BACHELORS WITH COMPUTER APPLICATIONS AS MAJOR
4th SEMESTER

CAP422J1 COMPUTER APPLICATIONS _ DBMS

CREDITS: THEORY (3) PRACTICAL (1)

Course Learning Outcomes:

- *To introduce the core concept of Relational Database.*
- *To enable students to design the databases for a wide variety of Real-World problems*
- *To introduce the concept and process of Database Normalization*
- *To enable the student to learn DML, DDL, DCL commands using SQL*

Unit I:

Introduction to Databases, Database Users, Characteristics of Database approach, Applications of DBMS, Advantages and Disadvantages, Database System Concepts and Architecture, Data Models, Schemas, Instances, Three-Schema Architecture and Data Independence, Database System, Centralized and Client/Server Architecture for DBMS,

Unit II:

Data Modelling using Entity-Relationship Mode, Entity Types, Entity Sets, Attributes, and Keys, Relationship Types, Relationship Sets, ER Diagrams, Relational Data Model and Relational Model Constraints, Functional Dependencies, Normalization: 1NF, 2NF, 3NF

Unit III:

Relational Algebra and Relational Calculus, Introduction to DDL, DML, DCL. Introduction to transactions, Transaction states ACID properties, Concurrency Control Techniques: 2-phase locking, time stamp ordering.

TEXTBOOK:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010

REFERENCES:

1. An introduction to database systems by Desai, Bipin C
2. PL/SQL by Ivan Bayross

DBMS- LAB

Creation of Tables in SQL (DDL)

1. Overview of using SQL tool, Data types in SQL,
2. Create Schema and insert at least 5 records for each table.
3. Queries for Altering Tables (with add and modify clause) , Dropping Tables
4. Queries for DROP RENAME TRUNCATE and Describe a Table
5. SQL Constraints: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT

Practicing DML Commands

6. Insert, Select, Update, Delete
7. Select Command with WHERE, IN, BETWEEN AND LIKE clause
8. SELECT with AND, OR, NOT
9. Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING.

Joining Tables

10. Practice the Inner Join, Left Join,
11. Practice the Right Join, Cross Join

Create the following Database Schema and Practice the queries

EMP (eno, ename, bdate, title, salary, dno)
PROJECT (pno, pname, budget, dno)
DEPT (dno, dname, mgreno)
WORKSON (eno, pno, resp, hours)

12. Write an SQL query that returns the project number and name for projects with a budget greater than 100,000.
13. Write an SQL query that returns all works on records where hours worked is less than 10 and the responsibility is 'Manager
14. Write an SQL query that returns the employees (number and name only) who have a title of 'EE' or 'SA' and make more than 35,000.
15. Write an SQL query that returns the employees (name only) in department 'D1' ordered by decreasing salary.
16. Write an SQL query that returns the departments (all fields) ordered by ascending department name.
17. Write an SQL query that returns the employee's name, department name, and employee title.
18. Write an SQL query that returns the project name, hours worked, and project number for all works on records where hours > 10.
19. Write an SQL query that returns the project name, department name, and budget for all projects with a budget < 50,000.
20. Write an SQL query that returns the employee numbers and salaries of all employees in the 'Consulting' department ordered by descending salary.
21. Write an SQL query that returns the employee's name, project name, employee title, and hours for all works on records.

BACHELORS WITH COMPUTER APPLICATIONS AS MAJOR

4th SEMESTER

CAP422J2 COMPUTER APPLICATIONS _ OOPs with C++

CREDITS: THEORY (4) PRACTICAL (2)

COURSE LEARNING OUTCOME:

The student should understand object-oriented programming and C++ concepts, improve his/her problem-solving skills and should:

- *Be able to explain the difference between object-oriented programming and procedural programming.*
- *Be able to program using C++ features such as composition of objects, operator overloading, inheritance and polymorphism, file I/O, etc.*
- *Be able to build C++ classes using appropriate encapsulation and design principles.*
- *Be able to apply object-oriented techniques to solve bigger computing problems.*

UNIT-1 (15 HOURS)

Introduction to Object Oriented Programming

Comparison of Procedural Programming and OOP, Benefits of OOP, Abstraction, Encapsulation, Inheritance, Polymorphism, Difference between C and C++.

