Database Management Systems
68251102 (DC)

Objectives:

- To describe key concepts, issues, and operational terminology
- To understand the relationships of key components behind concepts such as hardware, networks, data storage, operating systems, and software programs.
- To normalize any problem using 1^{st},2^{nd},3^{rd},4^{th},5^{th} normal form

UNIT-I

Introduction: Advantage of DBMS approach, various view of data, data independence, schema and sub-schema, primary concepts of data models, Database languages, transaction management, Database administrator and users, data dictionary, overall system architecture.

ER model: basic concepts, design issues, mapping constraint, keys, ER diagram, weak and strong entity sets, specialization and generalization, aggregation, inheritance, design of ER schema, reduction of ER schema to tables.

UNIT-II

Domains, Relations and Keys: domains, relations, kind of relations, relational database, various types of keys, candidate, primary, alternate and foreign keys.

Relational Algebra & SQL: The structure, relational algebra with extended with extended operations, modifications of Database, idea of relational calculus, basic structure of SQL, set operations, aggregate functions, null values, nested sub queries, derived relations, views, modification of Database, join relations, DDL in SQL.

UNIT-III

Functional Dependencies and Normalization: basic definitions, trivial and non trivial dependencies, closure set of dependencies and of attributes, irreducible set of dependencies, introduction to normalization, non loss decomposition, FD diagram, first, second, third Normal forms, dependency preservation, BCNF, multivalued dependencies and fourth normal form, Join dependency and fifth normal form.

UNIT-IV

Transaction, concurrency and Recovery: basic concepts, ACID properties, Transaction states, implementation of atomicity and durability, concurrent executions, basic idea of serializability, basic idea of concurrency control, basic idea of deadlock, failure classification, storage structure types, stable storage implementation, data access, recovery and atomicity- log based recovery, deferred Database modification, immediate Database modification, checkpoints.

UNIT-V

Distributed Database: basic idea, distributed data storage, data replication, data fragmentation- horizontal vertical and mixed fragmentation

Storage structure and file organizations: overview of physical storage media, magnetic disks-performance and optimizations, basic idea of RAID, file organizations, organization of records in files, basic concepts of indexing, ordered indices, basic idea of B-tree and B+-tree organization.

Books:

1. Database System Concepts – A Silberschatz, H.F Korth, Sudersan, MGH Publication.
2. An introduction to Database Systems – C.J Date 6th ed.
3. Fundamentals of Database systems – L elmasri & Navathe III ed.
4. An introduction to Database systems – B.C. Desai.

Course Outcomes:

Student would be able to

CO1: **Design** ER-models to represent simple database application scenarios
CO2: **Apply** relational algebra & SQL to find solutions to a broad range of queries
CO3: **Apply** normalization techniques to improve database design
CO4: **Identify** the issues of transaction processing and concurrency control.
CO5: **Analyze** an information storage problem and derive an information model expressed in the form of an entity relation diagram and other optional analysis forms, such as a data dictionary.

	PO1		PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1
CO1	3		2	3	1			3		2	2	3
CO2	3		3	2	2	1	2		2	3	2	3
CO3		3	3	3			3		3		1	
CO4	2		3		3				3	2	3	2
CO5	3	3	2	3		3	3	3	2	3	3	3

Computer Organization and Architecture
68251103 (DC)

Objectives:

1. To provide the fundamental knowledge of a computer system and its processing units.
2. To introduce basic concepts of computer organization.
3. To understand the architecture of modern computer.
4. To understand different instruction types.
5. To provide the details of input & output operations, memory management and performance measurement of the computer system.

UNIT-I

Representation of Information: Number system, integer and floating-point representation, character codes (ASCII, EBCDIC), Error detection and correction codes. Basic Building Blocks: Boolean Algebra, combinational blocks: gates, multiplexers, decoders etc, Sequential building blocks: flip-flops,, registers, counters, ALU, Random access memory etc.

UNIT-II

Register Transfer Language and Micro-operations: concept of bus, data movement among registers, language to represent conditional data transfer, data movement from/to memory, arithmetic and logical operations along with register transfer, timing in register transfer.

UNIT-III

Architecture of a simple processor: A simple computer organization and instruction set, instruction formats, addressing modes, instruction execution in terms of microinstructions, concepts of interrupt and simple I/O organization, implementation of processor using the building blocks.

UNIT - IV

Memory system: Memory Hierarchy, Semiconductor Memories, RAM (Random Access Memory), Read Only Memory (ROM), Types of ROM, Cache Memory, Performance considerations, Virtual memory, Paging, Secondary Storage, RAID.

UNIT – V

Input output: I/O interface, Programmed IO, Memory Mapped IO, Interrupt Driven IO, DMA.

