Course Code: PCAA101C	Programming and Problem Solving in C	Credits : 04
Hours/Week (L:T:P) : 2:2:0		CIE Marks : 50
Total Hours of Pedagogy (Theory + Lab): 40 hours Theory + 10 hours Lab		SEE Marks : 50
Course Type: Integrated(IPCC)		

Course Objectives:

1. Implement the constructs of C Language.
2. Construct C Programs using basic programming constructs
3. Develop C programs using arrays and strings
4. Organize modular applications in C using functions
   Integrate pointers and structures in C applications and Execute input/output and file handling in C

Module-1	8 Hrs.

BASICS OF C PROGRAMMING: Problem solving using Algorithm and flowchart, Structure of C program - C programming: Data Types - Constants – Enumeration Constants - Keywords – Operators: Precedence and Associatively - Expressions - Input/ Output statements, Assignment statements Decision making statements - Switch statement.

Module-2	8Hrs.

Jumping and Looping statements – Preprocessor directives - Compilation process. **ARRAYS** Introduction to Arrays: Declaration, Initialization – One dimensional array – Two dimensional arrays .

Module-3	8 Hrs.

STRINGS: String operations: length, compare, concatenate, copy –**FUNCTIONS:** Modular programming - Function prototype, function definition, function call, Built-in functions (string functions, math functions) Recursion, Binary Search using recursive functions –

Module-4	8 Hrs.

POINTERS: Pointer operators Pointer arithmetic Arrays and pointers – Array of pointers – Parameter passing: Pass by value, Pass by reference. **STRUCTURES:** Structure - Nested structures – Pointer and Structures – Array of structures Self referential structures typedef

Module-5	8 Hrs.

Union - Storage classes and Visibility. **FILE PROCESSING** Files Types of file processing: Sequential access, Random access Sequential access file - Random access file - Command line arguments.

Practical Module

Sl. NO	Experiments
1	Simulation of a Simple Calculator.
2	Implement Binary Search on Integers
3	Sort the given set of N numbers using Bubble sort.
4	Implement Matrix multiplication and validate the rules of multiplication.
5	An electricity board charges the following rates for the use of electricity: for the first 200 units 80 paisa per unit for the next 100 units 90 paisa per unit: beyond

	300 units Rs 1 per unit. All users are charged a minimum of R 100 as meter charge. If the total amount is more than Rs 400, then an additional surcharge of 15% of total Amount is charged. Write a program to read the name of the user, number of units consumed and print out the charges.
6	Write functions to implement string operations such as compare, concatenate, and find string length. Use the parameter passing techniques.
7	Implement structures to read, write and compute average- marks of the students, list the students scoring above and below the average marks for a class of N students.
8	Write a C program to copy a text file to another, read both the input file name and target file name.

Suggested Learning resources

TEXT BOOKS:

1. Reema Thareja, "Programming in C", Oxford University Press, Second Edition, 2016
2. Kernighan, B.W and Ritchie,D.M, "The C Programming language", Second Edition, Pearson Education, 2015.

REFERENCES:

1. Paul Deitel and Harvey Deitel, "C How to Program with an Introduction to C++", Eighth edition, Pearson Education, 2018.
2. Yashwant Kanetkar, Let us C, 17th Edition, BPB Publications, 2020.
3. Byron S. Gottfried, "Schaum's Outline of Theory and Problems of Programming with C", McGraw-Hill Education, 1996.
4. Pradip Dey, Manas Ghosh, "Computer Fundamentals and Programming in C", Second 5. Edition, Oxford University Press, 2013.
5. Anita Goel and Ajay Mittal, "Computer Fundamentals and Programming in C", 1st Edition, Pearson Education, 2013.

