ST. JOSEPH'S COLLEGE (AUTONOMOUS)

BENGALURU-27

Re-accredited with **'A++' GRADE with 3.79/4 CGPA** by NAAC Recognized by UGC as College of Excellence

DEPARTMENT OF COMPUTER SCIENCE AND COMPUTER APPLICATIONS

SYLLABUS FOR UNDERGRADUATE PROGRAMME

BSc (Computer Science as one major)

For Batch 2021-2025

DEPARTMENT OF COMPUTER SCIENCE ANDAPPLICATIONS
(BSc)(2021-2025)
SUMMARY OF CREDITS

Semester 1	Code Number	Title	No. of Hours of Instructions	Number of Hoursof teaching perweek	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CS121	Computer Fundamentals and Programming in C	52	04	04	40	60	100
Practical	CS1P1	C Programming Lab	52	04	02	25	25	50
Total Number of credits:			06					
Semester 2	Code Number	Title	No. of Hours of Instructions	Numberof teaching hrs. /week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CS221	Data Structures Using C	52	04	04	40	60	100
Practical	CS2P1	Data Structures Lab	52	04	02	25	25	50
Total Number of credits:			06					

Semester 3	**Code Number**	**Title**	**No. of Hours of Instructions**	**Number of teaching hrs. / week**	**Number of credits**	**Continuous Internal Assessment (CIA) Marks**	**End Semester Marks**	**Total marks**
Theory	CS321	Object Oriented Programming Concepts and Programming in JAVA	52	04	04	40	60	100
Practical	CS3P1	Java Lab	52	04	02	25	25	50
Total Number of credits:			**06**					

Semester 4	**Code Number**	**Title**	**No. of Hours of Instructions**	**Number of teaching hrs. /week**	**Number of credits**	**Continuous Internal Assessment (CIA) Marks**	**End Semester Marks**	**Total marks**
Theory	CS421	Database Management Systems and Software Engineering	52	04	04	40	60	100
Practical	CS4P1	DBMS Lab	52	04	02	25	25	50
Total Number of credits:			**06**					

Model Syllabus for BSc (Basic and Honors), Semesters I and II

Semester: I

Course Code: CS121	**Course Title: Computer Fundamentals and Programming in C**
Course Credits: 04	Hour of Teaching/Week: 04
Total Contact Hours:52	Formative Assessment Marks:40
Exam Marks: 100	Exam Duration: 03 Hrs

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Confidently operate Desktop Computers to carry out computational tasks
- Understand working of Hardware and Software and the importance of operating systems
- Understand programming languages, number systems, peripheral devices, networking,multimedia and internet concepts
- Read, understand and trace the execution of programs written in C language
- Write the C code for a given problem
- Perform input and output operations using programs in C
- Write programs that perform operations on arrays

Course Content

Conte nt	**Hour s**
Unit – 1	
Fundamentals of Computers: Introduction to Computers - Computer Definition, Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organisation of a Digital Computer; Memories -primary memory, secondary memory, cache memory. Number Systems – different types, conversion from one number system to another; Computer Codes – BCD, Gray Code, ASCII and Unicode; Boolean Algebra – Boolean Operators with Truth Tables; Types of Software – System Software and Utility Software; Computer Languages - Machine Level, Assembly Level & High Level Languages, Translator Programs – Assembler, Interpreter and Compiler; Planning a Computer Program - Algorithm, Flowchart and Pseudo code with Examples.	8
Unit – 2	
Introduction to C Programming: Over View of C; History and Features of C; Structure of a CProgram with Examples; Creating and Executing a C Program; Compilation process in C. **C Programming Basic Concepts:** C Character Set; C tokens - keywords, identifiers, constants, andvariables; Data types; Declaration & initialization of variables; Symbolic constants. **Input and output with C:** Formatted I/O functions - *printf* and *scanf,* control stings and escape sequences, output specifications with *printf* functions; Unformatted I/O functions	10

