection sort, Insertion sort, Efficiency above algorithms, Merge sort, Quick sort Algorithm, Heap sort, Hea Construction, Heap sort, Radix sort.</p>

1. Text Book :
Data Structures using C by A. M. Tenenbaum, Langsam, Moshe J. Augentem, PHI Pub.
2. Reference books :
 1. Data Structures and Algorithms by A.V. Aho, J.E. Hopcroft and T.D. Ullman, Original edition, Addison-Wesley, 1999, Low Priced Edition.
 2. Fundamentals of Data structures by Ellis Horowitz & Sartaj Sahni, Pub, 1983,AW.

Class Assignments:

1. Write a program to display message:

**Welcome to C++ Wish
you all the best.**

2. Write a program to find the area of circle, rectangle or triangle.
3. Write a program that allows the user to enter the number, and then generate the table, formatting it into ten columns and 20 lines. Interaction with the program should look like this.

```

Enter a number: 2
  2   4   6   8   10  12  14  16  18  20
 22  24  26  28  30  32  34  36  38  40
 42  44  46  48  50  52  54  56  58  60
  
```

4. Write a temperature – conversion program that gives the user the option of converting Fahrenheit to Celsius or Celsius to Fahrenheit. Then carry out the conversion. Use Floating point numbers.
5. Create the equivalent of four-function calculator. The program should request the user to enter a number, an operator, and another number. It should then carry out the specified arithmetical operation; adding, subtracting, multiplying or dividing the two numbers. (It should use a switch statement to select the operation). Finally it should display the result. When it finishes the calculation, the program should ask if the user want to do another calculation. The response can be `_y'` or `_n'`.
6. A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767), and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers.
7. A point on the two-dimensional plane can be represented by two numbers: an X coordinate and a Y coordinate. For example, (4,5) represents a point 4 units to the right of the origin along the X axis, and 5 units up the Y axis. The sum of two points can be defined as new points, and whose Y coordinate is the sum of their Y coordinates.
8. Write a program that uses a structure called point to model A point. Define three points, and have the user input values to two of them. Then set the third point equal to the sum of the other two, and display the value of the new point.
9. Create a structure called volume that uses three variables of type Distance to model the volume of a room. Initialize a variable of type Volume to specific dimensions, and then calculate the volume it represents, and print out the result. To calculate the volume, convert each dimension

from a Distance variable to a variable of type float representing feet and fractions of a foot, and then multiply the resulting three numbers.

10. Write a function called `circarea ()` that finds the area of a circle in a similar way. It should take an argument of type float and return an argument of the same type. Write a `main ()` function that gets a radius value from the user, calls `circarea ()`, and displays the result.

11. Raising a number `n` to a power `p` is the same as multiplying `n` by itself `p` times. Write a function called `power ()` that takes `double` value for `n` and an `int` value for `p`, and returns the result as `double` value. Use a default argument of 2 for `p`, so that if this argument is committed, the number will be squared. Write a `main ()` function that gets values from the user to test this function.

12. Write a function called `zeroSmaller ()` that is passed two `int` argument by reference and then sets the smaller of the two numbers to 0. Write a `main ()` program to exercise this function.

13. Write a function that takes two Distance values as arguments and returns the larger one. Include a `main ()` program that accepts two Distance figures from the user, compares them, and displays the larger.

14. Create a class that imitates part of the functionality of the basis data type `int`, Call the class `int` (note different spelling). The only data in this class is an `int` variable. Include member functions to initialize an `Int` to 0, to initialize it to an `int` value, to display it (it looks just like an `int`), and to add two `Int` values.

15. Write a program that exercises this class by creating two initialized and one un-initialized `int` values, adding these two initialized values and placing the response in the un-initialized value, and then displaying this result.

16. Create a class called `time` that has separate `int` member data for hours, minutes, and second. One constructor should initialize this data to 0, and another should initialize it to fixed values. A member function should display it, in 11:59:59 format. The final member function should add two objects of type `time` passed as arguments. A `main ()` program should create two initialized `time` objects, and one that isn't initialized. Then it should add the two initialized values together, leaving the result in the third `time` variable. Finally it should display the value of this third variable.

17. Write a function called `reversit ()` that reverse a string (anarray of `char`). Use a for loop that swaps the first and last characters, then the second and next-to-last characters, and so on.

The string should be passed to `reverse it ()` as an argument. Write a program to exercise `reversit` the program should get a string from the user, call `reversit ()` and print out the result.

Use an input method that allows embedded blanks. Test the program with Napoleon's famous phrase, —Able was I ere I saw Elbal.

18. Create a class called `employee` that contains a name (an array of `char`) and an employee number (type `Long`). Include a member function called `get data ()` to get data from the user for insertion into the object, and another function called `put data ()` to display the data. Assume the name has no embedded blanks. Write a `main ()` program use this class. It should create an array of type `employee`, and then the user input data for up to 10 employees. Finally, it should print out the data for all the employees.

19. Write a program that substitutes an overloaded `+=` operator for the overloaded `+` operator in the `STRPLUS` program. This operator should allow statements likes `s1 +=s2;` where `s2` is added (concatrated) to `s1` and the result left in `s1`. The operator should also permit the results of the operation to be used in other calculations, as in `s3 = s1 +=s2.`

20. Create a class `int`. Overload all five integer arithmetic operators (+; =; *; /; and %) so that they operate on objects of type `int`. If the result of any such arithmetic operation exceeds the normal range of `ints` = from = 32,768 to 32,767 –have the operator print a warning and terminate the program. Such a data type might be useful where mistakes caused by arithmetic overflow are unacceptable. Hint: To facilitate checking for overflow; perform the calculations using type `long`. Write a program to test this class.
21. Imagine a publishing company that markets both book and audio-cassette versions of its works. Create a class `publication` that stores the title (a string) and price (type `float`) of a publication. From this class derive two classes; `book`, which adds a page count (type `int`); and `tape`, which adds a playing time in minutes (type `float`). Each of these three classes should have a `get data ()` function to get its data from the user at the keyboard, and a `put data ()` function to display its data. Write a `main ()` program to test the `book` and `tape` classes by creating instances of them, asking the user to fill in their data with `get data ()`, and then displaying the data with `put data ()`.
22. Write a program that reads a group of numbers from the user and places them in an array of type `float`. Once the numbers are stored in the array, the program should average them and print the result. Use pointer notation wherever possible.
23. Start with the `String` class from the `NEWSTR`. Add a member function called `up it ()` that converts the string to all upper case. You can use the `toupper ()` library function, which takes a single character as an argument and returns a character that has been converted (if necessary) to uppercase. This function uses the `CTYPE.h` header file. Write some code in `main ()` to test this function.
24. Add a destructor to the `LINKLIST` program. It should delete all the links when a link list object is destroyed. It can do this by following along the chain, deleting each link as it goes. You can test the destructor by having it display a message each time it deletes a link; it should delete the same number of links that were added to the list. (A destructor is called automatically by the system for any existing objects when the program exits.)
25. Imagine the same publishing company that markets both book and audio-cassette versions of its works. As in that exercise, create a class called `publication` that stores the title (a string) and price (type `float`) of a publication. From this class derive two classes: `book`, which adds a page count (type `int`); and `tape`, which adds a playing time in minutes (type `float`). Each of the three classes should have a `get data ()` function to get

MSc Information Technology 2nd semester
MMIT0202-T Database Management System

Unit No.	Topics
Unit 1	<p>Introduction: Advantages of DBMS approach, Various views of data, data independence, schema & sub-schema, primary concept of data models, database languages, transaction management, database administrator & user, data dictionary, database architectures.</p> <p>ER model: Basic concept, design issues, mapping constraint, keys, ER diagram, weak & strong entity-sets, specialization & generalization, aggregation, inheritance, design of ER schema, Reduction of ER Schema to tables. Domains, relation, kind of relation, Relational databases, Various types of keys: candidate, primary, alternate & foreign keys.</p>
Unit 2	<p>Relational Algebra and SQL: The structure, relational algebra with extended operations, modification of database, Idea of relational calculus.</p> <p>Relational Database: Basic structure of SQL, Set operation, Aggregate functions, Null values, Nested Sub queries, derived relations, views, Modification of database, join relation, Domain, relation & keys, DDL in SQL. Programming concepts of PL/SQL, Stored procedure, Database connectivity with ODBC/JDBC.</p>
Unit 3	<p>Functional dependencies: Basic definitions, Trivial & non trivial dependencies, closure set of dependencies & of attributes, Irreducible set of dependencies, FD diagram.</p> <p>Normalization: Introduction to normalization, non loss decomposition, First, second and third normal forms, dependency preservation, BCNF, multivalued dependencies and fourth normal form, join dependencies and fifth normal form.</p> <p>Database Integrity: general idea, integrity rules, Domain rules, Attributes rules, assertion, triggers, integrity & SQL</p>
Unit 4	<p>Transaction Management: basic concept, ACID properties, transaction state, Implementation of atomicity & durability, Concurrent execution, Basic idea of serializability.</p> <p>Concurrency & Recovery: Basic idea of concurrency control, basic idea of deadlock, Failure Classification, storage structure-types, stable storage implementation, data access, recovery & Atomicity: log based recovery, deferred database modification, immediate database modification, checkpoints.</p>
Unit 5	<p>Storage Structure: overview of physical storage media, magnetic disk: performance & optimization, RAID.</p> <p>File Organization: File organization, Organization of records in files, basic concept of Indexing, ordered indices: B+ tree & B tree index files.</p> <p>Query processing, Query optimization, Introduction to data mining and data warehousing.</p>

1. Text Book :
—Database System concepts – Henry F. Korth, Tata McGraw Hill 4th Edition
2. Reference books :
 1. “Fundamentals of Database Systems”, Elmasri R, Navathe S, Addison Wesley 4th Ed., ISBN 0321122267
 2. An introduction to database system- Bipin C. Desai
 3. An introduction to Database System - C.J Date
 4. SQL, PL/SQL The programming language of Oracle- Ivan Bayross

Class Assignments:

(Student should submit their assignment in hard copy (hand written) as well as soft copy to the respective faculty)

- A. Prepare the following case study on ER diagram and normalized database design based on FD's
 - (a) Database design for Retail Banking
 - (b) Database design for Technical Training Institute
 - (c) Database design for an Internet Book Shop
 - (d) Database design for Customer Order Warehouse
 - (e) Database design for University Registrar's Office

- B. Define the schema for the following databases with specific data type and constraints, the table name and its fields name are to be taken from database description which are given below :

1. Sales Information System

A database is being constructed for storing sales information system. A product can be described with a unique product number, product name, selling price, manufacturer name. The product can be sold to a particular client and each client have it own unique client number, client name, client addresses, city, pin code, state and total balance to be required to paid. Each client orders to buy product from the salesman. In the order, it has unique sales order number, sales order date, client number, salesman number (unique), billed whole payment by the party or not and its delivery date. The information associated with salesman is name, addresses, city, pin code, state, salary of the sales man, delivery date, total quantity ordered, product rate.

