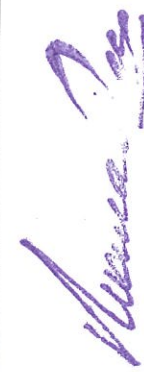


Sr. No.	Course/ Paper Code	Title of the Course	Mode of Learni ng	Course Type	Cre dit			Contact Hours per Week			Examination Scheme				Total Mar ks
					The ory	Practical/ Seminar	Total	The ory	Practical /Seminar	Total	End Sem ester Exa m	Internal Assesse nt	Practical (Exter nal)	Practical (Intern al)	
1	25BCAICC- N-201	Database Management system	CL/ BL	CC	3	1	4	3	2	5	50	20	20	10	100
2	25BCAICC- N-202	Object Oriented Programming Using Python	CL/ BL	CC	3	1	4	3	2	5	50	20	20	10	100
3	25BCAICC- N-203	Software Engineering	CL/ BL	CC	4	..	4	4	..	4	70	30	..	..	100
4	24BCAI-N- 204	Mathematical Foundation of Computer Science	CL/ BL	MIC	3	..	3	3	..	3	50	20	..	..	70
5	24UN-BBA- MDC201	Introduction to Entrepreneur ship Development	CL/ BL	MDC	3	..	3	3	..	3	50	20	..	..	70
6	24UN-ENG- AEC201	Business Communication	CL/BL	AEC	2	..	2	2	..	2	35	15	..	..	50
7	24BCAI-N- SEC-201	Front End Engineering	CL/ BL	SEC	2	1	3	2	2	4	35	15	15	5	70
8	24UN-PSY- VAC101	Human Values and Ethics	CL/ BL	VAC	2	..	2	2	..	2	35	15	..	..	50
Total							25			28					610

24BCAI- N - SEC -201	Front End Engineering	2L:0T:0P	2 Credits
-------------------------	-----------------------	----------	-----------

Maximum Marks:50
External Examination: 35
Internal Assessment: 15
Max. Time: 3 Hrs

Course Objectives

- To understand advanced tools of web technologies and develop front end applications using JavaScript.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Introduction to Web, Static and dynamic Pages, Introduction and fast revision to html tags and their usages, HTML Tags.

CSS: Selectors, Specificity, Selector Chaining, Box Model- Height & Width, Margin, Padding, Borders, Margin Collapse, Overflow, visibility.

JavaScript: Introduction and its advantages and disadvantages, Internal and external JS, Introduction to Variables and basics programming concepts in JavaScript, Scope of variables, conditional statements, loops Arrays accessing array elements using different methods.

Unit II

JSON and local storage: Functions: Type of function and their scope anonymous functions and arrow functions

Arrays Revisited: Iterators, foreach, filter, map and creating custom iterators.

Objects: Introduction to Object, managing Objects.

JSON and local storage: Nested JSON and Different JSON methods, Data transfer/exchange methods using JSON, Advantages and drawbacks of Using local storage and other possible ways, JSON handling different properties, Different ways of storing and retrieving the JSON

Unit III

DOM: Introduction to DOM, Accessing Elements of HTML using Dom, Various functions for handling DOM, Event Handling, ways of handling the events, Dynamic handling of DOM Elements, Dynamic Creation of different elements of DOM

Async Programming Introduction and its usage, JavaScript methods to perform background tasks, Timeout and Interval,

Async and await, Promises and handling promises, Callback and exceptions.

Unit IV

GIT Version Control System (Cycle, Clone, GIT log, diff, commit, push, pull, Branches, and contributions)

Introduction to React, FED in JavaScript and React, developing code in VS Code and introduction to webpack, Managing webpack and other components, Components in React

Course Outcomes

A student will be able to:

- Students will be able to understand and apply HTML5, CSS3, JavaScript to develop front End.
- Students will have sound practical know-how of using arrays, objects, JSON and local Storage.
- They will be able to manipulate the front-end using DOM manipulation.
- Understand the concept of Front-end development, API's, Async Programming.
- Students will be able to develop Single Page Applications using the component-based approach.