to read anddisplay single character and a string - *getchar, putchar, gets* and *puts* functions.	
Unit – 3	
C Operators & Expressions: Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associatively; Evaluation of arithmetic expressions; Type conversion.	12
Control Structures: Decision making Statements - S*imple if, if_else*, *nested if_else, else_if ladder,* *Switch-case*, *goto, break* & *continue* statements; Looping Statements - Entry controlled and Exitcontrolled statements, *while, do-while, for* loops, Nested loops.	
Unit – 4	
Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; TwoDimensional arrays - Declaration, Initialization and Memory representation. Strings: Declaring & Initializing string variables; String handling functions - *strlen, strcmp, strcpy andstrcat;* Character handling functions - *toascii, toupper, tolower, isalpha, isnumeric* etc. **User Defined Functions:** Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type	12
Unit – 5	
. **Pointers in C:** Understanding pointers - Declaring and initializing pointers, accessing address andvalue of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Advantages and disadvantages of using pointers; **User defined data types:** Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.	10

Text Books

1. Pradeep K. Sinha and Priti Sinha: Computer Fundamentals (Sixth Edition), BPB Publication
2. E. Balgurusamy: Programming in ANSI C (TMH)

References

1. Kamthane: Programming with ANSI and TURBO C (Pearson Education)
2. V. Rajaraman: Programming in C (PHI – EEE)
3. S. ByronGottfried: Programming with C (TMH)
4. Kernighan & Ritche: The C Programming Language (PHI)
5. Yashwant Kanitkar: Let us C
6. P.B. Kottur: Programming in C (Sapna Book House)

BLUE PRINT

Unit Nos	**No. of Hours**	**Total marks for which questions are to be asked (including bonus questions)**
Unit 1	10	32
Unit 2	8	28
Unit 3	12	29
Unit 4	12	31
Unit 5	10	18
TOTAL	52	138
Maximum marks for the paper (Excluding bonus questions) =100		

Course Code: CS1P1	**Course Title: C Programming Lab**
Course Credits: 02	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration:03 Hrs

Practice Lab

The following activities be carried out/ discussed in the lab during the initial period of the semester.

1. Basic Computer Proficiency
    a. Familiarization of Computer Hardware Parts
    b. Basic Computer Operations and Maintenance.
    c. Do's and Don'ts, Safety Guidelines in Computer Lab
2. Familiarization of Basic Software – Operating System, Word Processors, Internet Browsers,Integrated Development Environment (IDE) with Examples.
3. Type Program Code, Debug and Compile basic programs covering C Programmingfundamentals discussed during theory classes.

Programming

LabPart A:

1. Write a C Program to read radius of a circle and to find area and circumference
2. Write a C Program to read three numbers and find the biggest of three
3. Write a C Program to demonstrate library functions in *math.h*
4. Write a C Program to check for prime
5. Write a C Program to generate n primes
6. Write a C Program to read a number, find the sum of the digits, reverse the number andcheck it for palindrome
7. Write a C Program to read numbers from keyboard continuously till the user presses 999and to find the sum of only positive numbers
8. Write a C Program to read percentage of marks and to display appropriate message(Demonstration of else-if ladder)
9. Write a C Program to find the roots of quadratic equation (demonstration of switch-casestatement)
10. Write a C program to read marks scored by n students and find the average of marks(Demonstration of single dimensional array)
11. Write a C Program to remove Duplicate Element in a single dimensional Array
12. Program to perform addition and subtraction of Matrices

Part B:

1. Write a C Program to find the length of a string without using built in function
2. Write a C Program to demonstrate string functions.
3. Write a C Program to demonstrate pointers in C
4. Write a C Program to check a number for prime by defining *isprime( )* function
5. Write a C Program to read, display and to find the trace of a square matrix
6. Write a C Program to read, display and add two m x n matrices using functions
7. Write a C Program to read, display and multiply two m x n matrices using functions
8. Write a C Program to read a string and to find the number of alphabets, digits, vowels,consonants, spaces and special characters.
9. Write a C Program to Reverse a String using Pointer
10. Write a C Program to Swap Two Numbers using Pointers
11. Write a C Program to demonstrate student structure to read & display records of nstudents.
12. Write a C Program to demonstrate the difference between structure & union.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part B	Flowchart / Algorithm	
	Writing the Program	3
	Execution and Formatting	5
Program -2 from Part B	Flowchart/Algorithm	
	Writing the Program	3
	Execution and Formatting	5
Viva Voice based on C Programming		5
Practical Record		4
Total		**25**