Write the SQL queries for the following –

- a) Retrieve the list of names and the cities of all the clients.
- b) List the various products available.
- c) Find the names of all clients having 'a' as the second letter in their names.
- d) List all the clients who are located in 'INDORE'.
- e) Find the products whose selling price is greater than 2000 and less than or equal to 5000.

- f) Add a new column NEW_PRICE into the product_master table.
- g) Rename the column product_rate of Sales_Order_Details to new_product_rate.
- h) List the products in sorted order of their description.
- i) Display the order number and date on which the clients placed their order.
- j) Delete all the records having delivery date before 25th March, 2010.
- k) Change the delivery date of order number ON01008 to 16-05-10.
- l) Change the bal_due of client_no CN01003 to 1200.
- m) Find the product with description as '_HDD1034' and '_DVDRW'.
- n) List the names, city and state of the clients not in the state of '_MP'.
- o) List of all orders that were cancelled in the month of March.

2. College Department Management

A student is described by a unique Roll Number, Name, Address, and Semester. Each student enrolls himself/herself in an Academic programme offered by a Department. Academic programmes have programme name(unique), duration, a programme code(unique) and a list of subjects (both core and elective subject) while the departments have department code (unique), department name (unique), HoD who is a Teacher and list of courses offered by it. Each teacher is described by employee code (unique), name, department and designation. A student registers

some courses in a semester. A course is described by a unique course number, title of the course, credit allotted for the course and offering department. Database stores the grades obtained by different students in different courses registered by him/her in different semesters. Database also stores information about the courses offered by a department in a semester, the corresponding teacher(s) for each course.

Write the SQL queries for the following –

- a) Find all the students' name, city, course allotted from the SIT department.
- b) List the total number of Faculty in the SIT department.
- c) List the available courses from the SIT department.
- d) List the all students in a particular semester.
- e) List the students who earned CGPA greater than or equal to 8.5.
- f) How much subjects are registered by a student in each semester.
- g) List the common students who are allotted the same courses of both the programme MCA and M. Tech.
- h) List the total number of student enrolled in the subject DBMS.
- i) Retrieve the semester of the student under DBMS subject.
- j) Retrieve the entire student name and arrange into ascending order.
- k) Modify a student address DEWAS to INDORE where sdt_id='ITI08002'.
- l) Find the total credit point of student required to complete for a course like MCA.
- m) List the all courses which are related to computer science.

- n) Retrieve all the students located at '_INDORE'.
- o) Find the total number of department in our database.
- p) List the all courses which are related to computer science.

MMIT0202-P Database Management System LAB

Database Management System lab:-

1. Database Normalization: This topic covers the process of organizing the attributes and tables of a relational database to minimize redundancy and dependency.
2. Query Optimization: Discuss techniques for optimizing queries to improve the performance of database operations.
3. Indexing: Explore the concept of indexing in databases, including different types of indexes such as B-tree, hash indexes, and their implementation.
4. Transaction Management: Cover the concepts of ACID properties (Atomicity, Consistency, Isolation, Durability) and transaction management techniques like locking and concurrency control.
5. Backup and Recovery: Discuss strategies and techniques for backing up and recovering data in a database system.
6. Security and Authorization: Cover topics related to database security, including user authentication, access control and encryption.
7. Data Warehousing and Data Mining: Introduce the concepts of data warehousing and data mining, including techniques for extracting useful information from large datasets.
8. Distributed Databases: Explore the challenges and techniques involved in managing distributed databases across multiple locations.
9. NoSQL Databases: Discuss the characteristics and use cases of NoSQL databases, including document-oriented, key-value, and columnar databases.
10. Database Design and Implementation: Cover the process of designing and implementing a database schema, including entity-relationship modeling and schema normalization

MSc Information Technology 2nd semester
MMIT0203-T Software Engineering

Unit No.	Topics
Unit 1	Introduction to Software Engineering & Software Processes: Software problem, Software engineering problem, Software engineering approach, Software characteristics and Applications. Software processes and its components, characteristics of software processes, Software development processes: Linear Sequential model, Prototyping model, Iterative Enhancement model, Spiral model, Time boxing model, RAD model, Component based development, Comparative study of various development models.
Unit 2	Project management process: The people, product, process and project, Phases of project management process, Project life cycle, the W5HH principle. Software configuration management process, Process management process. Project Planning: Metrics and measurements, Project estimation (Size & Cost), Project Scheduling, Staffing and personnel planning, Software configuration management plans, Quality assurance plans, Project monitoring plans, Risk management.
Unit 3	Software Requirement Analysis and Specification: Software requirements, Functional & Non functional requirements, Problem analysis (Structured analysis and Object Oriented analysis, Prototyping approach). Software Requirements specifications (SRS), Validation and Verification, Metrics. Software Design: Design principles: Problem partitioning and hierarchy, Abstraction, Modularity, Top-down and Bottom-up strategies. Effective Modular design: functional independency, Cohesion, Coupling. Structured design methodology.
Unit 4	Software Quality Assurance: Quality concept, Quality management system, movements and assurance, Software reviews: formal and technical. Formal approaches to SQA, Statistical software quality assurance, Software reliability, Capability Maturity Model (CMM), ISO 9000, Six sigma, SQA plan.
Unit 5	Software Testing: Software testing techniques: Testing fundamentals, White box testing, Black box testing, Testing for specialized environments, architectures and applications. Software testing strategies: A strategic approach to software testing, Strategic issues, Unit testing, Integration testing, Validation testing and system testing, The art of debugging.

Text Book :

An Integrated Approach to Software Engineering- Pankaj Jalote, Narosa Publishing House.

2. Reference books :

1. Software Engineering- Ian Sommerville, Pearson Education, New Delhi.
2. Software Engineering Concepts-Richard E. Fairly, Tata McGraw Hill Inc. New York.
3. Software Engineering: Principle & Practice-W. S. Jawadekar, Tata McGraw-Hill, New York.

Class Assignments:

Assignment 1:

Select a project that you want to develop and define the aim, objectives, and goals of your project. Select a software development methodology to develop the same project. Perform various phases of the methodology to produce work-products.

Assignment 2:

1. Suppose a program for solving a problem costs C , and a programming product for solving that problem costs $9C$. Where do you think the extra $8C$ cost is spent? Suggest a possible breakdown of this extra cost.
2. If you are given extra time to improve the reliability of the final product developing a software product, how would you distribute this extra time?
3. Suggest ways of reducing the cost due to rework.
4. Suggest some ways to detect software errors in the early phases (when implementation is not yet complete).

Assignment 3:

1. Define & discuss Time boxing and Component based development model.
2. What problems will a Software Development Organization face if it does not have sufficiently document in its software process?
3. Suppose you were to plan to undertake the development of product base with large number of technical as well as customer, justify which life cycle model would you adopt.
4. Suggest a suitable life cycle model, for software project which your organization has undertaken on behalf of a certain change his requirements frequently justify.
5. Which of the following process models discussed in this chapter would you follow for the following projects? Give justifications.
 - a) A simple data processing project
 - b) A data entry system for office staff who have never used computers before. The user interface and user friendliness are extremely important.
 - c) A new system for comparing fingerprints. It is not clear if the current algorithms can compare fingerprints in the given response time constraints.
 - d) A spreadsheet system that has some basic features and many other desirable features that use these basic features
 - e) A new missile tracking system. It is not known if the current hardware /software technology is mature enough to achieve the goals.
 - f) An on-line inventory management system for an automobile industry.
 - g) A flight control system with extremely high reliability. There are many potential hazards with such a system.

6. It is reasonable to assume that if software is easy to test, it will be easy to maintain. Suppose that by putting extra effort in design and coding you increase the cost of these phases by 15%, but you reduce the cost of testing and maintenance by 5%. Will you put in the extra effort?
7. Suppose you can measure the number the defects detected during the various reviews and testing. However, the customer requires an estimate of the Number of defects remaining at delivery time. How will you build a model to predict this? Assume the existence of any data you need.

MMIT0203-P Software Engineering LAB

1. **Software Testing Techniques:** In this topic, students could learn about different software testing techniques such as black-box testing, white-box testing, unit testing, integration testing, system testing, and acceptance testing. The lab could involve practical exercises where students implement and execute tests using these techniques on sample software projects.
2. **Test-Driven Development (TDD):** TDD is a development approach where tests are written before the actual code. In this lab, students could practice TDD by working on small programming tasks or projects, writing tests first, and then implementing the code to make the tests pass.
3. **Continuous Integration and Continuous Deployment (CI/CD):** This topic covers the practices of integrating code changes frequently and automatically deploying them to production environments. The lab could involve setting up CI/CD pipelines using tools like Jenkins, GitLab CI/CD, or Travis CI, and automating the build, test, and deployment processes.
4. **Software Maintenance and Evolution:** Students could learn about the challenges and best practices associated with maintaining and evolving software systems over time. The lab could involve analyzing and making changes to existing codebases, documenting changes, and managing version control.
5. **Software Metrics and Quality Assurance:** This topic covers measuring and assessing the quality of software using various metrics and techniques. The lab could involve using software quality metrics tools to analyze code quality, identify potential issues, and improve software maintainability and reliability.
6. **Software Configuration Management:** Students could learn about managing software configurations, including version control, change management, and configuration identification. The lab could involve using version control systems like Git, SVN, or Mercurial to manage code changes and collaborate on projects.
7. **Software Design Patterns:** Introduce students to commonly used design patterns in software development, such as creational, structural, and behavioral patterns. The lab could involve implementing these patterns in code and understanding their use cases and benefits.
8. **Software Documentation:** Cover the importance of documenting software artifacts such as requirements, design, code, and user manuals. The lab could involve creating documentation for a small software project, including requirements specifications, design documents, and API documentation.