Course Outcomes:

CO1: Demonstrate knowledge on C Programming constructs
CO2: Develop simple applications in C using basic constructs
CO3: Design and implement applications using arrays and strings
CO4: Develop and implement modular applications in C using functions
CO5: Develop applications in C using structures and pointers

Course Code: PCAA102C	Discrete Mathematics and Graph Theory	Credits : 03
Hours/Week (L:T:P) : 2:0:1		CIE Marks : 50
Total Hours of Pedagogy (Theory + Lab): 40		SEE Marks : 50
Course Type: Theory		

Course Objectives:

1. Analyze basic concepts of mathematical logic for analyzing propositions and proving theorems
2. Apply sets and their operations algebraically to solve real-world problems.
3. Examine the basics of graph theory and their various properties.
4. Model problems using graphs and to solve these problems algorithmically.
5. Apply graph theory concepts to solve real world problems in Computer Networking, Telecommunication Infrastructure, Transportation etc.

Module-1	8 Hrs.

Basic Structures: Sets, Principle of Inclusion, Exclusion and Pigeonhole principle Functions and Matrices: Eigen values and Eigenvectors.

Module-2	8 Hrs.

Mathematical Logic, Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference Introduction to Proofs

Module-3	8 Hrs.

Introduction to Graphs: Application of graphs – finite, infinite and bipartite graphs – Incidence and Degree – Isolated vertex, pendant vertex and Null graph. Paths and circuits – Isomorphism, sub-graphs, walks, paths and circuits, connected graphs, disconnected graphs and components.

Module-4	8 Hrs.

Eulerian and Hamiltonian graphs: Euler graphs, Operations on graphs, Hamiltonian paths and circuits, Travelling salesman problem. Directed graphs – types of digraphs, Digraphs and binary relation.

Module-5	8 Hrs.

Graph Colouring: Colouring- Chromatic number, Chromatic polynomial, Matchings, Coverings, Four colour problems and Five colour problem. Greedy colouring algorithm.

Suggested Learning resources

Text Books :

1. Kenneth H Rosen, "Discrete Mathematics and its Applications", McGraw Hill publications 7th edition.
2. Narsingh Deo, Graph theory with the applications to engineering & Computer Science, Dovers Publications, 2016
3. J.A. Bondy and U.S.R. Murty. Graph theory with Applications, Springer, 1 st edition, 2008.

References Books

1. J.K Sharma "Discrete Mathematics", Mac Millian Publishers India, 3rd edition,2011
2. Garry Chartand and Ping Zhang, Introduction to Graph Theory, Tata McGraw-Hill, 2006.

Frank Harary, Graph Theory, Narosa Publishing House, Latest edition.

Course Outcomes:

CO1: Apply the fundamentals of set theory and functions to perform various set operations to the real world problems.

CO2: Understand basic concepts of mathematical logic for analyzing propositions and proving theorems and its operations. Algebraically for solving real world problems.

CO3: Understand the basics of graph theory and their various properties

CO4: Model problems using graphs and to solve these problems algorithmically

CO5: Apply graph theory concepts to solve real world problems in Computer Networking, Telecommunication Infrastructure, Transportation etc

Course Code: PCAA103C	Database Management Systems (DBMS)	Credits : 03
Hours/Week (L:T:P) : 3:0:0		CIE Marks : 50
Total Hours of Pedagogy (Theory + Lab): 40		SEE Marks : 50
Course Type: Theory		

Course Objectives:

1. Analyze the basic concepts and the applications of database systems.
2. Evaluate the different issues involved in the design and implementation of Database System.
3. Explain the basic concepts of relational data model, entity relationship model, relational database design, relational algebra and database language SQL.
4. Design and build simple applications using database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS

Module-1	8 Hrs.

Databases and Database Users: Introduction, Characteristics of the Database Approach, Actors on the scene, Workers behind the scene, Advantages of using the DBMS approach, When not to use a DBMS, **Database System Concepts and Architecture**: Data models, schemas and instances, Three-schema architecture and data independence, Database language and interfaces, The database system environment.