Semester: II

Course Code: CS221	**Course Title: Data Structures using C**
Course Credits: 04	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 40
Exam Marks: 100	Exam Duration: 03 Hours

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Describe how arrays, records, linked structures, stacks, queues, trees, and graphs arerepresented in memory and used by algorithms
- Describe common applications for arrays, records, linked structures, stacks, queues, trees,and graphs
- Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs
- Demonstrate different methods for traversing trees
- Compare alternative implementations of data structures with respect to performance
- Describe the concept of recursion, give examples of its use
- Discuss the computational efficiency of the principal algorithms for sorting and searching

Course Content

Content	**Hours**
Unit – 1	
Introduction to data structures: Definition; Types of data structures - Primitive & Non-primitive, Linear and Non-linear; Operations on data structures. Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - *malloc*, *calloc*, *realloc* and *free.* Algorithm Specification, Performance Analysis, Performance Measurement-Asymptotic notations.	10
Unit – 2	
Arrays: Basic Concepts – Definition, Declaration, Initialization, Operations on arrays; Types of arrays; Arrays as abstract data types (ADT); Representation of Linear Arrays in memory; Traversing linear arrays; Inserting and deleting elements; Sorting – Selection sort, Bubble sort, Quick sort, Insertion sort; Searching - Sequential Search, Binary search; Iterative and Recursive searching; Multidimensional arrays; Representation of multidimensional arrays;	10
Unit – 3	

Stacks: Basic Concepts – Definition and Representation of stacks; Operations on stacks; Applications of stacks; Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression using stack; Application of stack in function calls.	10

Unit – 4	
Queues: Basic Concepts – Definition and Representation of queues; Types of queues- Simple queues, Circular queues, Double ended queues, Priority queues; Operationson Simple queues; **Linked list**: Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly liked list, Header liked list, Circular linked list;Representation of Linked list in Memory; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation; Garbage collection.	12
Unit – 5	
Trees: Definition; Tree terminologies –node, root node, parent node, ancestors of a node, siblings, terminal & non-terminal nodes, degree of a node, level, edge, path, depth; **Binary tree:** Type of binary trees - strict binary tree, complete binary tree, binary search tree and heap tree; Array and pointer representation of binary tree. Traversal of binary tree; preorder, inorder and postorder traversal;	10

Text Books

1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures

References

1. Tanenbaum: Data structures using C (Pearson Education)
2. Kamathane: Introduction to Data structures (Pearson Education)
3. Y. Kanitkar: Data Structures Using C (BPB)
4. Kottur: Data Structure Using C
5. Padma Reddy: Data Structure Using C
6. Sudipa Mukherjee: Data Structures using C – 1000 Problems and Solutions(McGraw Hill Education, 2007))

BLUE PRINT

Unit Nos	**No. of hours**	**Total marks for which questions are to be asked (including bonus questions)**
Unit 1	10	18
Unit 2	10	31
Unit 3	10	29
Unit 4	12	31
Unit 5	10	29
TOTAL	52	138
Maximum marks for the paper (Excluding bonus questions) =100		

Course Code: CS2P1	**Course Title: Data Structures Lab**
Course Credits: 02	Hour of Teaching/Week: 04
Total Contact Hours: 44	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 03 Hrs

Programming

LabPart A:

1. Write a C Program to find GCD using recursive function
2. Write a C Program to display Pascal Triangle using binomial function
3. Write a C Program to generate n Fibonacci numbers using recursive function.
4. Write a C Program to implement Towers of Hanoi.
5. Write a C Program to implement dynamic array, find smallest and largest element of thearray.
6. Write a C Program to create two files to store even and odd numbers.
7. Write a C Program to create a file to store student records.
8. Write a C Program to read the names of cities and arrange them alphabetically.
9. Write a C Program to sort the given list using selection sort technique.
10. Write a C Program to sort the given list using bubble sort technique.