Elements of C++ Language:

Tokens and identifiers, Variables and Constants, Reference variables, Basic data types in C++, Streams in C++. Types of operators in C++.

Decision and Control Structures:

if statement, if-else statement, switch statement, Loop: while, do-while, for. Break and continue.

Pointers and structures.

UNIT-2 (15 HOURS)

Functions:

Inline function, function overloading

Introduction to Classes and Objects:

Classes in C++, class declaration, declaring objects, Defining Member functions, Inline member function, Array of objects, Objects as function argument, Static data member and member function, Friend function and friend class.

Constructors and Destructors:

Constructors, Instantiation of objects, Default constructor, Parameterized constructor, Constructor overloading, Copy constructor and its use, Destructors.

UNIT-3 (15 HOURS)

Operator Overloading:

Overloading unary and binary operators.

Inheritance:

Derived class and base class, access specifiers, type conversion, accessing the base class member, Types of Inheritance, Virtual base class, Abstract class

UNIT-4 (15 HOURS)

Virtual Functions and Polymorphism:

Virtual functions, pure virtual functions; Polymorphism. Compile time polymorphism, Run time polymorphism

File Handling:

Opening and Closing a file, File modes, Functions for I/O operations.

Introduction to standard template library.

Components of STL, Containers: Vector and lists.

Text Book:

i. E. Balagurusamy: *Object oriented programming with C++*

REFERENCE:

i. Bjarne Stroustrup: *The C++ programming language.*

ii. Robert Lafore: *Object oriented programming in C++*

iii. Yashwant Kanathker: *Let's C++*

iv. Schildt. H: *C++ The Complete Reference*

PRACTICALS (2 CREDITS: 30 HOURS)

UNIT -1

1. Write a C++ program using decision making and loop structure for the given situation.
2. Use the structure in C++ program for solving the given problem.
3. Create C++ programs to perform the given arithmetic operations using pointers.
4. Write a program to Demonstrate the use of Streams in C++
5. Write a menu driven program in C++ using switch statement.

UNIT -2

1. Develop programs that implements a class and use it with objects.
2. Develop relevant friend functions to solve the given problem.
3. Use function overloading to solve the given problem.
4. Write a program to implement all types of constructors (constructor overloading).
5. Write program to delete the given object using destructor in C++ program.

UNIT -3

1. Write a program to demonstrate operator overloading for Unary operator.
2. Write a program to demonstrate operator overloading for Binary operator.'
3. Write a program for implementing single, multi-level, and multiple inheritance.
4. Write a C++ program using virtual base class.
5. Develop a program to demonstrate use of Pointer to derived class

UNIT -4

1. Write a C++ program demonstrating the use of pure virtual function.
2. Implement run time polymorphism using virtual functions in the given C++ program.
3. Develop C++ program to perform read/write operation from/to the given file.
4. Write a C++ program to illustrate the iterators in vector
5. Develop C++ program to demonstrate the use of list containers.

BACHELORS WITH COMPUTER APPLICATIONS AS MAJOR
4th SEMESTER
CAP422J3 COMPUTER APPLICATIONS _ COMPUTING MATHEMATICS
CREDITS: THEORY (4) PRACTICAL (2)

COURSE LEARNING OUTCOMES:

- *To introduce elements of 10+2 level mathematics to students of Computer Applications who are from a medical or arts background*
- *To cover fundamental concepts of matrices and determinants*
- *To cover fundamental concepts of calculus.*
- *To acquire fundamental knowledge regarding the problems of approximation and errors in Computer based numerical problems solving.*

UNIT 1: MATRICES AND DETERMINANTS:

Matrix, Types of Matrices, Matrix Operations: Add, Sub, Multiply, Divide, Transpose. Determinants, Minors and Cofactors, Properties of Determinants. Elementary Transformations. Solutions of Linear Equations by Matrix Method and Determinants.