Multiprocessors: Characteristics of multiprocessors, Interconnection structures, Inter Processor Arbitration, Inter processor Communication and Synchronization, Cache Coherence

BOOKS:

1. Computer System Architecture, Morris Mano, PHI.
2. Computer Organization, Carl Hamacher, THM.
3. Computer Organization and Architecture- designing for performance, William Stallings (2010), 8th edition, Prentice Hall, New Jersy.
4. Structured Computer Organization, Anrew S. Tanenbaum (2006), 5th edition, Pearson Education Inc,
5. Computer Architecture and Organization, J P Hayes, Mc-Graw Hills, New Delhi

Outcomes:

Student would be able to

CO1: **Design** various building blocks.(BL6)
CO2: **Evaluate** register-level operations, data movement, register transfers and micro-operations. (BL5)
CO3: **Demonstrate** concepts of interrupt handling, simple processor architecture, instruction sets, and addressing modes. (BL3)
CO4: **Analyze** different memories and their performance.(BL4)
CO5: **Illustrate** various modes of I/O data transfer, multiprocessor architectures and concepts (BL4)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1
CO1	2		2	2	3		2			3	2
CO2	2	1		1		2	2		2	2	2
CO3	1		2	2			1				
CO4	2	1		1	2	1		1	3	3	2
CO5	2		2			1	2	1		2	

Operating Systems
68251104 (DC)

Objectives:

1. To learn the fundamentals of Operating Systems
2. To understand the services provided by and the design of an operating system.
3. To understand the structure and organization of the file system.
4. To understand what a process is and how processes are synchronized and scheduled.
5. To understand different approaches to memory management.
6. To gain knowledge on Distributed operating system concepts that includes architecture, Mutual exclusion algorithms, Deadlock detection algorithms and agreement protocols
7. To gain insight on to the distributed resource management components viz. the algorithms for implementation of distributed shared memory, recovery and commit protocols
8. To know the components and management aspects of Real time, Mobile operating systems.

UNIT-I

Introduction: Evolution of operating systems, Types of operating systems, Different views of operating system, operating system concepts and structure.

Processes: The concept of process and thread, states of process, operating system services for processes management, Uniproocessor scheduling algorithm (FCFS, SJF, RR, Priority, and Multilevel feedback queue), Performance evaluation.

UNIT-II

Memory Management: Static and dynamic partitioned memory management, paged memory management, segmented memory management, virtual memory, demand paging, page replacement algorithms, modeling paging algorithms, design issues for paging system, segmentation.

UNIT-III

Interprocess communication and synchronization: The need for Interprocess synchronization, mutual exclusion, Dekkar's Algorithm, Petersons's Algorithm, Hardware solution for mutual exclusion, semaphores for mutual exclusion, queuing implementation of semaphores, classical problems in concurrent programming, critical region and conditional critical region, monitors messages.

UNIT-IV

Deadlocks: Deadlock prevention, deadlock avoidance, deadlock detection.

File system: File systems, directories, file system implementation security and protection mechanism.

Input/Output: Principles of I/O Hardware: I/O devices, device controllers, direct memory access, Principles of I/O software: Goals interrupt handlers, device drivers, and device independent I/O software, User space I/O software.

UNIT-V

Disks: Disk hardware, scheduling algorithms(FCFS, SCAN, C-SCAN, SSTF), RAID, Disk Cache.: Error handling, track-at-a time caching RAM disk.
Clocks: clock hardware, memory mapped terminals, I/O software
Distributed file system: Design, implementation and trends.
Performance measurement: monitoring and evaluation introduction, important trends affecting performance issues, why performance monitoring and evaluation are needed, performance measures, evaluation techniques, bottlenecks and saturation, feedback loops.
Case studies: MS-DOS, MS Windows and Linux(Unix) Operating System.

Books:

1. Deitel and Dietel "Operating Systems". Addison Wesley Publishing Company
2. Peter B. Galvin, Greg Gagne, Abraham Silberschatz, "Operating Systems Concepts", John Wiley & Sons, Inc.
3. Tanenbaum A.S. "Modern Operating System" Prentice Hall of India Pvt Ltd

Outcomes:

Student would be able to

CO1: **Analyze** the concept of processes, and processor scheduling
CO2: **Evaluate** various memory management schemes and virtual memory
CO3: **Analyze** the concurrency issues and Mutual exclusion
CO4: Demonstrate the deadlock prevention, avoidance and detection, various file and directory systems
CO5: **Apply** the various resource management in various types of operating ssytems

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1
CO1	2			2		3		3	2	3	2
CO2	3	2	3	1			3		2	2	3
CO3	3	3	2	2	1	2		2	3	2	3
CO4		3	3			3		3		1	
CO5	2	3		3				3	2	3	2

Emerging Technologies in Computer Applications
68251105

Course Objectives:

1. To acquire knowledge of trending and emerging technologies along with their principles, issues, challenges, and mechanisms.
2. To provide a conceptual understanding of modern tools and techniques for Big data Analytics, Artificial intelligence, Cyber Security, and IoT.