Part B:

1. Write a C Program to sort the given list using insertion sort technique.
2. Write a C Program to sort the given list using quick sort technique.
3. Write a C Program to sort the given list using merge sort technique.
4. Write a C Program to search an element using linear search technique.
5. Write a C Program to search an element using recursive binary search technique.
6. Write a C Program to implement Stack.
7. Write a C Program to convert an infix expression to postfix.
8. Write a C Program to implement simple queue.
9. Write a C Program to implement linear linked list.
10. Write a C Program to display traversal of a tree.

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part A	Flowchart / Algorithm	
	Writing the Program	3
	Execution and Formatting	5
Program -2 from Part B	Flowchart/Algorithm	
	Writing the Program	3
	Execution and Formatting	5
Viva Voice based on C Programming		5
Practical Record		4
Total		**25**

III SEMESTER

Course Code: CA 321	**Course Title:** Object Oriented Programming Concepts and Programming in JAVA
Course Credits: 04	Hours / Week : 04
Total Contact Hours: 52	Formative Assessment Marks:40
Exam Marks:60	Exam Duration: 2 hrs.

Course Out comes (COs):
After completing this course satisfactorily, a student will be able to:

- Understand the features of Java and the architecture of JVM
  Write, compile, and execute Java programs that may include basic data types and control flow constructs and how typecasting is done.

- Identify classes, objects, members of a class and relationships among them needed for a specific problem and demonstrate the concepts of polymorphism and inheritance.

- The students will be able to demonstrate programs based on interfaces and threads and explain the benefits of JAVA's Exceptional handling mechanism compared to other Programming Language
  Write, compile, execute Java programs that include GUIs and event driven programming and also programs based on files

Course Content

Content	**Hours**
Unit -1	
Introduction to Java : Basics of Java programming, Data types, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, Math class, Arrays in java.	6
Unit -2	
Object Oriented Programming Concepts: Concept of programming paradigm, procedural paradigm and draw backs, object oriented paradigm concepts, OOP features – inheritance, polymorphism, encapsulation, abstraction and others (with examples), comparison of object oriented paradigm and other paradigms. OOP as a way of viewing world – Members and methods, Responsibilities, Classes and Instances, Summary of Object-Oriented concepts, Introducing classes, Methods and Classes, Constructors, Finalize, Visibility modifiers, Inbuilt classes like String, Character, String Buffer, File, this reference.	10
Unit -3	

Inheritance and Polymorphism: Inheritance in java, Super and subclass, Overriding, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java, Package in java, UTIL package.	10
Unit -4	
Event and GUI programming: Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows, Menus, Dialog Box, Applet and its life cycle, Introduction to swing, Exception handling mechanism. Contd.	10

Unit -5	
I/O Programming: Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files.	6
Unit -6	
Multithreading and Exception handling in java: Thread life cycle and methods, Runnable interface, Thread synchronization, Exception handling with try catch-finally, Collections in java, Introduction to Java Beans and Network Programming.	10

Text Books:

1. "Introduction to Java Programming" by Daniel Liang
2. Programming with Java, By E Balagurusamy – A Primer, Fourth Edition, Tata McGraw Hill Education Private Limited.
3. Core Java Volume I–Fundamentals, By Cay S. Horstmann, Prentice Hall
4. Object Oriented Programming with Java :Somashekara, M.T., Guru, D.S., Manjunatha, K.S

Reference Books:

1. Java 2-The Complete Reference–McGraw Hill publication.
2. Java - The Complete Reference, 7th Edition, By Herbert Schildt McGraw Hill publication.