Data Modelling Using the Entity-Relationship(ER) Model: Using High-Level Conceptual Data Models for Database Design; An Example Database Application; Entity Types, Entity Sets, Attributes and Keys; Relationships, Relationship Types, Roles and Structural Constraints; Weak Entity Types; Refining the ER Design for the COMPANY Database; ER Diagrams, Naming Conventions and Design Issues.

Module-2	8 Hrs.

The Relational Data Model and Relational Database Constraint: Relational Model Concepts, Relational Constraints and Relational Database Schemas, Update Operations, Transactions, and Dealing with Constraint Violations. Relational Algebra and Relational Calculus: Unary Relational Operations, Relational Algebra Operations from Set Theory, Binary Relational Operations, Additional Relational Operations, Overview of Tuple Relational Calculus and Domain Relational Calculus; Examples of Queries in Relational Algebra. **Relational Database Design Using ER and EER to-Relational Mapping:** Relational Database Design Using ER to Relational Mapping.

SQL-99: Schema Definition, Constraints, Queries and Views: SQL Data Definition and Data types, Specifying Constraints in SQL, Schema Change statement in SQL, Basic Queries in SQL, Insert, Delete and Update Statements in SQL, Specifying Constraints as Assertion and Triggers, Views in SQL.

Module-3	8 Hrs.

Functional Dependencies and Normalization for Relational Database: Informal Design Guidelines for Relation Schemas, Functional Dependencies, Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form. **Relational Database Design Algorithms and Further Dependencies:** Multi-valued Dependencies and fourth normal form, Join Dependencies and fifth normal form.

Module-4	8Hrs.

Overview of Transaction Management: The ACID Properties: Consistency and Isolation, Atomicity and Durability; Transactions and Schedules; Concurrent Execution of Transactions: Motivation for Concurrent Execution, Serializability, Anomalies due to Interleaved Execution, Schedules Involving Aborted Transactions; Lock- Based Concurrency Control: Strict Two-Phase Locking, Deadlocks;

Performance of Locking; Timestamp Based Protocols- Validation- Based Protocols, Multiple Granularity. Transaction Support in SQL: Creating and Terminating Transactions, What Should We Lock? Transaction Characteristics in SQL.

Module-5	8 Hrs.

Introduction to Crash Recovery: Stealing Frames and Forcing Pages, Recovery - Related Steps during Normal Execution, Overview of ARIES recovery algorithm, Atomicity: Implementing Rollback. Check Points Buffer Management, Failure with loss of nonvolatile storage.

Database Security: Introduction to Database Security; Access Control; Discretionary Access Control: Grant and Revoke on Views and Integrity Constraints; Mandatory Access Control: Multilevel Relations and Poly instantiation, Covert Channels, DoD Security Levels.

Suggested Learning resources

Text Books:

1. Elmasri and Navathe, Fundamentals of Database Systems, 6th Edition, Pearson Education, 2011.
2. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, 3rd Edition, , TATA McGrawHill.
3. Silberschatz and Korth, Database System Concepts, 7th edition, Mc Graw hill.

Reference Books:

1. C.J. Date, A.Kannan, S.Swami Nadhan, An Introduction to Database systems, 8th Edition, Pearson.
2. M. L. Gillenson, Fundamentals of Database Management Systems, Wiley Student Edition.
3. S.Shah and V. Shah, Oracle for Professionals, The X Team, SPD.

Course Outcomes:

CO1: Demonstrate the basic elements of a relational database management system
CO2: Ability to identify and build the data models for relevant problems**.**
CO3: Design entity relationship and convert entity relationship diagrams into relations and formulate SQL queries to process the data.
CO4: Ability to analyze the relational model on rules of normal forms.
CO5: Build transaction with lock and unlock utility.