Unit I –

Artificial Intelligence: Introduction: Need and Scope of AI, History, Definition of AI, Techniques of AI, Characteristics of AI applications, Basic Search Techniques, General problem solving, Speech Recognition, Natural Language Processing, Computer Vision, Introduction of expert systems

Unit II –

Cloud Computing: Introduction to cloud computing, Software as a service, platform as a service, and infrastructure as a service. Cloud deployment model: Public cloud, Private clouds, Community clouds and Hybrid clouds. Virtualization: Compute virtualization, Storage virtualization, Full and paravirtualization.

Unit III –

Cyber Security: Overview of Cyber Security, Cyber-crime, Cyberwarfare, cyber Terrorism, Cyber espionage, Cyber Vandalism (Hacking), Cyber Stalking, Internet Frauds and Software piracy, Cyber Security Threats and Vulnerabilities: Hacker, Types of Hacker- white, Gray and black, Malicious Software: Virus, Worm, Trojan Horse, Backdoors and Spywares, Sniffers, Denial of Service attack and Phishing.

Unit IV –

Internet of Things: IoT definition, Characteristics, IoT conceptual and architectural framework, Components of IoT ecosystems, Review of Basic Microcontrollers and interfacing, Basic components and challenges of a sensor, Sensor features, RFID: Features & working principle.

Unit V –

Big Data Analytics: Introduction to Big data, Big data characteristics, Traditional data versus Big data, Evolution of Big data, challenges with Big Data, Technologies available for Big Data, Use of Data Analytics, Hadoop Ecosystem, Core Hadoop components, ETL Processing

COURSE OUTCOMES:

After successful completion of the course, the learners would be able to:

CO 1. **Illustrate** concepts & applications of Artificial Intelligence.
CO 2. **Describe** the fundamental ideas behind Cloud Computing, the evolution of the paradigm, its applicability benefits, as well as current and future challenges;
CO 3. **Analyze** various Cyber Security Threats and Vulnerabilities
CO 4. **Discuss** the Internet of Things and its hardware and software components.
CO 5. **Define** the concept and challenges of Big Data, along with the basic understanding of Big Data Solutions using the Hadoop Eco System

Books:-

• RadhaShankarmani, M. Vijaylakshmi, " Big Data Analytics", Wiley, Second edition
• Rich & Knight - Artificial Intelligence • Kai Hawang, Geoferrey C Fox, “Distributed and Cloud Computing”
• Cryptography and Network Security Principles and Practice Fourth Edition, William Stallings, Pearson Education
• Cuno Pfister, “Getting Started with the Internet of Things”, O Reilly Media

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1
CO1	3			3				2		2	
CO2			3		2		2			3	
CO3				2	3				2	2	3
CO4	3	2								3	
CO5			3			3	3	3	2	2	2

Database Management Laboratory
68251106
(DLC-1)

Basic structure of SQL, set operations, aggregate functions, null values, nested sub queries, derived relations, views, modification of Database, join relations, DDL in SQL. Creation of a database and writing SQL queries to retrieve information from the database. Performing insertion, deletion, modifying, altering, updating and viewing records based on conditions. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints. Creation of Views, Creating an Employee database to set various constraints. Creating relationships between the databases. Case Study using real life database applications.

List of Experiments

1. Introduction to Structure Query Language (SQL)
a) Overview of SQL b) Various Data Types in SQL
c) Various Commands in SQL d) Various Constraints in SQL
2. Implementation of DDL commands of SQL with suitable examples
a) Create table
b) Alter table
c) Drop table
3. Implementation of DML commands of SQL with suitable examples
a) Insert
b) Update
c) Delete
4. Study and implementation of different types of constraints.
5. Implementation of different types of functions with suitable examples
a) Number function
b) Aggregate function
c) Character function
6. Implementation of different types of operators in SQL
a) Arithmetic operators
b) Logical operators
c) Comparison operators
d) Set operation
7. Implementation of different types of joins
a) Inner join
b) Outer join
c) Natural join
8. Study and implementation of
a) Group by and having clause
b) Order by clause
c) Indexing

9. Study and implementation of
a) Sub-queries
b) Views
10. Study and implementation of Database Backup and Recovery commands.