BLUEPRINT

Unit Nos	No. of hours	Total marks for which questions are to be asked (including bonus questions)
Unit 1	6	6
Unit 2	10	21
Unit 3	10	20
Unit 4	10	20
Unit 5	6	6
Unit 6	10	10
TOTAL		83
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code: CA3P1	**Course Title:** JAVA Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 03 Hours

Course Outcomes (COs):

- After completing this course, student will be able to:
- Implement Object Oriented programming concept using basic syntax of control Structures.
- Identify classes, objects, members of a class and the relationships among them needed for finding the solution for the specific problem.
- Demonstrate show to achieve reusability using inheritance.
- Demonstrate understanding and use of interfaces, packages, different exception handling mechanisms and concept of multithreading for robust faster and efficient application development.
- Identify and describe common user interface components to design GUI in Java using Applet & AWT along with response to events.

Practice Lab List

1. Program to print the following triangle of numbers

```
1
12
123
1234
12345
```

2. Program to simple java application, to print the message, “Welcome to java”.

3. Program to display the month of a year. Months of the year should be held in an array.

4. Program to find the area of rectangle.

5. Program to demonstrate a division by zero exception
   Program to create a user defined exception say Pay out of Bounds.

Programming Lab

PART A: Java Fundamentals OOPs in Java

1. Program to assign two integer values to X and Y. Using the 'if' statement the Output of the program should display a message whether X is greater than Y.

2. Program to list the factorial of the numbers 1 to 10. To calculate the factorial value, use while loop.(Hint Fact of4=4*3*2*1).

3. Program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use function overloading.
4. Program to perform mathematical operations. Create a class called Add Sub with methods to add and subtract. Create another class called Mul Div that extends from Add Subclass to use the member data of the super class. Mul Div should have methods to multiply and divide A main function should access the methods and perform the mathematical operations.

5. Program with class variable that is available for all instances of a class. Use static variable declaration. Observe the changes that occur in the object's member variable values.

   Program

   i. To find the area and circumference of the circle by accepting the radius from the user.
   ii. To accept a number and find whether the number is Prime or not.

6. Program to create as student class with following attributes;
   Enrollment No: Name, Mark of sub1, Mark of sub2, mark of sub3, Total Marks. The three marks must be calculated only when the student passes in all three subjects. The pass mark for each subject is 50.If a candidate fails in any one of the subject, his total mark must be declared as zero. Using this condition write a constructor for this class. Write separate functions for accepting and displaying student details. In the main method create an array of three student objects and display the details.
   - In a college first year class are having the following attributes Name of the class (BCA, B.Com, BSc), Name of the staff No of the students in the class, Array of students in the class
   - Define a class called first year with above attributes and define a suit able constructor. Also write a method called best Student () which process a first-year object and return the student with the highest total mark. In the main method define a first-year object and find the best student of this class.

7. Program to define a class called employee with the name and date of appointment. Create ten employee objects as an array and sort them as per their date of appointment.ie, print them as per their seniority.

8. Create a package ‘student. Full time. BCA ‘in your current working directory
   - Create a default class student in the above package with the following attributes: Name, age, sex.
   - Have methods for storing as well as displaying.

PARTB: Exception Handling & GUI Programming.

1. Program to catch Negative Array Size Exception. This exception is caused when the array is initialized to negative values.
2. Program to handle Null Pointer Exception and use the “finally” method to Display a message to the user.
3. Program which create and displays a message on the window.
4. Program to draw several shapes in the created window.
5. Program to create an applet and draw grid lines.
6. Program which creates a frame with two buttons father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother similar details of mother also appear.
7. Create a frame which displays your personal details with respect to a button click
8. Create a simple applet which reveals the personal information of yours.
9. Program to move different shapes according to the arrow key pressed.
10. Demonstrate the various mouse handling events using suitable example.
11. Program to create menu bar and pull-down menus.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course.

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program–1 from Part A	Flowchart/Algorithm	
	Writing the Program	5
	Execution and Formatting	5
Program-2fromPart B	Flowchart/Algorithm	
	Writing the Program	5
	Execution and Formatting	5
Viva Voice based on Programming		5
Total		**25**

SEMESTER IV

Course Code: CS421	**Course Title: DATABASE MANAGEMENT SYSTEMS AND SOFTWARE ENGINEERING**
Course Credits:04	Hours/Week:04
Total Contact Hours: 52	Formative Assessment Marks:40
Exam Marks:60	Exam Duration: 02 Hrs.