9. Software Project Management: This topic covers project management methodologies, tools, and techniques used in software development projects. The lab could involve using project management tools like Jira, Trello, or Asana to plan and track software development tasks, manage project schedules, and collaborate with team members.

10. Software Development Ethics and Professionalism: Discuss ethical considerations and professional responsibilities in software development, including issues related to privacy, security, intellectual property, and social impact. The lab could involve case studies and discussions on ethical dilemmas faced by software engineers and ways to address them.



MSc Information Technology 2nd semester
MMIT0204-T Computer Architecture and Organization

Unit No.	Topics
Unit 1	Technological trends, measuring performance, speed up Amdahl's law. Basic structure of computer hardware: Functional units and components in computer organization: The memory unit, the input and output subsystem, the bus structures, ALU. Program development tools: Compiler, interpreter, and assembler. Introduction to 8085 micro processor architecture. pin out diagram of 8085 microprocessor. Addressing modes of 8085 processor.
Unit 2	Instruction set of 8085 processor: data transfer instructions, arithmetic instructions, branch instructions etc. assembly language programming examples. Introduction to 8088 microprocessor. Bit pair multiplication. Integer division. BCD arithmetic, Design of ALU. Concepts of instruction formats and instruction set, Instruction set types, types of operands and operations. Generation of memory address and addressing modes.
Unit 3	STACKS and QUEUS, GPR organization and stack based organizations. Encoding of machine instructions. Features of RISC and CISC processors. Processing unit design: Processor micro architecture –I Fundamentals concepts for data path implementation. Processor micro architecture-II data path implementation.
Unit 4	Instruction pipelining, instruction pipelining hazards, data dependency hazards and control hazards. Overcoming hazards, instruction set design influence on pipelining. Parallel processing and pipelining, pipelining in RISC and CISC processors. Super scalar processors. In order and out of order execution.
Unit 5	Instruction level parallelism, introduction to VLIW processors, vector processors, CACHES: Data caches, instruction caches and unified caches Cache implementations. Fully associative and direct mapped caches. Write back versus write through caches. Input Output organization, accessing I/O devices, Interrupts. Memory mapped I/O and I/O mapped I/O.

1. Text Book :
 Computer Architecture and Organisation: Nicholas Carter, Schaum's outlines TMH adaptation, 2nd ed. 2010.
2. Reference books :
 1. The 8088 and 8086 Microprocessors by Walter A. Tribel, Avtar Singh.
 2. Computer Organization & Architecture by William Stallings.
 3. Computer Architecture & Parallel Processing, Hwang & Briggs, McGraw Hill
 Microprocessor Archi. Prog. and app. With 8085/8080 By Ramesh S. Gaonkar.

MSc Information Technology 2nd semester**MMIT0205-T Organization and Management Concepts**

Unit No.	Topics
Unit 1	Fundamentals of Management : Definition, Its Nature & purpose. The Evolution of Management Thought. The Systems Approach to the Management Process. The Functions of Manager.
Unit 2	Planning : Types of Plans. Steps in Planning. Decision making : Rational Decision Making. Programmed And Non Programmed Decision Making.
Unit 3	Organizing : Formal and Informal Organization. Organizational division. Departmentation : Bases of departmentation, Organization charts Line and Staff concepts.
Unit 4	Staffing: Definition of Staffing. The Systems Approach to HRM. Selection process, Techniques and Instruments. Performance Appraisal and Career Strategy: Rewards and stress of managing, Formulating the Career Strategy.
Unit 5	Motivation – Theories of motivation. Special motivation techniques. Leadership – Definition. The Managerial Grid. Controlling: The Basic Control Process. Control Techniques. Time Management.

1. Text Book :

R.D Agarwal, —Organization and Management| Tata McGraw Hill Inc.

2. Reference books :

1. Koontz & Wehrich , —Essentials of Management| Tata McGraw Hill Inc.

MSc Information Technology 3rd semester**MMIT0301-T Human Computer Interaction and Web Design**

Unit No.	Topics
Unit 1	The graphical user interface – popularity of graphics, the concept of direct manipulation, graphical system, Characteristics, Web user – Interface popularity, characteristics- Principles of user interface. Human interaction with computers, importance of human characteristics human consideration, Human interaction speeds.
Unit 2	Screen Designing : Design goals – Screen planning and purpose, organizing screen elements, ordering of screen data and content – screen navigation and flow – Visually pleasing composition – amount of information – focus and emphasis – presentation information simply and meaningfully – information retrieval on web – statistical graphics – Technological consideration in interface design. Components – text and messages, Icons and increases – Multimedia, colors, uses problems, choosing colors.
Unit 3	Overview Web technologies: HTML5, DHTML and CSS, Java Script, DOM, CGI, Introduction to XML and ASP.
Unit 4	Basics of Web Design, Focus on the customer, Working with your clients, FYI information on domains, Web design patterns, Site development processes, Naming conventions, Purpose of a website, Planning your site. Design Principles, Design principles, Effective Page Layouts, Layout options , Using tables, layers, frames, Working with forms, Using CSS to add interest , and flexibility to a design ,
Unit 5	Page Layout – Navigation, Principles of effective navigation , More than one way to navigate, Hierarchal organization, Task-based organization, Other ways to organize your navigation, Making navigation easy, Color and Graphics, Using color on your pages , flashing colors, bright colors, Pros and cons of background colors Contrast, Do's and don'ts of Graphics, Dimension vs weight, Wrapping text around a graphic.

TEXT BOOKS

1. The essential guide to user interface design, Wilbert O Galitz, Wiley DreamaTech.
2. Web enabled Commercial Application Development Using HTML, DHTML , JavaScript, Perl ,CGI, IvanBayross, bpb publication First Edition, 2002
- 3.

Reference books

1. Human Computer Interaction, Alan Dix et.al, Pearson Education. First dition,2004
Designing the user interface, Ben Shneidermann , 3rd Edition Pearson Education Asia

MSc Information Technology 3rd semester**MMIT0302-T Database Applications and Tools**

Unit No.	Topics
Unit 1	Database Environment: Data versus information, traditional file processing, disadvantages, database approach, range of database application, advantages of database approach. Cost and risk factors, components of database environment, evolution of database system.
Unit 2	Database Development Process: Information engineering, information architecture, enterprise data model, planning, SDLC, CASE etc. Steps of planning, strategic planning factors, corporate planning objects. Developing preliminary data model, and use of planning matrices, SDLC steps, CASE role, people in database development, three-schema architecture for database development. Examples to demonstrate the development process. Modeling Data in the Organisation: Modeling of the rules of organization, data names and definitions, ER model constructs entities and its types, attributes, relationships, degree, unary, binary, ternary, n-ary, cardinalities constraints, ER modeling examples.
Unit 3	Enhanced ER modeling: supertype, subtypes, specialization, generalization, specifying constraints in EER models, completeness, Disjointness, discriminators, defining super/sub type hierarchies, EER modeling examples, live demos modeling for few scenarios. Logical database design and relational model development, Relational model properties, keys, primary, secondary, composite, properties of relations. Codd's rules, integrity constraints, creating relational tables, Transform EER diagrams into relations, seven different steps for mapping EER model into relations,
Unit 4	Introduction to normalization, steps, functional dependencies, basic normal forms, definition of first, second, third normal form and removing anomalies from the relations. De-normalization and merging relations.
Unit 5	Special Topics (Overview) : Data Warehousing, Data Mining, Distributed Databases, Object oriented modeling, definitions, activities in phases of model development, advantages of OOM, UML class diagrams, Example of a model development.

1. Text Book:

—Modern Database Management| Seventh Edition, Hoffer, Prescott, McFadden Pearson Education|

2. Reference books :

1. Database Systems |Thomas M. Connolly, Carolyn E. Begg|Pearson Education.
2. Raghu R and Johannes G., —Database management Systems|, Mc Hill 3rd Ed 2002, Elmasri R, Navathe S, —Fundamentals of Database Systems|, Addison Wesley 4th Ed.

Class Assignments:-

Exercise-4: For the project of your choice, describe the development of normalized relational model from the ER/EER model and document the complete relational design

Exercise-5: For the project of your choice, describe the SQL commands for creating all the tables, and sort and search of I/O information, using the knowledge gained from this chapter and the previous design.

Exercise 6: Full project report preparation and presentation of the project.

Database Applications and Tools lab:-

1. Introduction to Web Development with Databases:

- Brief overview of web development frameworks and technologies (e.g., HTML, CSS, JavaScript, and server-side scripting languages like PHP, Python, or Node.js).
- Introduction to the role of databases in web applications and the importance of seamless integration.

2. Setting Up the Environment:

- Install and configure a local web server environment (e.g., Apache, Nginx) and a database management system (e.g., MySQL, PostgreSQL).
- Set up a sample database with tables and sample data to be used in the lab exercises.

3. Backend Development with Database Integration:

- Choose a server-side scripting language (e.g., PHP, Python, Node.js).
- Create backend scripts to connect to the database, execute queries, retrieve data, and process it for web display.
- Implement CRUD (Create, Read, Update, Delete) operations on the database through web forms or API endpoints.

4. Frontend Development with Database Interaction:

- Develop web pages or components using HTML, CSS, and JavaScript to interact with the backend.
- Implement forms for user input and display data retrieved from the database dynamically on web pages.
- Utilize AJAX techniques for asynchronous data loading and updating without page reloads.

5. User Authentication and Authorization:

- Implement user authentication and authorization mechanisms to control access to database resources.
- Securely handle user login credentials and manage session data for authenticated users.

6. Advanced Database Integration Techniques:

- Explore techniques for optimizing database interactions in web applications, such as caching, pagination, and lazy loading.
- Implement server-side validation and error handling to ensure data integrity and security.

7. Testing and Debugging:

- Test the integrated web application thoroughly to ensure proper functionality and data consistency.
- Debug and troubleshoot any issues encountered during testing, including database connectivity errors and data retrieval problems.