Suggested Readings:

1. Head First HTML and CSS – Elisabeth Robson & Eric Freeman
2. Web Design Playground: HTML & CSS the Interactive Way – Paul McFedries
3. Bootstrap 5 Foundations – Daniel Foreman



Lab-1 : Front End Engineering Lab (Based on 24BCAI- N - SEC -201)	0L:0T:4P	2 Credits
---	----------	-----------

Maximum Marks:20
External Examination: 15
Internal Assessment: 5

Course Outcomes: By the end of the course the student will be able to:

- To practically apply HTML5, CSS3, JavaScript to develop front End.
- Practically know-how of using arrays, objects, JSON and local Storage.
- To manipulate the front-end using DOM manipulation.
- Use the concept of Front-end development, API's, Async Programming.
- To develop Single Page Applications using the component-based approach of React.

List of Practical:

1. Book Review Blog, you have to create a simple blog page to display the contents of blog with pictures, you have to complete the required tasks.
2. Html Table- basic table generation to get into the understanding, you have to complete the task as defined
3. College Admission Form, create a basic admission form using html forms to capture the admission information from the user.
4. Share Your Recipe, your task is to display the contents of the Food Recipe with various components like ingredients, preparations time.
5. Chess Board Write a program in js that creates a chessboard that represent a grid using
 to separate lines. At each position of the grid there is either a " * " or a "#".
6. Calculate Bill, You have been given 2 arrays of objects (menu and categories) and another json to have bill, from these various object we need to calculate the Bill and generate a JSON.
7. To Do App – Basic Application, you have to build a basic to do app to manage the tasks and complete the user story as defined.
8. Discussion App - D1 Basic Discussion app to include questions and responses to all the questions, application will display questions on the left pane and their response on right pane on the click of question, till then displays the new question form to add new question.
9. Build Pomodoro Clock – Pomodoro is a time management technique developed in the 1980s which uses a timer to break down work into intervals, traditionally 25 minutes in length, separated by short breaks. In this assignment, you'll know how to create such a timer in the browser with JavaScript.
10. Web API, in this assignment, you need Build a basic app to fetch the records using API's with the help of AJAX. This assignment uses the Githubapi to access data using API.
11. Build a basic app to compile coding questions using API's with the help of AJAX. In this assignment you need to create a compiler app using the codequotientapi and follow the tasks.



25BCAICC-N-201	Database Management System	3L:0T:0P	3 Credits
----------------	----------------------------	----------	-----------

Maximum Marks:70
 External Examination: 50
 Internal Assessment: 20
 Max. Time: 3 Hrs

Course Objectives

- Understand the relational database design principles. Familiar with the basic issues of transaction processing and concurrency control.
- Familiar with SQL queries, database storage structures and access techniques.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

UNIT – I

Basic Concepts – Data, Information, Records and files. Traditional file – based Systems-File Based Approach-Limitations of File Based Approach, Database Approach-Characteristics of Database Approach, Database Management System (DBMS), Components of DBMS Environment, DBMS Functions and Components, Advantages and Disadvantages of DBMS, Roles in the Database Environment - Data and Database Administrator, Database Designers, Applications Developers and Users.

Data Models: Records- based Data Models, Object-based Data Models, Physical Data Models and Conceptual Modeling.

UNIT – II

Database System Architecture – Three Levels of Architecture, External, Conceptual and Internal Levels, Schemas, Mappings and Instances, Data Independence – Logical and Physical Data Independence, Normalization with various normal forms.

Entity-Relationship Model – Entity Types, Entity Sets, Attributes Relationship Types, Relationship Instances and ER Diagrams.

Relational Data Model - Terminology in Relational Data Structure, Relations, Properties of Relations, Keys, Domains, Integrity Constraints over Relations, Base Tables and Views, Basic Concepts of Hierarchical and Network Data Model.