Course Code: PCAA104C	Operating System	Credits : 03
Hours/Week (L:T:P) : 2:0:1		CIE Marks : 50
Total Hours of Pedagogy (Theory + Lab): 40		SEE Marks : 50
Course Type: Theory		

Course Objectives:

1. Explain the need and services of the operating system
2. Explore how the operating system handles processes and manages memory.
3. Recognize deadlock condition and technique to handle deadlock situation.
4. Analyze various memory management strategies and file handling concepts.

Module-1	8 Hrs.

Introduction to Operating Systems, System Structure What operating systems do, Operating System Operations, Computing Environments, Operating System Services, System Calls, Types of System Calls, System Programs, Operating System Structure, System Boot.

Process Concept Process Concept, Process Scheduling, Interprocess Communication

Module-2	8 Hrs.

Process Scheduling Basic Concepts, Scheduling Criteria, Scheduling Algorithms. Synchronization Background, The Critical Section Problem, Mutex Locks, Semaphores, Classic Problems of Synchronization: Readers-Writers Problem, Dining Philosophers Problem using Semaphores

Module-3	8 Hrs.

Deadlocks: System model, Deadlock Characterization, Methods for handling deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock

Module-4	8 Hrs.

Memory Management Strategies Basic Hardware, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Virtual Memory Management Background, Demand Paging, Page Replacement

Module-5	8 Hrs.

File System File concept, Access methods, Directory overview Implementing File System Allocation methods, Free Space Management

Suggested Learning resources

Text Books

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating Systems Principles, 10th Edition, Wiley – India, 2019.

Reference Books:

1. D M Dhamdhere: Operating Systems A Concept Based Approach, 3rd Edition, Tata McGraw Hill, 2017.
2. Harvey M Deital: Operating Systems, 3rdEdition, Addison Wesley, 1990.

Course Outcomes:

CO1: Describe the elements and various functionalities of the operating system
CO2: Apply the techniques of process management and demonstrate process synchronization.
CO3: Recognize deadlock condition and technique to handle deadlock situation.
CO4: Analyze various memory management strategies.
CO5: Describe file handling concepts.

Course Code: PCAA105C	Web Technologies	Credits : 03
Hours/Week (L:T:P) : 3:0:0		CIE Marks : 50
Total Hours of Pedagogy (Theory + Lab): 40		SEE Marks : 50
Course Type: Theory		

Course Objectives:

1. Understand Internet, Web technology and Web Applications.
2. Creating the small web page using HTML5 and CSS.
3. Developing the interactive web pages using JavaScript
4. Create web pages using angular JS framework

Module-1	8 Hrs.

Web browsers, web servers, MIME, URL, HTTP Introduction to HTML5 tags, Basic syntax and structure, text markups, images, lists, tables, progress, Media tags-audio and video, forms, span and div tags.

Module-2	8 Hrs.

Introduction to CSS, Levels of CSS, Selectors, Font, color and Text Properties, BOX Model, Introduction to JavaScript, JavaScript variables, operators, Conditional and loop statements in JavaScript, Functions and Arrays in JavaScript

Module-3	8 Hrs.

Event Handling and Document Object model in JavaScript, Handling strings and working with window object

Module-4	8 Hrs.

Introduction to AngularJS, Expressions, Modules, Directives, Model, Data binding, Controllers,Scopes, Filters

Module-5	8 Hrs.

Services, Tables, Select box, Forms, Events, Validations

Suggested Learning resources

Books

1. Web Programming By Chris Bates , Wiley Publications
2. HTML5 Black Book by Dreamtech
3. Angular JS By Krishna Rungta

Course Outcomes:

CO1: Explain the fundamental concepts of web technologies
CO2: Create the web pages using HTML and CSS
CO3: Implement user interactive web pages
CO4: Demonstrate the single window applications using AngularJS
CO5: Apply advance web technology concepts to build web pages.

Course Code: PCAA106L	DBMS and Web Technologies Laboratory	Credits : 02
Hours/Week (L:T:P) : 0:2:0		CIE Marks : 50
Total Hours of Pedagogy (Theory + Lab):		SEE Marks : 50

Course Type: Practical

Course Objectives:

1. Create database objects that include tables, constraints, Views and indexes.
2. Create SQL queries to read information form tables.
3. Design Web page.
4. Building interactive web pages.