Course outcomes:

Student would be able to

CO1: **Convert** the ER-model to relational tables, populate relational databases and formulate SQL queries on data.
CO2: **Adapt** normalization theory for a database.
CO3: **Implement** data definition language for the schema using a DBMS
CO4: **Construct** database application system solutions to a broad range of query problems.

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1
CO1		3				3				3	
CO2				2	3				2	2	3
CO3	3	2								3	
CO4			3			3	3	3	2	2	2

Object Oriented Programming Lab
68251107
(DLC-2)

Data Types; Constant & Variable; Operators & expressions. Priority & associativity of operators.

Control Constructs- if-else, for while, do-while; Case statement; Arrays; Formatted & unformatted I/O; Type modifiers & storage classes; Ternary operator; Type conversion & type casting; Special constructs-Break, continue, exit (), goto & labels;

Functions; Arguments; Return Value; Parameter passing- call by value, call by reference; Return statement; Scope, visibility and life-time rules for various types of variable, static variable; Calling a function; Recursion – basics, comparison with iteration, tail recursion, when to avoid recursion, examples.

Overview of object oriented programming, features, comparison with procedural languages, applications, advantages. C++ basics, data types, Operators, loops and decisions, structures and functions, references.

Class and Objects: Class, Objects, Constructors, Destructors, Access Modifiers-Public, Private and Protected. Accessibility and Scope, Encapsulation, Inheritance, Polymorphism. Dynamic Binding, Exceptions

List of Experiments

1. Write a Program to Generate a Random Number
2. Write a Program to Convert Kilometers to Miles
3. Write a Program to Check if a Number is Positive, Negative or 0
4. Write a Program to Print the Fibonacci sequence
5. Write a Program to Find ASCII Value of Character
6. Write a Program to Display Calendar
7. Write a Program to Find the Sum of Natural Numbers Using Recursion
8. Write a Program to Find the Factorial of Number Using Recursion
9. Write a Program to Convert Decimal to Binary Using Recursion
10. Write a Program to Add Two Matrices, Transpose a Matrix, Multiply Two Matrices
11. Write a Program to Check Whether a String is Palindrome or Not
12. Write a Program to Remove Punctuations from a String
13. Write a Program to Sort Words in Alphabetic Order
14. Write a program to find addition of five number using Array.
15. Write a program to find Number is prime or not prime.
16. Write a program to find CGPA of three subject.
17. Write a program to print a table of given number.
18. Write a program to print star pattern.

19. Write a program to find square of two number.
20. Write a program to print days using Switch Statement.
21. Write a Program to Catch Multiple Exceptions
22. Write a program to find Area of circle and Circumference of Circle using class
23. Write a program to create a person class. Include attributes like name, country, and date of birth. Implement a method to determine the person's age
24. Write a program to create a class representing a bank. Include methods for managing customer accounts and transactions.
25. Write a program to create a class representing a shopping cart. Include methods for adding and removing items and calculating the total price.

Course outcomes:

Student would be able to

CO1: **Implement** inheritance, polymorphism, encapsulation, abstraction.
CO2: **Modify** existing codes and classes as per the requirement of software development.
CO3: **Develop** object-oriented application system as part of a team in industry.
CO4: **Design** the classes and constructs for real time software as per societal needs.

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1
CO1	3						2	3			2
CO2			2	2	3	3				3	
CO3		3		3	3	3	3			3	
CO4	3		3		2		3	2	3	3	3

Lab-I (Programming Lab in Unix/Linux/Ubunto)
68251108

List of Experiments

S. No.	List of programs
1	Write a shell script for arithmetic operations (+, -, *, /) using switch case.
2	Write a shell script to find whether the number is positive, negative or zero.
3	Write a shell script for swapping two numbers without using third variable.
4	Write a shell script for palindrome.
5	Write a shell script for factorial.
6	Write a shell script for Fibonacci series.
7	Write a shell script to print given number in reverse order. (123 = 321)
8	Write a shell script to print a sum of all digits in a given number. (123 = 6)
9	Write a shell script to print number of digits in a given number. (123 = 3)
10	Write a script to print the given series * * * * * * * * * * * * * * *
11	Write a script to print the given series 1 2 3

Course outcomes:

Student would be able to

CO1: Utilize Shell commands to develop Shell Script
CO2: Apply shell programming concepts to solve problems using control structures, loops, and functions.
CO3: Design a text file with the vi text editor using the standard vi editor commands.
CO4: Evaluate the effectiveness of different shell programming techniques and tools in solving real-world problems.

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1
CO1	2		3			2		2			
CO2	2		3			2		2			
CO3			3					2			2
CO4	2		3	3	3	2	3	2	3	3	3