UNIT – III

SQL fundamentals, data types, DDL, DML, DCL, Data Constraints, working with Tables, Defining different constraints on the table, Defining Integrity Constraints in the ALTER TABLE Command, Select Command, Logical Operator, Range Searching, Pattern Matching, Oracle Function, Grouping data from Tables in SQL, Manipulation Data in SQL.

UNIT – IV

Joining Multiple Tables (Equi Joins), Joining a Table to itself (self Joins), Sub queries Union, intersect & Minus Clause, Creating view, Renaming the Column of a view

Basics of PL/SQL: PL/SQL - Introduction and advantages Understanding, PL/SQL Block structure, Fundamentals of PL/SQL Language - data types, variables, constants and expressions, Operators, Conditional statement, Controlling loop iterations

Course Outcomes

A student will be able to:

- Gain knowledge of database systems and database management systems software.
- Ability to model data in applications using conceptual modeling tools such as ER Diagrams and design database schemas based on the model.
- Formulate, using SQL, solutions to a broad range of query and data update problems.
- Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.
- Be acquainted with the basics of transaction processing and concurrency control.

Suggested Readings:

1. Database System Concepts, 7th Edition, Silberschatz, Korth and Sudarshan, McGraw-Hill. Indian Edition released 2021 .
2. Fundamentals of Database Systems, 7th Edition, Elmasri and Navathe, Pearson Pubs, 2017 .
3. Principles of Database Management, Lemahieu, Broucke and Baesens, Cambridge University Press, 2018.



Lab-2 :Database Management Lab (Based on 25BCAICC-N-201)	0L:0T:2P	1 Credits
--	----------	-----------

Maximum Marks:30
External Examination: 20
Internal Assessment: 10

Course Outcomes: By the end of the course the student will be able to:

- To practically work with SQL Server and Oracle Database Server.
- To work with PL/SQL
- Implement triggers and stored procedures to handle complex business data.

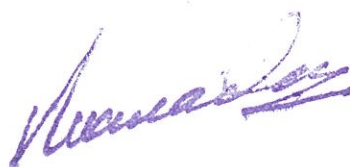
List of Practical:

1. To install the SQL Server.
2. The STUDENT detail database has a table with the following attributes. The primary keys are underlined. STUDENT(regno: int, name: string, dob: date, marks: int, city: string)
 - i) Create the above table.
 - ii) Remove the city attribute from the table.
 - iii) Change the date type of regno from integer to string.
 - iv) Add a new attribute phoneno to the existing table.
 - v) Enter five tuples into the table.
 - vi) Display all the tuples in the student table.
3. A LIBRARY database has a table with the following attributes. LIBRARY(bookid:int, title:string, author:string, publication:string, yearpub:int, price:real)
 - i) Create the above table.
 - ii) Enter the five tuples into the table
 - iii) Display all the tuples in the library table.
 - iv) Display the different publishers from the list.
 - v) Arrange the tuples in the alphabetical order of the book titles.
 - vi) List the details of all the books whose price ranges between Rs. 100 and Rs. 300
4. The SALARY database of an organization has a table with the following attributes. EMPSALARY(empcod:int, empname:string, dob:date, department:string, salary:real)
 - i) Create the above table.
 - ii) Enter the five tuples into the table
 - iii) Display all the number of employees working in each department.
 - iv) Find the sum of the salaries of all employees.
 - v) Find the sum and average of the salaries of employees of a particular department.
 - vi) Find the least and highest salaries that an employee draws.
5. Consider the insurance database given below. The primary keys are underlined and the data types are specified.