Sl. No.	Experiments
01	Create the following tables with properly specifying Primary keys, Foreign keys and solve the following queries. BRANCH (Branchid, Branchname, HOD) STUDENT (USN, Name, Address, Branchid, sem) BOOK (Bookid, Bookname, Authorid, Publisher, Branchid) AUTHOR (Authorid, Authorname, Country, age) BORROW (USN, Bookid, Borrowed_Date) Execute the following Queries: i. List the details of Students who are all studying in 2nd sem MCA. ii. List the students who are not borrowed any books. iii. Display the USN, Student name, Branch_name, Book_name, Author_name, Books_Borrowed_Date of 2nd sem MCA Students who borrowed books. iv. Display the number of books written by each Author. v. Display the student details who borrowed more than two books. vi. Display the student details who borrowed books of more than one Author.vii.Display the Book names in descending order of their names. viii.List the details of students who borrowed the books which are all published by the same publisher.
02	Consider the following schema: STUDENT (USN, name, date_of_birth, branch, mark1, mark2, mark3, total, GPA) Execute the following queries: i. Update the column total by adding the columns mark1, mark2, mark3. ii. Find the GPA score of all the students. iii. Find the students who born on a particular year of birth from the date_of_birth column. iv. List the students who are studying in a particular branch of study. v. Find the maximum GPA score of the student branch-wise. vi. Find the students whose name starts with the alphabet "S" vii. Find the students whose name ends with the alphabets "AK" viii. Delete the student details whose USN is given as 1001
03	Design an ER-diagram for the following scenario, Convert the same into a relational model and then solve the Following queries. Consider a Cricket Tournament "ABC CUP" organized by an organization. In the tournament there are many teams are contesting each having a Teamid,Team_Name, City, a coach. Each team is uniquely identified by using Teamid. A team can have many Players and a captain.

	Each player is uniquely identified by Playerid, having a Name, and multiple phone numbers,age. A player represents only one team. There are many Stadiums to conduct matches. Each stadium is identified using Stadiumid, having a stadium_name,Address (involves city,area_name,pincode). A team can play many matches. Each match played between the two teams in the scheduled date and time in the predefined Stadium. Each match is identified uniquely by using Matchid. Each match won by any of the one team that also wants to record in the database. For each match man_of_the match award given to a player. Execute the following Queries: i. Display the youngest player (in terms of age) Name, Team name, age in which he belongs of the tournament. ii. List the details of the stadium where the maximum number of matches were played. iii. List the details of the player who is not a captain but got the man_of _match award at least in two matches. iv. Display the Team details who won the maximum matches. v. Display the team name where all its won matches played in the same stadium.
04	A country wants to conduct an election for the parliament. A country having many constituencies. Each constituency is identified uniquely by Constituency_id, having the Name, belongs to a state,Number_of_voters. A constituency can have many voters. Each voter is uniquely identified by using Voter_id, having the Name, age, address (involves Houseno,city,state,pincode). Each voter belongs to only one constituency. There are many candidates contesting in the election. Each candidates are uniquely identified by using candidate_id, having Name, phone_no, age, state. A candidate belongs to only one party.Thereare many parties. Each party is uniquely identified by using Party_id, having Party_Name,Party_symbol. A candidate can contest from many constituencies under a same party. A party can have many candidates contesting from different constituencies. No constituency having the candidates from the same party. A constituency can have many contesting candidates belongs to different parties. Each voter votes only one candidate of his/her constituencty. Queries: i. List the details of the candidates who are contesting from more than one constituencies which are belongs to different states. ii. Display the state name having maximum number of constituencies. iii. Create a stored procedure to insert the tuple into the voter table by checking the voter age.if voter's age is at least 18 years old, then insert the tuple into the voter else display "Not an eligible viter msg" iv. Create a stored procedure to display the number_of_voters in the specified constituency. Where the constituency name is passed as an argument to the stored procedure. v. Create a TRIGGER to UPDATE the count of Number_of_voters of the respective constituency in "CONSTITUENCY" table, AFTER inserting a tuple into the "VOTERS" table.