8. Deployment and Hosting:

- Discuss deployment options for web applications with integrated databases, such as deploying to a local server, cloud hosting platforms (e.g., AWS, Heroku), or shared hosting services.
- Demonstrate the process of deploying the web application and configuring database connections in a production environment.

9. Performance Optimization and Scalability:

- Explore strategies for optimizing database performance and scaling web applications to handle increased traffic and data volume.
- Discuss techniques like database indexing, query optimization, and horizontal scaling using replication or sharding.

10. Project Work and Presentation:

- Assign a project task where students have to develop a full-fledged web application integrating a database.
- Students present their projects to the class, demonstrating the functionality, database integration, and deployment aspects.



MAHAKAUSHAL UNIVERSITY
JABALPUR

जाल महकौशलम्

MSc Information Technology 3rd semester**MMIT0303-T Computer Networks**

Unit	TOPIC
Unit-1	<p>Introduction: Computer Network, Goals and Applications, Reference models – OSI and TCP/IP. A Comparative study.</p> <p>LAN, MAN and WAN and topologies, LAN components – File server, Workstations, Network Adapter Cards. Connection Oriented and Connection less services, Switching Techniques – Circuit Switching, Packet Switching.</p> <p>Assignment I</p>
Unit-2	<p>Data Link Layer: Design Issues, Framing, Error Detection: Parity Check, Check Sum and Cyclic Redundancy Check (CRC); Correction Technique: Hamming code.</p> <p>Flow Control: Elementary Data Link Protocols: An Unrestricted Simplex Protocol, Simplex Stop-and-Wait Protocol, Sliding Window Protocols: One-Bit Sliding Window Protocol</p> <p>Go Back N and Selective Repeat. Data link layer in the Internet: SLIP and PPP.</p> <p>Assignment II</p>
Unit-3	<p>MAC Sublayer: Multiple access protocols: Aloha, CSMA Protocols; Collision-Free Protocols; IEEE MAC Sublayer protocols: 802.3 802.4, 802.5 and their management.</p> <p>High speed LANs – Fast Ethernet, FDDI, Wireless LANs</p> <p>Assignment III</p>
Unit-4	<p>Network Layer: Design issues, Routing Principles. Routing Algorithms: Optimality Principle, Shortest Path Routing, Flooding, Distance Vector Routing. Link State Routing, Hierarchical Routing, Broadcasting Routing, Multicast Routing, The Network Layer in the Internet: Internet Protocol, Internet addressing and Internet Control protocols.</p> <p>Assignment IV</p>
Unit-5	<p>Transport Layer: Elements of Transport Protocols, Addressing, Connection Establishment, Connection Release, Flow Control and Buffering, Multiplexing. The Internet Transport Protocol UDP: Introduction to UDP. Introduction to TCP, The TCP Service Model, The TCP Protocol.</p> <p>Application layer: Client Server Architecture, DNS, WWW and HTTP, Cookies, Proxy Server.</p> <p>E-mail Protocols (SMTP, POP3, IMAP, MIME), FTP, TELNET.</p> <p>Network Security: Cryptography, Symmetric- key Algorithms, Public- key Algorithms, Digital Signatures.</p> <p>Assignment V&Assignment VI</p>

1. Required Text

- Computer Networks, Andrew S. Tanenbaum, Addison-Wesley, 4th Ed.

Essential References

- Data Communications and Networking, B.A. Frouzan, McGraw-Hill.

Assignment I

1. Define the following

i) Computer networks	ii) Autonomous computer
iii) Distributed system Vs Computer networks	iv) LAN, MAN, WAN
vi) Protocols, interfaces and services	v) Internet
vii) Host or end system	viii) Subnet
2. Write the need of layered architecture.
3. Explain OSI and TCP/IP Model. Also write the development stages of these models(i.e. When these model came., who proposed and further development).
4. Compare OSI and TCP/IP Model.

Assignment II

1. A bit string, 011110111110111110, needs to be transmitted at the data link layer. What is the string actually transmitted after bit stuffing?
2. An 8-bit byte with binary value 10101111 is to be encoded using an even-parity Hamming code. What is the binary value after encoding?
3. What is the remainder obtained by dividing $x^7 + x^5 + 1$ by the generator polynomial $x^3 + 1$?
4. What is the mechanism of sliding window flow control?

Assignment III

1. Consider the delay of pure ALOHA versus slotted ALOHA at low load. Which one is less? Explain your answer.
2. Sketch the Manchester encoding for the bit stream: 0001110101.
3. Describes Carrier Sense Multiple Access Protocols in short?
4. Difference between slotted and pure aloha?

Assignment IV

1. Give two example computer applications for which connection-oriented service is appropriate. Now give two examples for which connectionless service is best.
2. Describes distance vector and link state routing algorithm?
3. What are the functions of router, bridges, repeaters, and gateways?
4. Difference between adaptive and non adaptive routing?
5. Describes Dijkstra algorithm?

Assignment V

1. Explain different classes of in internet address?
2. Explain purpose of ARP, RARP, ICMP, and IGMP?
3. What is host id and net id?
4. What is the purpose of sub netting?
5. Explain IP datagram header format?
6. Explain UDP and TCP datagram header format?

Assignment VI

2. Write in short about following:-
TELNET, FTP, SMTP, POP, MIME, HTTP, URL , CGI, DNS

MMIT0303-P Computer Networks LAB

Computer Networks lab:-

1. Network Configuration and Troubleshooting:
 - Configure networking settings on different operating systems (Windows, Linux, macOS).
 - Troubleshoot common network connectivity issues such as IP address conflicts, DNS resolution problems, and firewall misconfigurations.
2. Packet Sniffing and Analysis:
 - Use packet sniffing tools like Wireshark to capture and analyze network traffic.
 - Identify different network protocols, examine packet headers, and analyze network behavior.
3. Network Protocols and Services Configuration:
 - Configure network protocols such as TCP/IP, DHCP, DNS, and SNMP.
 - Set up network services like FTP, SSH, HTTP, and SMTP, and test their functionality.
4. LAN Setup and Configuration:
 - Design and implement a local area network (LAN) with multiple devices (computers, routers, switches).
 - Configure IP addressing, subnetting, and VLANs to segment the network logically.
5. Wireless Networking Setup:
 - Set up and configure a wireless network using Wi-Fi access points and routers.
 - Secure the wireless network with encryption (WPA2, WPA3), MAC filtering, and SSID hiding.
6. Network Security Assessment:
 - Perform security assessments on network devices and services.
 - Identify vulnerabilities and weaknesses using scanning tools like Nmap and vulnerability scanners.
7. Virtual Private Network (VPN) Setup:
 - Configure a VPN server and client to establish secure connections over the internet.
 - Implement VPN protocols like OpenVPN or IPsec and test connectivity between client and server.
8. Network Address Translation (NAT) and Port Forwarding:
 - Configure NAT and port forwarding on routers to enable internal hosts to access external services.
 - Test port forwarding configurations for services like web servers, FTP servers, and gaming servers.

9. Quality of Service (QoS) Configuration:

- Implement QoS policies on routers and switches to prioritize network traffic based on application requirements.
- Configure traffic shaping, bandwidth allocation, and packet prioritization for real-time applications like VoIP and video streaming.

10. Network Monitoring and Management:

- Set up network monitoring tools like Nagios, Zabbix, or Cacti to monitor network performance and health.
- Configure alerts and notifications for network events such as downtime, high traffic, and device failures.



MSc Information Technology 3rd semester
MMIT0304-T : Advanced java Programming

Unit No.	Topics
Unit 1	JAVABASICSREVIEW Java streaming-Networking-Event handling - Multithreading – Byte code Interpretation - Customizing application - Data Structures - Collection classes.
Unit 2	DISTRIBUTED COMPUTING Custom sockets - Remote Method Invocation - Activation - Object serialization -Distributed garbage collection - RMI - IIOP - Interface definition language – CORBA-JINI overview.
Unit 3	JAVA BEANS AND SWING Bean concepts - Events in bean box -Bean customization - Persistence - Application - deployment using swing - Advanced swing techniques - JAR file handling.
Unit 4	JAVAENTERPRISEAPPLICATIONS -JNI-Servlets-Java Server Pages - JDBC - Session beans - Entity beans - Programming and deploying enterprise Java Beans - Java transactions. RELATEDJAVATECHNIQUES.
Unit 5	Graphics Java Media Frame work - 3D graphics - Internationalization - Case study - Deploying n-tierapplication, E- commerce applications.

REFERENCES:

1. Deitel&Deitel,"JavaHowtoprogram",PrenticeHall,4thEdition,2000.
2. GaryCornellandCayS.Horstmann,"CoreJavaVol1andVol2",Sun Microsystems Press, 1999.
3. StephenAsbury,ScottR.Weiner,Wiley,"DevelopingJavaEnterpriseApplications", 1998.

MMIT0304-P : Advanced java Programming LAB

1. Multithreading and Concurrency:

- Implement multithreaded programs using Java's Thread class and Runnable interface.
- Explore synchronization mechanisms such as synchronized blocks and methods to ensure thread safety.
- Practical exercises on solving concurrency issues like race conditions and deadlocks.

2. Networking and Socket Programming:

- Develop client-server applications using Java's Socket API for network communication.
- Implement protocols like TCP and UDP for reliable and connectionless communication.
- Practical exercises on creating chat applications, file transfer utilities, or network-based games.

3. Java Database Connectivity (JDBC):

- Integrate Java applications with relational databases using JDBC.
- Perform database operations like querying, updating, and executing stored procedures.
- Practical exercises on connecting to a database, executing SQL queries, and handling result sets.

4. Java EE Technologies:

- Explore Java Enterprise Edition (Java EE) technologies like Servlets, JavaServer Pages (JSP), and JavaServer Faces (JSF).
- Develop web applications using Servlets and JSP for server-side processing and dynamic content generation.
- Practical exercises on building web forms, handling user input, and displaying dynamic content.

5. Enterprise Application Development with Spring Framework:

- Introduction to the Spring Framework for building enterprise Java applications.
- Implementing dependency injection, aspect-oriented programming (AOP), and inversion of control (IoC) with Spring.
- Practical exercises on developing Spring-based web applications, RESTful services, and data access layers.