PERSON(driver-id-no: string, name: string, address:string)
CAR(regno: string, model: string, year: int)
ACCIDENT(report-no: int, date: date, location: String)
OWNS(driver-id-no: string, regno: string)
PARTICIPATED(driver-id-no: string, regno: string, report-no: int, damage-amount: int)

 - i) Create the above tables by properly specifying the primary keys and the foreign keys
 - ii) Enter at least five tuples for each relation.
 - iii) Demonstrate how you
 - a) Update the damage amount for the car with a specific regno in the accident with report no 12 to 25000.
 - b) Add a new accident to the database.
 - iv) Find total number of people who owned cars that were involved in accidents in 2002
 - v) Find the number of accidents in which cars belonging to a specific model were involved

6. Consider the following database of students enrollment in courses and books adopted for each course.
- STUDENT(regno: string, name: string, major: string, bdate: date)
 COURSE(course-no: int, name: string, dept: string)
 ENROLL(reg-no: string, course-no: int, sem: int, marks: int)
 BOOK-ADOPTION(course-no: int, sem: int, book-isbn: int)
 TEXT(book-isbn: int, book-title: string, publisher: string, author: string)
- Create the above tables by properly specifying the primary keys and the foreign keys
 - Enter atleast five tuples for each relation.
 - Demonstrate how you add a new text book to the database and make this book be adopted by some department.
 - Produce a list of textbooks (include Course-no, book-isbn, book-title) in the alphabetical order for courses offered by the 'Compute Science' department that use more than two books.
 - List any department that has all its adopted books published by a specific publisher.
7. The following tables are maintained by a book dealer
- AUTHOR(author-id: int, name: string, city: string, country: string)
 PUBLISHER(publisher-id: int, name: string, city: string, country: string)
 CATALOG(book-id: int, title : string, author-id: int, publisher-id: int, category: int, year: int, price: int)
 CATEGORY(category-id: int, description: string)
 ORDER-DETAILS(order-no: int, book-id: int, quantity: int)
- Create above tables by properly specifying the primary keys and the foreign keys.
 - Enter atleast five tuples for each relation.
 - Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2010.
 - Find the author of the book which has maximum sales.
 - Demonstrate how to increase price of books published by specific publisher by 10%
8. Consider the following database for BANK.
- BRANCH(branch-name: string, branch-city: string, assets: real)
 ACCOUNT(accno: int, branch-name: string, balance: real)
 DEPOSITOR(customer-name: string, accno: int)
 CUSTOMER(customer-name: string, customer-street: string, customer-city: string)
 LOAN(loan-no: int, branch-name: string, amount: real)
 BORROWER(customer-name: string, loan-no: int)
- Create the above tables by properly specifying the primary keys and foreign keys.
 - Enter atleast five tuples for each relation.
 - Find all the customers who have atleast two accounts at the main branch.
 - Find all customers who have an account at all the branches located in a specific city.
 - Demonstrate how to delete all account tuples at every branch located in a specific city.
9. Consider the following database for ORDER PROCESSING.
- CUSTOMER(cust-no: int, cname: string, city: string)
 ORDER(orderno: int, odate: date, ord-amt: real)
 ORDER_ITEM(orderno: int, itemno: int, qty: int)
 ITEM(itemno: int, unitprice: real)
 SHIPMENT(orderno: int, warehouseno: int, ship-date: date)
 WAREHOUSE(warehouseno: int, city: string)
- Create the above tables by properly specifying the primary keys and the foreign keys
 - Enter atleast five tuples for each relation.
 - List the order number and ship date for all orders shipped from particular warehouse
 - Produce a listing: customer name, no of orders, average order amount
 - List the orders that were not shipped within 30 days of ordering
10. Create two tables: motherboard (motherboard_Name, Price) and processor (processor_Name, Price) and insert 3 records in each table. Now write a query to display all maximum combinations of motherboard and processor with their price.
11. Write a PL/SQL program to find the largest of three numbers
12. Write a PL/SQL program to find the factorial of a number..