05	Design an ER-diagram for the following scenario, Convert the same into a relational model, normalize Relations into a suitable Normal form and then solve the following queries. A country can have many Tourist places . Each Tourist place is identified by using tourist_place_id, having a name, belongs to a state, Number of kilometers away from the 02.03.2021 updated 52/ 104 capital city of that state,history. There are many Tourists visits tourist places every year. Each tourist is identified uniquely by using Tourist_id, having a Name, age, Country and multiple emailids. A tourist visits many Tourist places, it is also required to record the visted_date in the database. A tourist can visit a Tourist place many times at different dates. A Tourist place can be visited by many tourists either in the same date or at different dates. Queries: i. List the state name which is having maximum number of tourist places. ii. List details of Tourist place where maximum number of tourists visited. iii. List the details of tourists visited all tourist places of the state "KARNATAKA" iv. Display the details of the tourists visited at least one tourist place of the state, but visited all states tourist places. v. Display the details of the tourist place visited by the tourists of all country.
06	Create an XHTML page that provides information about your department. Your XHTML page must use the following tags: a) Text Formatting tags b) Horizontal rule c) Meta element d) Links e) Images f) Tables (Use of additional tags encouraged).
07	Develop and demonstrate a XHTML file that includes Javascript script for the following problems: a) Input : A number n obtained using prompt Output : The first n Fibonacci numbers b) Input : A number n obtained using prompt Output : A table of numbers from 1 to n and their squares using alert
08	Develop and demonstrate, using JavaScript script, a XHTML document that contains three short paragraphs of text, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to become completely visible. Modify the above document so that when a text is moved from the top stacking position, it returns to its original position rather than to the bottom
	Demonstration Experiments (For CIE) if any
09	Consider the following database of student enrollment in courses and books adopted for each course. STUDENT (regno#: string, name: string, major: string, bdate: date) COURSE (course#: int, cname: string, dept: String) TEXT (book_ISBN#: int, book_title: string, publisher: string, author: string) ENROLL (regno#: string, course#: int, sem: int, marks: int) BOOK_ADOPTION (course#: int, sem: int, book_ISBN: int) • Create the above tables by properly specifying the primary keys and the foreign keys • Enter at least 7 to 10 records to each table. Execute SQL queries for the following requirements: 1. List out the student details, and their course details. The records should be ordered in a semester wise manner.

	2. List out the student details under a particular department whose name is ordered in a semester wise 3. List out all the book details under a particular course 4. Find out the Courses in which number of students studying will be more than 2. 5. Find out the Publisher who has published more than 2 books. 6. Find out the authors who have written book for I semester, computer science course. 7. List out the student details whose total number of months starting from their date of birth is more than 225 8. Find out the course name to which maximum number of students have joined
10	Develop, test and validate an XHTML document that has checkboxes for apple (59 cents each), orange (49 cents each), and banana (39 cents each) along with submit button. Each check boxes should have its own onclick event handler. These handlers must add the cost of their fruit to a total cost. An event handler for the submit button must produce an alert window with the message your total cost is $XXX where XXX is the cost of chose fruit, including 5% of sale tax. this handler must return false "(to avoid actual submission of the form data). Modify the document to accept quantity for each item using textboxes. **Course Outcomes (COs):** **After completion of the course student will be able to:** 1. Create database objects like table, constraint, view and index. 2. Design entity-relationship diagrams to solve given database applications. 3. Formulate SQL queries for the given problem. 4. Design simple web pages to demonstrate aspects of web application. 5. Develop, test and validate an XHTML document. ‘