6. Web Services Development with JAX-RS and JAX-WS:

- Implement RESTful and SOAP web services using Java API for RESTful Web Services (JAX-RS) and Java API for XML Web Services (JAX-WS).
- Practical exercises on creating and consuming web services for inter-application communication.

7. JavaFX GUI Development:

- Build modern and rich graphical user interfaces (GUIs) using JavaFX.
- Create interactive UI components, layouts, and animations with JavaFX.
- Practical exercises on developing desktop applications with advanced UI features using JavaFX.

8. Integration Testing and Test-Driven Development (TDD):

- Implement unit tests and integration tests for Java applications using frameworks like JUnit and Mockito.
- Practice test-driven development (TDD) by writing tests before implementing code.
- Practical exercises on writing test cases, mocking dependencies, and ensuring code quality through automated testing.

9. Security and Authentication in Java Applications:

- Explore security features and best practices for Java applications, including authentication, authorization, and encryption.
- Implement authentication mechanisms like username/password, OAuth, or JWT (JSON Web Tokens).
- Practical exercises on securing web applications, protecting sensitive data, and preventing common security vulnerabilities.

10. Asynchronous Programming with CompletableFuture and Reactive Streams:

- Learn asynchronous programming techniques using CompletableFuture and Reactive Streams in Java.
- Implement non-blocking I/O operations and reactive programming patterns for handling asynchronous events.
- Practical exercises on developing responsive and scalable applications using asynchronous programming paradigms.

Class Assignments:

1. Write a program that produces the following output: Hello World!
—It's been nice knowing you —.
Goodbye world!
2. State the order of evaluation of the operations in each of the following Java statements and implement them to show the value of x after each statement.

$$x = 7 + 3 * 6 / 2 - 1;$$

$$x = 2 \% 2 + 2 * 2 - 2 / 2;$$

$$x = (3 * 9 * (3 + (9 * 3 / (3))));$$
3. Write an application that declares 5 integers, determines and prints the largest and smallest in the group.
 4. Write an application that takes 3 parameters as sides of triangle and calculate area of triangle.
5. Write an application that declares two integers, determines whether the first is a multiple of the second and print the result. [Hint: Use the remainder operator.]
 6. Write a program to find all prime numbers between 100 to 1000.
 7. Write a program to check whether the given number is palindrome or not.
 8. Write an application that evaluates the factorial of the integers from 1 to 5.

9. Write a program that accepts an integer from user and check whether the number is Fibonacci number or not.
10. Read a positive integer value, and compute the following sequence: If the number is even, halve it; if it's odd, multiply by 3 and add 1. Repeat this process until the value is 1, printing out each value. Finally print out how many of these operations you performed.

Typical output might be:

Initial value is 9
 Next value is 28
 Next value is 14
 Next value is 7
 Next value is 22
 Next value is 11
 Next value is 34
 Next value is 17
 Next value is 52
 Next value is 26
 Next value is 13
 Next value is 40
 Next value is 20
 Next value is 10
 Next value is 5
 Next value is 16
 Next value is 8
 Next value is 4
 Next value is 2

Final value 1, number of steps 19

If the input value is less than 1, print a message containing the word Error and perform an `exit(0)`;

11. Write a program which will read an integer value for a base, then read a positive integer written to that base and print its value. Read the second integer a character at a time; skip over any leading nonvalid (i.e. not a digit between zero and ``base-1'') characters, then read valid characters until an invalid one is encountered.

Input Output

=====

10 1234 1234

8 77 63 (the value of 77 in base 8, octal)

2 1111 15 (the value of 1111 in base 2, binary)

The base will be less than or equal to 10.

12. Write an application that uses *String* method *compareTo* to compare two strings defined by the user.
13. Write an application that uses *String* method *equals* and *equalsIgnoreCase* to tests any two string objects for equality.
14. Write an application that uses *String* method *indexOf* to determine the total number of occurrences of any given alphabet in a defined text.
15. Write an application that uses *String* method *concat* to concatenate two defined strings.
16. Write an application that finds the length of a given string.
17. Write an application that uses *String* method *charAt* to reverse the string.
18. Write an application that finds the substring from any given string using *substring* method and *startsWith* & *endsWith* methods.

19. Write an application that changes any given string with uppercase letters, displays it , changes it back to lowercase letters and displays it.
20. Create a class called *Employee* that includes three pieces of information as instance variables – a first name (type String), a last name (type String) and a monthly salary (double)
21. Write an application that uses *String* method *compareTo* to compare two strings defined by the user.
22. Write an application that uses *String* method *equals* and *equalsIgnoreCase* to tests any two string objects for equality.
23. Write an application that uses *String* method *indexOf* to determine the total number of occurrences of any given alphabet in a defined text.
24. Write an application that uses *String* method *concat* to concatenate two defined strings.
25. Write an application that finds the length of a given string.
26. Write an application that uses *String* method *charAt* to reverse the string.
27. Write an application that finds the substring from any given string using *substring* method and *startsWith* & *endsWith* methods.
28. Write an application that changes any given string with uppercase letters, displays it , changes it back to lowercase letters and displays it.
29. Create a class called *Employee* that includes three pieces of information as instance variables – a first name (type String), a last name (type String) and a monthly salary (double)
 - write application that take a series of integers in pair(e.g. (a,b)) and find the following
 - a. Is first one is greater
 - b. Is both are even
 - c. They are relatively prime or not
30. Write an application that uses *String* method *compareTo* to compare two strings defined by the user.
31. Write an application that uses *String* method *equals* and *equalsIgnoreCase* to tests any two string objects for equality.
32. Write an application that uses *String* method *indexOf* to determine the total number of occurrences of any given alphabet in a defined text.
33. Write an application that uses *String* method *concat* to concatenate two defined strings.
34. Write an application that finds the length of a given string.
35. Write an application that uses *String* method *charAt* to reverse the string.
36. Write an application that finds the substring from any given string using *substring* method and *startsWith* & *endsWith* methods.
37. Write an application that changes any given string with uppercase letters, displays it , changes it back to lowercase letters and displays it.
38. Create a class called *Employee* that includes three pieces of information as instance variables – a first name (type String), a last name (type String) and a monthly salary (double)
39. Create a constructor in above class to initialize the three instance variables. Provide a get method for each instance variable.

40. Write a test application named *Employee Test* that demonstrates class *Employee*'s capabilities. Create two employee objects and display each object's yearly salary.

41. Give each employee a 10% raise and display each *Employee*'s yearly salary again.

Create a class *Account* with an instance variable *balance* (double). It should contain a constructor that initializes the *balance*, ensure that the initial balance is greater than 0.0.



MSc Information Technology 3rd semester
MMIT0305(A)-T Accounting & Financial System

Unit No.	Topics
Unit 1	Introduction to Book Keeping: Meaning , Nature, development, Objectives, merits and difference between Book Keeping and Accountancy. Fundamentals of Accounting: Accounting concepts and conventions. Brief Introduction to GAAP and its importance. Accounting Structure : The Process of Accounting –Journal, Ledger, Subsidiary books.
Unit 2	Trial Balance based on Double Entry Book Keeping System. Financial Systems and related concepts : Form and preparation of Income statements (P & L A/C), Statement of Financial Position.
Unit 3	Methods of Depreciation – SLM Method and WDV method. Financing Decisions : Tools of Financial Analysis : Financial Statement Analysis, Statement of Financial position.
Unit 4	Break Even Analysis. Leverages : operating , financial and combined. Accounting Package – Tally (Operations)
Unit 5	Inventory Management and Responsibility Accounting : Methods of Inventory Management and Material Issues. Responsibility Accounting _ Meaning , Objectives and Importance.

Required Text(s) :

1. Tulsian's Accountancy for Class XI, Financial Management by Khan &Jain.

Reference books :

1. Financial Accounting by TS Grewal.
2. Financial Management by Khan and Jain.
3. NCERT Books on Accounting and FM for Class XI and XII.

MSc Information Technology 3rd semester
MMIT0305(B)-T Mobile Application Development

Unit No.	Topics
Unit 1	Introduction to Mobile Development:- Overview of mobile development platforms (iOS, Android), Introduction to mobile development frameworks (React Native, Flutter), Basics of mobile UI/UX design principles, Setting up development environments (Xcode, Android Studio).
Unit 2	Native Mobile Development:- Introduction to native mobile development (iOS and Android), Understanding iOS development using Swift and Xcode, Understanding Android development using Java/Kotlin and Android Studio, Building basic user interfaces and handling user input.
Unit 3	Cross-Platform Development with React Native:- Introduction to React Native framework, Building mobile applications using React Native, Handling navigation, state management, and data fetching in React Native, Testing and debugging React Native applications.
Unit 4	Cross-Platform Development with Flutter:- Introduction to Flutter framework, Building mobile applications using Flutter, Understanding Flutter widgets and layout systems, State management and asynchronous programming in Flutter.
Unit 5	Advanced Topics in Mobile Development:- Integrating APIs and third-party libraries in mobile applications, Implementing authentication and authorization mechanisms, Handling device features like camera, location, and sensors, Deploying mobile applications to app stores and managing updates.

Reference Book:

"Mobile Application Development: React Native and Flutter" by John Smith, "iOS Programming: The Big Nerd Ranch Guide" by Joe Conway and Aaron Hillegass "Android Programming: The Big Nerd Ranch Guide" by Bill Phillips and Chris Stewart "iOS Programming: The Big Nerd Ranch Guide" by Joe Conway and Aaron Hillegass "Android Programming: The Big Nerd Ranch Guide" by Bill Phillips and Chris Stewart, "Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter" by Robert Anderson, "Mastering Mobile App Development: Build robust and scalable mobile apps with React Native, Flutter, and native development" by Willia Smith.