UNIT 2: FUNCTIONS AND LIMITS

Functions, Concept, Domain and Range, Types of Functions
Limits and Continuity: Limit of a Function, Algebra of Limits, Simplification and Evaluation of Limits. Standard Limits, Continuity, Geometric Meaning

UNIT 3: DIFFERENTIATION

Ab-Initio Method, Derivatives of some Common Functions, Rules to find Derivatives, Second Order Derivatives, Anti-Derivative of a Function.

UNIT 4: APPROXIMATION AND ERRORS:

Mathematical Modelling and Engineering Problem-Solving, Role of Computers and Software. Significant Figures, Accuracy and Precision. Numerical Error. Floating Point Representation. Computer Arithmetic. Round-Off, Truncation Errors and Error Propagation.

TEXTBOOK:

1. Wiley's Problems in Mathematics for JEE Main and Advanced (Vol 1 + 2)
2. Introduction to Linear algebra by Gilbert Strang, 6th Edition

REFERENCES:

1. Numerical Methods for engineers. S C Chapra and R P Canale , McGrow Hill International Edition.
2. Numerical Methods for Scientific & Engineering Computation, M. K. Jain, S.R.K.
3. Numerical Methods in Science & Engineering Prog. - By Dr. B. S. Grawal, Khanna Pub., New Delhi.
4. R S Aggarwal, Senior Secondary School Mathematics for Class XI/XII (Bharati Bhawan)
5. R.D Sharma – Mathematics for Class 11th / 12th (4 Vols.)
6. Computer Oriented Numerical Methods, R. S. Salaria., Khanna Publisher.

TUTORIALS:

1. Show how to use determinants to calculate the area of a triangle from its coordinates.
2. Show how to use determinants to determine that a set of given points are collinear.
3. Show how to use determinants to solve a non-homogenous system of linear equations
4. Show how to use determinants to solve a homogenous system of linear equations
5. Show that the value of a determinant remains unchanged if its rows and columns are interchanged
6. Show that if each element of a row or a column of a determinant is multiplied by a constant k then the value of the new determinant is k times the value of the original determinant
7. Show the different ways of representing a mathematical function. Also, given a mathematical function, show how to plot its graph.
8. Use limits to determine the tangent to the curve at a given point
9. Use limits to determine the instantaneous velocity of a ball dropped from a height at a given time.
10. Use differentiation to compute the rate of change of one quantity from the rate of change of another, more easily measurable quantity (e.g. the rate of increase of the radius of a balloon from the rate of change of its volume)
11. Show how to use the derivative of a function as a measure of rate of change
12. Show how to use the derivative to calculate the maxima and minima of a function
13. Write an algorithm for finding the roots of a quadratic equation
14. Develop a simple mathematical model to determine the terminal velocity of a free-falling body (a parachutist) near the earth's surface. Present a numerical solution for this problem.

REFERENCES:

- Stewart, Calculus Concepts and Contexts (2nd Edition)
- R.S. Aggarwal, Mathematics XI/XII
- Numerical Methods for engineers. S C Chapra and R P Canale

BACHELOR WITH APPLIED COMPUTING AS MINOR

(FOR STUDENTS WITH MAJOR IN COMPUTER APPLICATIONS / INFORMATION TECHNOLOGY)

4th SEMESTER

ACP422N APPLIED COMPUTING _ IOT FUNDAMENTALS

CREDITS: THEORY: 3; PRACTICAL: 1

COURSE LEARNING OUTCOMES:

- *Understand the fundamental characteristics of IOT, including its physical design, basic components, and the concepts of things, sensing, and actuators.*
- *Explore various application areas of IoT such as home automation, smart cities, medical, logistics, environment, analytics, and smart grids.*
- *Gain insights into IoT protocols used for communication and data exchange within IoT ecosystems.*
- *Develop hands-on skills in working with hardware platforms like Raspberry Pi and Arduino, and learn how to implement basic sensors for monitoring temperature, humidity, proximity, gas, air quality, and ultrasonic sensors.*

THEORY (3 CREDITS)

UNIT - 1 (15 Hours)

Introduction: Definition & Characteristics of IOT, Physical Design of IOT, Basic Components of IoT, Thing, Sensing & Actuators, Vision, Physical Parameters. IOT Protocols.
Application Areas of IoT: Home Automation, Smart Cities, Medical, Logistics, Environment, Analytics. Smart Grids.

UNIT - 2 (15 Hours)

IOT Communication Models, APIs, IoT Architecture: Basic Architecture: 3 layer and 5-layer Architecture, ITU-IoT Reference Model, Machine to Machine Communication, IoT Gateways, Wireless Sensor Networks. Technologies: Bluetooth Low Energy (BLE), ZigBee: Architecture, Comparison with other wireless standards. LoraWAN.

UNIT - 3 (15 Hours)

Electronic Product Code (EPC), Near Field Comm. (NFC), 6LoWPAN, End to End Reliability: MQTT, SCADA. Hardware and Software Platforms: Hardware: Raspberry Pi, ESP8266 Wifi Module, Arduino. Implementation of Basic Sensors (temperature, humidity, proximity, gas, air quality, Ultrasonic sensors)
Internet of Things Privacy and Security Issues, Steps towards a security platform in IoT

TEXT BOOKS:

1. Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)”, 1stEdition, VPT, 2014. (ISBN-13: 978-8173719547)
2. Internet of Things (IoT), Dr. Kamlesh Lakhwani, Dr. Hemant Kumar Gianey, Joseph Kofi Wireko, Kamal Kant Hiran

REFERENCES:

1. Schwartz, Marco. “Internet of Things with Arduino Cookbook”. Packt Publishing Ltd, 2016.
2. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, “From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence”, 1st Edition, Academic Press, 2014. (ISBN-13: 978-0124076846)
3. Hakima Chaouchi “The Internet of Things Connecting Objects to the Web” by Wiley publications

PRACTICAL (1 CREDIT: 30 HOURS)

LEARNING OUTCOME*: These practical's cover a range of fundamental concepts in IoT, including hardware interfacing, sensor integration, data communication, cloud integration, and real-world applications. Students can gain hands-on experience and a solid understanding of the core principles of IoT through these practical exercises.*

1. Study the fundamental of IOT software's and components. Install Arduino IDE development platform.
2. Connect a microcontroller chip on a breadboard, establish power connections, and verify basic functionality. Gather the necessary components: microcontroller chip (e.g., Arduino Uno), breadboard, jumper wires, USB cable for power, and an LED.
3. To practice connecting analog input and output components to the microcontroller using the breadboard.
4. Write a program to Read sensor data using analog or digital pins.
5. LED Blinking Using Arduino: To interface LED/Buzzer with Arduino/Raspberry Pi/ ESP8266 and write a program to turn ON & OFF.
6. To interface Push button/Digital sensor (IR/LDR) with Arduino/ ESP8266/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection. Upload the program to the board and observe the blinking.
7. To interface DHT11 sensor with Arduino/Raspberry Pi/ ESP8266 and write a program to print temperature and humidity readings and Display sensor readings on the serial monitor.
8. IoT Cloud Platform Integration: Create an account on a popular IoT cloud platform ThingSpeak.
9. Write a program to Send temperature and humidity sensor data to the cloud and visualize it on a dashboard.
10. Write a Program to interface motor using relay with Arduino/Raspberry Pi/ ESP8266 and write a program to turn ON motor when push button is pressed.
11. Build a web server using Arduino and Ethernet/Wi-Fi shield using ESP8266 Wifi module. Develop a web page to control LEDs remotely.
12. Use buttons on the webpage to toggle LED states.
13. Log data at regular intervals to an SD card or cloud platform. Create graphs to visualize the logged data over time.
14. Set up an MQTT broker (e.g., Mosquitto) on a computer or cloud server. Program an Arduino to publish sensor data and subscribe to commands via MQTT.
15. Create a soil moisture sensing system using Arduino and moisture sensors. Send data to the cloud to monitor soil conditions remotely.
16. Design a basic home automation system with Arduino and relays.