25BCAICC-N-202	Object Oriented Programming using Python	3L:0T:0P	3 Credits
----------------	--	----------	-----------

Maximum Marks:70
External Examination: 50
Internal Assessment: 20
Max. Time: 3 Hrs

Course Objectives:

- Understand the fundamental principles of Object-Oriented Programming (OOP) and their application in Python.
- Design and implement classes and objects to model real-world entities.
- Apply the concepts of encapsulation, inheritance, and polymorphism effectively.
- Handle exceptions gracefully in Python programs.
- Develop robust and maintainable Python applications using OOP principles.
- Gain a practical understanding of how OOP is used in frameworks and libraries.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

UNIT – I

Introduction to OOP and Classes & Objects: 1 **Introduction to OOP:** Procedural vs. Object-Oriented Programming. Key concepts of OOP: Class, Object, Encapsulation, Inheritance, Polymorphism, Abstraction. Benefits of OOP (modularity, reusability, maintainability, data security). **Classes and Objects in Python:** Defining a class (class keyword). Creating objects (instances) of a class.

Attributes (instance variables and class variables). Methods (instance methods, class methods, static methods - conceptual introduction). The self-parameter.

Constructors and Destructors: The init() method (constructor). Initializing object attributes. The del() method (destructor - brief discussion on garbage collection).

UNIT – II

Encapsulation and Data Hiding: Encapsulation: Concept of bundling data and methods that operate on the data within a single unit (class). Importance of data protection.

Access Modifiers/Conventions in Python: Public members (default). Protected members (single underscore _name). Private members (double underscore __name). Understanding the conventions and their implications (name mangling).

Getter and Setter Methods (Properties): Implementing encapsulation using getter and setter methods. The @property decorator for creating "Pythonic" getters and setters.

UNIT – III

Inheritance: Concept of Inheritance: Purpose and benefits of inheritance (code reusability, creating IS-A relationships). Base class (superclass) and Derived class (subclass).

Types of Inheritance: Single Inheritance, Multilevel Inheritance, Hierarchical Inheritance, Multiple Inheritance (understanding MRO - Method Resolution Order).

UNIT – IV

Polymorphism and Abstraction: Concept of "many forms. "Method Overloading (discussion of Python's approach - default arguments, variable-length arguments, *args, **kwargs). Method Overriding (revisit from Inheritance context).

Method Overriding and super(): Overriding methods in derived classes, Calling parent class methods using super().

Course Outcomes

- An understanding of the principles behind the object oriented development process.
- Competence in the use of object oriented programming language in the development of small to medium sized application programs.

Suggested Readings:

1. Kamthane, A. N., & Kamthane, A.A. Programming and Problem Solving with Python, McGraw Hill Education. 2017.
2. Balaguruswamy E., "Introduction to Computing and Problem Solving using Python", 2nd Edition, McGraw Hill Education, 2018.
3. Taneja, S., Kumar, N. Python Programming- A Modular Approach. Pearson Education India, 2018.



Lab-3 : Object Oriented Programming using Python	0L:0T:2P	1 Credits
--	----------	-----------

Maximum Marks:20
 External Examination: 15
 Internal Assessment: 5

Course Outcomes: By the end of the course the student will be able to:

List of Practical:

1. Define simple classes (e.g., Car, Student, Book).
2. Create multiple objects of these classes.
3. Implement instance and class attributes.
4. Define methods to interact with object data.
5. Use `__init__()` to initialize objects.
6. Refactor previous classes to use encapsulation principles.
7. Implement protected and private attributes using Pythonic conventions.
8. Create properties using the `@property` decorator for controlled attribute access.
9. Design and implement class hierarchies using single, multilevel, and hierarchical inheritance.
10. Experiment with multiple inheritance and observe MRO.
11. Practice method overriding and using `super()` to extend parent class functionality.



25BCAICC-N-203	Software Engineering	4L:0T:0P	4 Credits
----------------	----------------------	----------	-----------

Maximum Marks:100
External Examination: 70
Internal Assessment: 30
Max. Time: 3 Hrs

Course Objectives

- The objective of this course is to educate the students about the different models of software development and metrics used in software engineering.
- Students will get knowledge of Software Testing.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

UNIT – I

Introduction: Program vs. Software, Software Engineering, Programming paradigms, Software Crisis – problem and causes, Phases in Software development: Requirement Analysis, Software Design, Coding, Testing, Maintenance, Software Development Process Models: Waterfall, Prototype, Evolutionary and Spiral models, Role of Metrics.