MSc Information Technology 4th semester
MMIT0401-T LINUX / UNIX ADMINISTRATION

Unit No.	Topics
Unit 1	Background: Evolution of Unix OS. Unix implementations. Features of Unix operating system. Linux operating system: Development of Linux. Applications of Linux operating system.
Unit 2	Basic UNIX environment: Basic commands, directory management, pipes, tee, I/O redirection and other utilities. Advanced commands: File system and process management commands, Shell, Pattern matching, Navigating the File Systems.
Unit 3	Unix editor: VI editor, Creating new files. Text addition, deletion and changes. Dealing with sentences and paragraphs. Searching. Cut, paste and copy. Running C/C++ programs. Shell programming: Features of shell. Shell variables. Control statements. Advance shell programming: Command line arguments. Interactive shell scripts. Debugging of shell scripts. Communication facilities in Unix.
Unit 4	Structure of unix operating system: Structure of unix kernel, Unix system calls. Unix system: File system calls, Process management calls. Advance Filter: Awk: Number processing, Interface with shell, functions.
Unit 5	Unix system administration: Adding and removing users. User accounting. Adding and removing hardware. Performing backups and restore. Disk space management. Unix system administration: Configuring the kernel. Network management in Unix. Performance analysis. Unix Desktop.

Text Book :

UNIX Operating Systems: Sumitabh Das, Tata McGraw Hills publication.

1. Reference books :

1. **UNIX System Administration Handbook**(Second edition): Evi Nemeth, Garth Synder, Scott Seebass, Trent R Hein, Pearson Education - Asia, 2000.
2. **C: Design of UNIX Operating System:** Maurice J. Back, Pearson Education - Asia.

LAB Assignments:

Aim: The aim of this laboratory is to get students acquainted with Unix/Linux commands, shell programming & system administration.

LAB -1 Information management

Cal, date, tty, sh, env, set man ,who, whoami

Basic Operations

- i. Connecting to the system
- ii. Disconnecting from the system
- iii. Text and graphic mode
- iv. Changing your password
- v. Navigating through the file system
- vi. Determining file type

- vii. Looking at text files
- viii. Finding help
- ix. List the different types of file comparisons command.

LAB- 2 Utility commands

Wc,eco,tail,less,more,short,grep,bc,cmp,comm.

LAB-3 File system management

Ls,ln,rn,rmdir,mkdir,file,chmod,find,od,pwd,locate,updated,mount,umount,mv

LAB- 4 Process management

Ps,kill

LAB- 5 Compilation and debugging

cc, gdb

LAB- 6 Editor

Vi,joe,mcedit,emacs

LAB -7 Using unix commands –

- a) Count the number of users who logged in and display the result.
- b) Count the number of files in the current directory
- c) Count the number of differences(in term of lines) between two files

Shell programming

LAB- 8

- 1. Execute the following shell variables
\$HOME
\$PATH
- 2. concatenate two shell variables

LAB -9

- 1. Write a shell script to find the sum of first N natural numbers
- 2. Write a shell script to read n numbers from the user and display its sum.
- 3. Write a shell script to read 10 numbers from the user and to find the sum and Average of the numbers.

LAB-10

- 1. Write a shell script to generate even numbers and to calculate its sum .
- 2. Write a shell script for swapping two numbers (using only two variables) .
- 3. Write a shell script to find the count and sum of even and odd numbers separately.

LAB-11

1. Write a shell script to find the factorial of a given number.
2. Write a shell script to find the sum of digits in an integer .
3. Write a shell script to count the number of digits in an integer .

LAB-12

1. Write a shell script to find whether the given number is Armstrong or not .
2. Write a shell script to print the multiplication table of the given number .
3. Write a shell script to print all divisors of a given integer .

LAB-13

1. Write a shell script to generate Fibonacci series.
2. Write a shell script to find the length of the string.
3. Write a shell script to extract a sub string from a given string.

MMIT0401-P LINUX / UNIX ADMINISTRATION LAB

1. User and Group Management: Understanding how to create, modify, and delete users and groups, manage their permissions, and handle password policies.
2. File System Management: Knowledge of different file systems (ext4, XFS, etc.), partitioning, mounting, and managing disks and file systems, including tools like fdisk, mkfs, mount, and df.
3. Package Management: Familiarity with package managers like apt, yum, or dnf for installing, updating, and removing software packages, as well as managing repositories.
4. Network Configuration: Configuring network interfaces, setting up static or dynamic IP addresses, managing routing tables, and troubleshooting network connectivity using tools like ifconfig, ip, netstat, and traceroute.
5. Security: Implementing security measures such as setting up firewalls (iptables, firewalld), managing permissions and ownership of files and directories, configuring SELinux or AppArmor, and keeping systems updated with security patches.
6. System Monitoring and Performance Tuning: Using tools like top, htop, sar, and vmstat to monitor system performance, identify bottlenecks, and optimize system resources.
7. Backup and Recovery: Implementing backup strategies using tools like rsync, tar, or backup software, scheduling backups, and testing recovery procedures to ensure data integrity.
8. Shell Scripting: Writing shell scripts (bash, sh) to automate common tasks, perform system maintenance, or create custom utilities.
9. Virtualization and Containers: Understanding concepts of virtualization (KVM, VirtualBox) and containerization (Docker, Kubernetes), and managing virtual machines or containers for deploying applications.
10. Logging and Log Analysis: Configuring and managing system logs (syslog, journald), analyzing log files for troubleshooting, monitoring system activities, and detecting security incidents.

MSc Information Technology 4th semester**MMIT0402-T Data Mining & Data Warehousing**

Unit No	TOPIC
Unit 1:	Introduction: Data Warehouse, Evolution, Definition, Very large database, Application, Multidimensional Data Model, OLTP vs Data Warehouse, Warehouse Schema, Data Warehouse Architecture, Data Warehouse Server, Data Warehouse Implementation, Metadata, Data Warehouse Backend Process: Data Extraction, Data Cleaning, Data Transformation, Data Reduction, Data loading and refreshing. ETL and Data warehouse, Metadata
Unit 2	Structuring/Modeling Issues, Derived Data, Schema Design, Dimension Tables, Fact Table, Star Schema, Snowflake schema, Fact Constellation, De-normalization, Data Partitioning, Data Warehouse and Data Marts. SQL Extensions, PLSQL. OLAP, Strengths of OLAP, OLTP vs OLAP, Multi-dimensional Data, Slicing and Dicing, Roll-up and Drill Down, OLAP queries, Successful Warehouse, Data Warehouse Pitfalls, DW and OLAP Research Issues, Tools.
Unit 3	Fundamentals of data mining, Data Mining definitions, KDD vs Data Mining, Data Mining Functionalities, From Data Warehousing to Data Mining, DBMS vs DM, Issues and challenges in Data Mining. Data Mining Primitives, Data Mining Query Languages. Data Mining applications-Case studies.
Unit 4	Association rules: Methods to discover association rules. Various algorithms to discover association rules like A Priori Algorithm. Partition, Pincer search, Dynamic Itemset Counting Algorithm etc. Cluster Analysis Introduction : Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning Algorithms, Hierarchical and Categorical clustering,
Unit 5	Decision Trees, Neural networks, Genetic Algorithm. Web Mining , Web content mining, Web Structure mining, Text mining, Temporal Data Mining, Spatial Data Mining

Text Book:

1. Data Mining Techniques – ARUN K PUJARI, University Press

Essential References:

1. Data Mining – Concepts and Techniques - JIAWEI HAN & MICHELINE KAMBER Harcourt India.
2. Building the Data Warehouse- W. H. Inmon, Wiley Dreamtech India Pvt. Ltd..
3. The Data Warehouse Life cycle Tool kit – RALPH KIMBALL WILEY STUDENT EDITION
4. Data Warehousing in the Real World – SAM ANAHORY & DENNIS MURRAY. Pearson Edn Asia.
5. Data Warehousing Fundamentals – PAULRAJ PONNAIAH WILEY STUDENT EDITION
6. Data Mining Introductory and advanced topics –MARGARET H DUNHAM, PEARSON EDUCATION

ASSIGNMENTS:

Lab work comprising of mini projects using freeware data mining tools.

1. Search a voluminous data file and understand it.(hint: you may get free data from internet)
2. Replace all tabs with commas from file or vice versa.
3. Normalize the data: for each value, set the minimum value to 0 and the maximum to 100.
4. Transform the data file (text, excel etc) into database.
5. Create a subject oriented data warehouse for any business.
6. Use of free ETL tool.
7. Review of data mining tools, applications, and algorithms.

MMIT0402-P Data Mining & Data Warehousing LAB

1. Introduction to Association Rule Mining:
 - Brief overview of association rule mining and its applications in data mining.
 - Introduction to the Apriori algorithm for discovering frequent itemsets and association rules.
2. Understanding the Apriori Algorithm:
 - Explain the key concepts of support, confidence, and frequent itemsets.
 - Describe the iterative process of generating candidate itemsets and pruning infrequent ones in the Apriori algorithm.
3. Dataset Selection and Preprocessing:
 - Choose a suitable dataset for association rule mining (e.g., retail transactions, market basket data).
 - Preprocess the dataset by converting it into a transactional format suitable for Apriori algorithm implementation.
4. Implementing the Apriori Algorithm:
 - Write Python code to implement the Apriori algorithm for discovering frequent itemsets.
 - Implement functions for generating candidate itemsets, calculating support, and pruning infrequent itemsets.
5. Mining Association Rules:
 - Extend the Apriori implementation to generate association rules from frequent itemsets.
 - Implement functions to calculate confidence and filter association rules based on user-defined thresholds.
6. Evaluation and Analysis:
 - Evaluate the performance of the implemented Apriori algorithm on the selected dataset.
 - Analyze the generated association rules based on support, confidence, and interestingness measures.
7. Parameter Tuning and Optimization:
 - Experiment with different parameter settings (e.g., minimum support, minimum confidence) to tune the Apriori algorithm.
 - Optimize the implementation for efficiency by reducing redundant calculations and optimizing data structures.
8. Visualization and Interpretation:
 - Visualize the discovered frequent itemsets and association rules using plots or interactive visualizations.
 - Interpret the results to extract meaningful insights and patterns from the dataset.
9. Comparison with Other Algorithms:
 - Compare the performance of the Apriori algorithm with other association rule mining algorithms (e.g., FP-growth).

- Discuss the strengths and limitations of each algorithm and their suitability for different types of datasets.

10. Conclusion and Discussion:

- Summarize the key findings and insights obtained from association rule mining on the dataset.
- Discuss the practical applications and business implications of the discovered association rules.
- Reflect on the challenges faced during implementation and potential areas for future research or improvement.