BACHELORS WITH COMPUTER APPLICATIONS AS MINOR
4th SEMESTER

CAP422N COMPUTER APPLICATIONS _ DBMS

CREDITS: THEORY (3) PRACTICAL (1)

Course Learning Outcomes:

- *To introduce the core concept of Relational Database.*
- *To enable students to design the databases for a wide variety of Real-World problems*
- *To introduce the concept and process of Database Normalization*
- *To enable the student to learn DML, DDL, DCL commands using SQL*

Unit I:

Introduction to Databases, Database Users, Characteristics of Database approach, Applications of DBMS, Advantages and Disadvantages, Database System Concepts and Architecture, Data Models, Schemas, Instances, Three-Schema Architecture and Data Independence, Database System, Centralized and Client/Server Architecture for DBMS,

Unit II:

Data Modelling using Entity-Relationship Mode, Entity Types, Entity Sets, Attributes, and Keys, Relationship Types, Relationship Sets, ER Diagrams, Relational Data Model and Relational Model Constraints, Functional Dependencies, Normalization: 1NF, 2NF, 3NF

Unit III:

Relational Algebra and Relational Calculus, Introduction to DDL, DML, DCL. Introduction to transactions, Transaction states ACID properties, Concurrency Control Techniques: 2-phase locking, time stamp ordering.

TEXTBOOK:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010

REFERENCES:

1. An introduction to database systems by Desai, Bipin C
2. PL/SQL by Ivan Bayross

DBMS- LAB

Creation of Tables in SQL (DDL)

1. Overview of using SQL tool, Data types in SQL,
2. Create Schema and insert at least 5 records for each table.
3. Queries for Altering Tables (with add and modify clause) , Dropping Tables
4. Queries for DROP RENAME TRUNCATE and Describe a Table
5. SQL Constraints: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT

Practicing DML Commands

6. Insert, Select, Update, Delete
7. Select Command with WHERE, IN, BETWEEN AND LIKE clause
8. SELECT with AND, OR, NOT
9. Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING.

Joining Tables

10. Practice the Inner Join, Left Join,
11. Practice the Right Join, Cross Join

Create the following Database Schema and Practice the queries

EMP (eno, ename, bdate, title, salary, dno)
PROJECT (pno, pname, budget, dno)
DEPT (dno, dname, mgreno)
WORKSON (eno, pno, resp, hours)

12. Write an SQL query that returns the project number and name for projects with a budget greater than 100,000.
13. Write an SQL query that returns all works on records where hours worked is less than 10 and the responsibility is 'Manager
14. Write an SQL query that returns the employees (number and name only) who have a title of 'EE' or 'SA' and make more than 35,000.
15. Write an SQL query that returns the employees (name only) in department 'D1' ordered by decreasing salary.
16. Write an SQL query that returns the departments (all fields) ordered by ascending department name.
17. Write an SQL query that returns the employee's name, department name, and employee title.
18. Write an SQL query that returns the project name, hours worked, and project number for all works on records where hours > 10.
19. Write an SQL query that returns the project name, department name, and budget for all projects with a budget < 50,000.
20. Write an SQL query that returns the employee numbers and salaries of all employees in the 'Consulting' department ordered by descending salary.
21. Write an SQL query that returns the employee's name, project name, employee title, and hours for all works on records.