UNIT – II

Feasibility Study, Software Requirement Analysis and Specifications: SRS, Need for SRS, Characteristics of an SRS, Components of an SRS, Problem Analysis, Information gathering tools, Organizing and structuring information, Requirement specification, validation and Verification.

UNIT – III

Structured Analysis and Tools: Data Flow Diagram, Data Dictionary, Decision table, Decision trees, Structured English, Entity-Relationship diagrams, Cohesion and Coupling. Gantt chart, PERT Chart, Software Maintenance: Type of maintenance, Management of Maintenance, Maintenance Process, maintenance characteristics.

UNIT – IV

Software Project Planning: Cost estimation: COCOMO model, Project scheduling, Staffing and personnel planning, team structure, Software configuration management, Quality assurance plans, Project monitoring plans, Risk Management. Software testing strategies: unit testing, integration testing, Validation testing, System testing, Alpha and Beta testing.

Course Outcomes

A student will be able to:

- Understand concept of Software Engineering and various SDLC; DSE-III(III).
- Learn to calculate the cost of a software project for an Enterprise; DSE-III(III).
- Gather basic information of an enterprise to design a software; DSE-III(III).
- Understand the fundamentals of Software Testing and Software Maintenance.



Suggested Readings:

1. Pressman S. Roger, Software Engineering, Tata McGraw Hill.
2. Jalote Pankaj, An Integrated Approach to Software Engineering, Narosa Publ. House.
3. K. K. Aggarwal, Yogesh Singh, Software Engineering, New Age International.
4. Sommerville, Software Engineering, Pearson Education.
5. Fairley Richard, Software Engineering Concepts; Tata Mc-Graw Hill Ed.
6. Rajib Mall, Fundamentals of Software Engineering, PHI Learning.

A handwritten signature in purple ink, appearing to read 'Anurag', is located at the bottom center of the page.

24BCAI-N-204	Mathematical Foundation of Computer Science	3L:0T:0P	3 Credits
--------------	--	----------	-----------

Maximum Marks:70
External Examination: 50
Internal Assessment: 20
Max. Time: 3 Hrs

Course Objectives

- This course helps the students to demonstrate various principles involved in solving mathematical problems and thereby reducing the time taken for performing job functions and solving problems.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Heights & Distances: Interest, Heights and Distances, Profit and Loss, Partnership, Mixture and Allegation, Time and distance Series, Time & Work, The Data Interpretation part covers Tabulation., Calendar, Clocks, Heights & Distances

Unit II

Number Problems: Numbers, H.C.F. & L.C.M. of Numbers, Decimal Fractions, Simplification, Square Roots & Cube Roots, Whole numbers problems, Decimals problems, Problems on Trains

Unit III

Database Modeling: Fractions problems, Numbers and Ages, Percentage problems, Boats and Streams, Ratio & Proportion, Square roots, Averages.

Unit IV

Statistics and Correlation: Meaning, Scope and Limitations of Statistics, Types and Sources of Data, Methods and Problems of Collection of Primary and Secondary Data

Measures of Central Tendency (Arithmetic Mean, Median, Partition Values, Mode), Measure of Dispersion (Absolute and Relative Measures Range, Quartile Deviation, Mean Deviation, Standard Deviation and Coefficient of Variation)

Correlation: Definition, Scatter diagram, Karl Pearson's coefficient of correlation, Numerical Problems for determination of Correlation coefficient. Regression: Definition, dependent and independent variables, least Square method only, Numerical problems. Permutations & Combinations, Probability,



Course Outcomes

A student will be able to:

- Common methodologies of solving questions related to Quantitative Aptitude and basics of Statistics.
- Learn the various question types of aptitude and their basic approaches.
- Create a bridge for the advanced levels of aptitude and prepare a base for problem solving and employability of the student.

Suggested Readings:

- 1) Quantitative Aptitude for Competitive Examinations by R. S. Aggarwal.
- 2) Business Statistics by TR Jain and SC Aggarwal