MSc Information Technology 4th semester**MMIT0403-T Internet and Web Technology**

Unit No.	Topics
Unit 1	Introduction Dynamic Web Programming, HTML Forms, scripting languages, Introduction to HTTP, web Server and application Servers, Installation of Application servers, Configuration files, Web.xml. Java Servlet, Servlet Development Process, Deployment Descriptors, The Generic Servlet Lifecycle.
Unit 2	Servlet Packages, Classes, Interfaces, and Methods, Handling Forms with Servlets. Various methods of Session Handling. Various elements of deployment descriptors. Java Database Connectivity: various steps in process of connection to the database, Various type of JDBC Driver.
Unit 3	Connection of JSP and Servlet with different database viz. Oracle, MS-SQL Server, MySQL. java.sql Package. Accessing metadata from the database. Type of Statement, Connection pooling: multiple users and need of connection pooling.
Unit 4	JSP Basics: JSP lifecycle, Directives, scripting elements, standard actions, implicit objects. Writing JSPs. Expression Language (EL), Separating Business Logic and Presentation Logic, Building and using JavaBean.
Unit 5	Session handling in JSP, Types of errors and exceptions handling, Standard Tag Library in JSP, Building Custom Tag Library, JSP Tag Library, MVC Design pattern Advances in J2EE and Other Web technology

1. Text Book :

Kevin Mukhar, Chris Zelenak, James L Weaver, —Beginning Java EE 5: From Novice to Professional Apress

2. Reference books :

1. Marty Hall, Larry Brown, —Core Servlets and Java Server Pages, 2nd edition, Pearson Education
2. JavaDoc for various technologies
3. Internet and Web technologies, TMH, 2002

Class Assignments:

TOPIC

Introduction Dynamic Web Programming, HTML Forms, scripting languages, Introduction to HTTP, web Server and application Servers, Installation of Application servers, Configuration files, Web.xml.

Java Servlet, Servlet Development Process, Deployment Descriptors, The Generic Servlet Lifecycle.

Lab Assignments:

1. Write a servlet that prints "Hello World"
2. Write a servlet that knows to whom it's saying hello, This servlet must be called from an HTML page taking user name as input. (Use both get and post method)
3. Write a servlet that counts and displays the number of times it has been accessed since the last server reboot.
4. Write a servlet that counts the times it has been accessed, the number of instances created by the server, and the total times all of them have been accessed.
5. Write a servlet that counts and displays the number of times it has been accessed, and reads an init parameter to know what at what number to begin counting.
6. This servlet counts and displays the number of times it has been accessed, and saves the count to a file in its destroy() method to make the count persistent.
7. Write a servlet that searches for prime numbers above one quadrillion. The algorithm it uses couldn't be simpler: it selects odd-numbered candidates and attempts to divide them by every odd integer between 3 and their square root. If none of the integers evenly divides the candidate, it is declared prime. It's disabled to let

MAHAKAUSHAL UNIVERSITY
JALALPUR

ज्ञानं महकाशालम्

the server's CPU handle important tasks.

8. Write a servlet that prints the name and value for all of its init parameters.
9. Write a servlet that displays information about its server (The process is called Snooping).
10. Write a servlet that snoops the server's servlet and Java version.

Servlet Packages, Classes, Interfaces, and Methods, Handling Forms with Servlets.Lab Assignments:

11. Write a servlet to compute arithmetic operations on numbers and strings as follows:

First val: Enter a name:

Second val: Length:

Result:

Add Subtract Multiply Divide Compute Length

Reset

12. Write a servlet that checks the client machine and only allows access if the client appears to be coming from somewhere other than the Terrorist 7 countries. A servlet like this could help restrict the export of strong encryption.
13. Write a servlet that prints its query string and then prints the name and value for all its parameters.
14. Write a servlet that prints the extra path information it receives and the resulting translation to a real path.
15. Write a servlet that serves files by using the `getPathTranslated()` and `getMimeType()` methods to return whatever file is given by the extra path information.
16. Write a servlet that prints information about its HTTP request headers.
17. Write a servlet that performs a random redirect, sending a client to a random site selected from its site list. Depending on the site list, a servlet like this could have many uses. As it stands now, it's just a jump-off point to a selectino of cool servlet sites. With a site list containing advertising images, it can be used to select the next ad banner.
18. Redirections can be used to learn where clients go when they leave your site. Assume you have several pages containing lists of links to other sites. Instead of linking directly to the external site, you can link to a redirecting servlet that can record each time an external link is selected. The HTML looks like this:

```
<a href="/goto/http://www.servlets.com">Servlets.com</a>
```

 Write a servlet that can be registered to handle the `/goto/*` path prefix where it will receive the selected URL as extra path info and redirect the client to that location after making a note in the server log. servlet shows a servlet that uses client pull to display the current time, updated every 10 seconds.
19. Write a servlet that shows a servlet that uses client pull to display the current time, updated every 10 seconds.
20. Write a servlet that redirects requests for one host to another host, giving an explanation to the client before the redirection.
21. Write a servlet that demonstrates session tracking using hidden form fields by displaying the shopping cart for a booksite.

Various methods of Session Handling. Various elements of deployment descriptors.Lab Assignments:

22. Write a servlet that uses session tracking to count the number of times a client has accessed it.
23. Write a servlet that demonstrates how to programmatically alter the current timeout. On first execution, the current timeout displays the application-wide setting. On second execution, the current timeout displays two hours—because that's the timeout set during the first execution.

24. Write a servlet that snoops all the information about the current session.
25. Write a servlet that is protected by BASIC authentication as shown in web.xml and tomcat-users.xml. To
26.
27. see the salary information you'll need to login as a "manager" using names and passwords in tomcat-



users.xml.

26. Write a servlet that lets a user vote for his favorite food from a combo box or radio buttons (the user must be able to make multiple food selections per request).
- Store the favorite foods and the number of votes for each food.
 - Display all foods and their number of votes in alphabetical order back to the user.
 - Use an appropriate Collection class or Map class to store the data.

Java Database Connectivity: various steps in process of connection to the database, Various type of JDBC Driver

Connection of JSP and Servlet with different database viz. Oracle, MS-SQL Server, MySQL. java.sql Package. Accessing metadata from the database.

Lab Assignments:

27. Write a servlet that establishes a database connection using the values stored within the its sql.properties file having the following contents.

```
connection.driver=sun.jdbc.odbc.JdbcOdbcDriver
connection.url=jdbc:odbc:somedb
user=user
password=passwd
```

28. Write a servlet that uses the Oracle JDBC driver to perform a simple query, printing names and phone numbers for all employees listed in a database table. Here assume that the database contains a table named EMPLOYEES, with at least two fields, NAME and PHONE.

Type of Statement, Connection pooling: multiple users and need of connection pooling.

JSP Basics: JSP lifecycle, Directives, scripting elements, standard actions, implicit objects. Writing JSPs.

Lab Assignments:

29. Write a JSP to output the values returned by System.getProperty for various system properties such as java.version, java.home, os.name, user.name, user.home, user.dir
30. Write a JSP to output the entire line, "Hello! The time is now ..." but use a scriptlet for the complete string, including the HTML tags.
31. Write a JSP to output all the values returned by System.getProperties with "
" embedded after each property name and value. Do not output the "
" using the "out" variable.
32. Modify exercise # 28-30 to import the java.util packages.
33. Write a JSP to do either a forward or an include, depending upon a boolean variable.
34. Write a JSP/HTML set that allows a user to enter the name of a system property, and then displays the value returned by System.getProperty for that property name (handle errors appropriately.)
35. Make a JSP page that randomly selects a background color for each request. Just choose at random among a small set of predefined colors. Be sure you do not use the JSP-Styles.css style sheet, since it overrides the colors given by
<BODY BGCOLOR="...">.
36. Make a JSP page that lets the user supply a request parameter indicating the back-ground color. If no parameter is supplied, a background color should be selected at random.

Expression Language (EL), Separating Business Logic and Presentation Logic, Building and using JavaBean.
Lab Assignments:

37. Make a JSP page that lets the user supply a request parameter indicating the back-ground color. If no parameter is supplied, the most recently used background color (from a previous request by any user) should be used.
38. The java.math package has a class called BigInteger that lets you create whole numbers with an arbitrary number of digits. Create a JSP page that makes a large BigInteger from a String you supply as a request parameter, squares it, and prints out the result. Use the online API at <http://java.sun.com/j2se/1.5.0/docs/api/> to see the syntax for the BigInteger constructor and squaring operations.



39. Make an HTML `<signature>` block with your name and email address. Include it in two JSP pages.
40. The value of the page attribute of `jsp:include` is allowed to be a JSP expression. Use this idea to make a JSP page that includes a `<good news>` page or a `<bad news>` message at random.
41. Suppose that you have two different JSP pages that do two different things. However, for both pages you want to let the user supply a `bgColor` attribute to set the background color of the page. Implement this, but use an include mechanism to avoid repeating code. For example:

White background: `http://host/path/page1.jsp`

White background: `http://host/path/page2.jsp`

Red background: `http://host/path/page1.jsp?bgColor=RED`

Yellow background: `http://host/path/page2.jsp?bgColor=YELLOW`

For testing, I do not care if you write an HTML form to collect the `bgColor` parameter or if you simply attach it onto the end of the URL —by hand.

Session handling in JSP, Types of errors and exceptions handling, Standard Tag Library in JSP, Building Custom Tag Library

Lab Assignments:

42. Make two separate JSP pages that have bulleted lists containing random integers in a certain range. Avoid repeating code unnecessarily by including a page that defines a `randomInt` method.
43. Define a class called `ColorBean` that stores strings representing a foreground color and a background color. Compile and test it separately (i.e., without using a servlet or JSP page). Note: if your tester class (i.e., the one that has `public static void main(String[] args) {...}` in it) is in a package, remember that you have to use the package name when you run it from the command line. That is, you have to do `javac BeanTester.java` and then `java yourPackage.BeanTester`.
44. Make a `<color preference>` form that collects the user's preferred foreground and background colors. Send the data to a JSP page that displays some message using those colors. This JSP page should use a default value for any form value that the user fails to supply (but don't worry about empty strings). So, for example, if the user goes directly to the JSP page (bypassing the form), the JSP page should still work fine. For now, don't worry about the user sending you whitespace; just handle totally missing values.
45. Redo the color preference example, but if the user fails to supply either of the colors, use whatever value they gave last time. If you have no previous value, use a default. (Hint: this problem is almost exactly the same difficulty as the previous one.)
46. Redo the color preference example, but if the user fails to supply any of the parameters, use whatever color the most recent user gave last time. Why could this give bad results?
47. Write a JSP that takes the user's name and age from a form.
- Echo back the name and age along with a message stating the price of movie tickets.
 - The price is determined by the age passed to the JSP.
 - If the age is greater than 62, the movie ticket price is \$7.00.
 - If the user is less than 10 years old, the price is \$5.00.
 - For everyone else, the price is \$9.50.
48. Write a JSP that will allow a user to enter two values, select a type of mathematical operation to apply against them, and then upon clicking Submit, will display the result of the operation. An example of the default entry form:
- 49.

Enter two values, choose an operation, and click submit.

5

5

Addition
 Subtraction
 Multiplication
 Division

Submit

And the page that displays upon Submit:

5 times 5 equals 25

A valid code must also have:

- one *jsp* - *Math.jsp* - that submits to itself
- four types of operations - add, subtract, multiply, divide
- display some kind of error if the user attempts to divide by 0

50. Create a database table to store contact information, then write a web application to manage viewing, adding, and deleting contacts from that table. The view should look similar to:

A list of all contacts.

	action	id	first name	last name	phone number	city	state	zip
Delete	2	Laura	Juliano	555-1212	Urbandale	IA	50322	
Delete	3	Bob	Jones	555-1212	Des Moines	ia	50265	

[Insert a new Contact](#)

Some items of note from the view:

1. There is a delete link next to each record which triggers a delete servlet and will delete that record in the database before redirecting back to the view
 2. There is an 'Insert a new Contact' link at the bottom of the page that directs the user to *ContactsInsert.html*
- The insert page should look similar to:

Insert a contact

First Name:

Last Name:

Phone Number:

City:

State:

Zip:

An item to note about insert: once the user clicks submit, the form directs to a `ContactsInsertServlet.java`, which insert a record into the database then redirects to the view.

A description of each file needed for this exercise:

1. Write a sql script that creates a `Contacts` table in derby that holds text values for `id`, `firstName`, `lastName`, `phoneNumber`, `city`, `state`, and `zip`. Add to the script a couple of `INSERT` statements so that the table will not be empty. Run your script in Squirrel to confirm it works, and so you have the table on hand for the rest of the exercise. Save the script in the root of the web application as `Contacts.sql`.
2. `ContactsViewServlet.java` - displays an html table containing all records in the table. Also displays links to `ContactsDeleteServlet.java` and `ContactsInsert.html`.
3. `ContactsInsert.html` - have text fields for each field in the database. On submit, execution moves to `ContactsInsertServlet.java`.
4. `ContactsInsertServlet.java` - uses an `INSERT` statement to add a record to the database. After the `INSERT` has been performed, redirect back to `ContactsViewServlet.java`.
5. `ContactsDeleteServlet.java` - uses a `DELETE` statement to records from the database. After the `DELETE` has been performed, redirect back to `ContactsViewServlet.java`.
6. `web.xml` - must have valid references to `ContactsViewServlet.java`, `ContactsInsertServlet.java`, and `ContactsDeleteServlet.java`.

Internet and Web Technology lab:-

1. HTML Basics and Web Page Creation:

- Introduction to HTML syntax and structure.
- Creating simple web pages with text, images, links, and basic formatting.

2. CSS Styling and Layout:

- Understanding CSS syntax and selectors.
- Applying CSS styles to HTML elements for formatting and layout.
- Creating responsive web designs using CSS media queries.

3. JavaScript Basics and DOM Manipulation:

- Introduction to JavaScript programming language.
- Interacting with the Document Object Model (DOM) to dynamically update web page content.
- Implementing event handling and form validation using JavaScript.

4. AJAX and Asynchronous Web Development:

- Understanding Asynchronous JavaScript and XML (AJAX) for building dynamic web applications.
 - Making asynchronous HTTP requests to fetch data from the server without page reloads.
 - Implementing AJAX features such as auto-complete search, live updates, and infinite scrolling.
5. Server-Side Scripting with PHP or Node.js:
- Introduction to server-side scripting languages like PHP or Node.js.
 - Setting up a local web server environment.
 - Writing server-side scripts to handle form submissions, process data, and generate dynamic web content.
6. Database Connectivity with MySQL or MongoDB:
- Connecting web applications to relational databases (MySQL) or NoSQL databases (MongoDB).
 - Performing database operations such as insertion, retrieval, update, and deletion using server-side scripts.
 - Implementing user authentication and access control with database integration.
7. Web Application Frameworks (e.g., Laravel, Express.js):
- Exploring popular web application frameworks for rapid development.
 - Building CRUD (Create, Read, Update, Delete) applications using frameworks like Laravel (PHP) or Express.js (Node.js).
 - Understanding MVC (Model-View-Controller) architecture and routing in web frameworks.
8. Web Security Best Practices:
- Introduction to web security threats and vulnerabilities.
 - Implementing security measures such as input validation, output escaping, and secure authentication.
 - Protecting against common web attacks like SQL injection, XSS (Cross-Site Scripting), and CSRF (Cross-Site Request Forgery).
9. Responsive Web Design and Mobile Optimization:
- Designing web pages that adapt to different screen sizes and devices.
 - Using CSS frameworks like Bootstrap or Foundation for responsive layout and component styling.
 - Testing and optimizing web applications for mobile compatibility and performance.
10. Web Hosting and Deployment:
- Deploying web applications to a live server environment.
 - Configuring domain names, DNS settings, and SSL certificates for secure HTTPS connections.
 - Managing web hosting services, server administration, and troubleshooting deployment issues.

MSc Information Technology 4th semester**MMIT0404-T Mobile Communication and Computing Systems**

Unit No.	Topics
Unit 1	<p>Mobile Communications: An Overview, Guided Transmission, Unguided Transmission, Signal Propagation Frequencies, Antennae, Propagation of Signals, Modulation, Modulation Example, Multiplexing, Multiplexing Example, Circuit and Packet Switching, Circuit Switching Example, Packet Switching Example, Signal Processing, Voice-oriented Data Communication Standards, GSM, CDMA, Wireless Personal Area Network, Wireless Local Area Network 802.11 Wi-Fi, 802.11n and Internet Access, Near-Field Communication (NFC), 3G Network , 2G vs. 3G vs. 4G, Current 3G Networks,</p> <p>Mobile Computing, Novel applications, Smartphones, Enterprise Solutions, Music and Video, Mobile Cheque, Mobile Commerce,</p> <p>Limitations of Mobiles, Mobile Computing Architecture, Programming Languages, Functions of Operating Systems, Functions of Middleware for Mobile Systems, Mobile Computing Architecture Layers, Protocols, Layers</p> <p>Mobile Devices, Handheld Mobile Phones, Handheld Mobile Computers, Mobile System Networks, Cellular Network, WLAN Wi-Fi 802.11 Networks and Mobile IP, Ad hoc Networks, Mobility System Security</p>
Unit 2	<p>Wireless Medium Access Control, CDMA, 3G and 4G Communication</p> <p>Medium Access Control, Medium Access by Modulation, Controlling Medium Access, SDMA, TDMA, and FDMA, Introduction to CDMA-based Systems, Code Division Example, Spread Spectrum in CDMA Systems, Direct Sequence Spread Spectrum, Direct Sequence Spread Spectrum Example, Frequency Hopping Spread Spectrum, Frequency Hopping Spread Spectrum Example, Spread Factor, OFDM,</p> <p>3G Network (HSPA, HSPA+, WiMax 802.16e, Ultra-Wide-Band (UWB), 4G Networks,</p> <p>4G Network Requirements and Design, Modulation techniques required for 4G Networks, High Speed OFDM Packet Access (HSOPA), Pre-4G technology 3GPP Long Term Evolution (LTE), LTE Advanced, WiMax Advanced 802.16M</p>
Unit 3	<p>Cyclic repetition of data, Digital Audio Video Broadcasting, Multimedia object transfer Protocol, Wireless LAN topologies, requirements.</p> <p>Physical layer, MAC layer, IEEE802.11.HIPERLAN: Protocol architecture, layers, Information bases and networking, Bluetooth.</p> <p>Case Study on Wireless LAN infrastructure .Basics of Discrete Event Simulation, Application and Experimentation, Simulation models.</p> <p>Case Study on Performance Evolution of IEEE 802.11 WLAN configuration Using Simulation.</p>
Unit 4	<p>Mobile Application Languages-I -XML, Java, J2ME, and Java Card</p> <p>Applications - Workflow for Application Development, Techniques for Composing Applications Mobile Application Languages</p> <p>XML(Document Type Definitions and Parsing of XML Documents, Models of an X JAVA, Characteristic Features, Classes and Beans, Java 2 Enterprise Edition (J2EE), Java 2 Micro Edition (J2ME), Profiles, Configurations, Java Card,</p>

	XML and Java Program Examples in Mobile Computing ML Document and Parsing, SAX Model of an XML Document, DOM Model of anXML Document, Applications of Parsed Information and Data, XML-based Standards and Formats for Applications,
Unit 5	Operating System, Process, Task, Thread, ISR, and IST, Mobile Operating System, Windows CE and Windows 7, Dot Net Framework, Programming Examples, Symbian 3 /4, Linux for Mobile Devices (Android 2.2) Programming Examples

Required Text :

Mobile Communications author Jochen Schiller, publication John Willy & Sons, Ltd.

Mobile Computing, Oxford University Press

Mobile Computing Principles author Reza B'Far, publication Cambridge University Press.

Reference books :

Wireless And Mobile Systems author D P Agrawal & Qing-An zeng,

Wireless Networks author P Nicopotidis, publication Addison –Wesley.

Mobile Wireless Communications author Mischa Schwartz, publication Cambridge University Press.

MAHAKAUSHAL UNIVERSITY
JABALPUR

ज्ञानं महकाशालम्