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Understand database concepts and database management system software.
- Model an application's data requirements using conceptual modeling tools like ER diagrams.
- Write SQL commands to create tables and indexes, insert/update/delete data, and Query data in a relational DBMS.
- Understand the importance of software life cycle.
- Understand various aspects of design, coding, testing and reusability in Software development.

Course Content

Content	**Hours**
DATA BASE MANAGEMENT SYSTEM **Unit -1**	
INTRODUCTION: Data, database, DBMS, Disadvantages of File oriented systems, Advantages of DBMS, Database users, Database Languages, Characteristics of Database, Role of DBA, Data Abstraction (Views) – Logical, Conceptual & Physical, Data independence – physical and logical independence. **DATA MODELS:** Introduction to Data Models: E-R model, Relational model, network model and hierarchical model.	10
Unit –2	
RDBMS: Relational database concepts, attribute, tuple, types of attributes, single, multi-valued, stored, derived etc., keys, primary, index, candidate, alternate, foreign, Relationships, Relational algebra operations, union, intersection, difference, Cartesian product, selection, projection, join, division, relational calculus, Normalization and its properties, , I, II , III Normal forms and BCNF.	10

Unit – 3	
DDL and DML in SQL: DDL commands - create table/views/index, drop, alter, DML commands – select, insert, delete, update, etc., DCL commands – grant, revoke, commit, TCL commands, SQL – query, sub-query, nested query, Joins – natural, inner, outer join.	10
Unit – 4	
SOFTWARE ENGINEERING: Software and Software Engineering: Defining Software, Software Application Domains, Software Engineering, Software Process, Software Engineering Practice, Software Myths, Agile Development: Agility, Agility and the cost of change, Agile Process, Extreme Programming and Other Agile Process. Models. Understanding Requirements: Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Developing the use cases, Building the Requirements Model, Negotiating Requirements and Validating Requirements.	12
Unit – 5	
Requirements Modeling: Requirements Analysis, Scenario-Based Modeling, Design Concepts: The Design Process, Design Concepts, The Design Model, Architectural Design, Component-Level Design, User Interface Design, Pattern-Based Design, Quality Concepts: Software Quality, Review Techniques, Software Quality Assurance Software Testing Strategies: A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, System Testing, The Art of Debugging, Software Testing Fundamentals, White box Testing, Block-Box Testing.	10

Text Books:

1. "Remez Elmasri and Shamkant B. Navathe, "Fundamentals of Database Systems", 5th Edition, Pearson Education, 2007.
2. Abrahamsi. Silberschatz, Henry. F. Korth, S. Sudarshan, "Database System Concepts", 6th Edition, McGraw Hill, 2012.
3. Sundarraman, Oracle 9i programming A Primer,1/e Pearson Education.
4. Roger S. Pressman, "Software Engineering, A Practitioner's approach", 7th Edition, McGRAW-HILL Publication, 2010.
5. Pankaj Jalote, "An integrated approach to Software Engineering", 3rd Edition, Narosa Publishing House, 2013.

Reference Books:

1. C.J.Date, "Introduction to Database Systems", Eight Edition, Addison Wesley, 2003.
2. Karate, "Introduction to Database Management System", Pearson India, 2004
3. Ian Sommerville, "Software Engineering", 9th Edition, Pearson Education Ltd, 2010

BLUEPRINT

Unit No.	No. of hours	Total marks for which questions are to be asked (including bonus questions)
Unit 1	10	6
Unit 2	10	6
Unit 3	10	21
Unit 4	12	25
Unit 5	10	25
TOTAL	52	83
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code:CS4P1	**Course Title: DATABASE LAB**
Course Credits:02	Hours/Week:04
Total Contact Hours:52	Formative Assessment Marks:25
ExamMarks:25	Exam Duration: 03 Hours

Course Outcomes (COs):

This course will enable students to

- To understand basic database concepts, applications, data models, schemas and instances.
- Describe the basics of SQL and construct queries using SQL. Emphasize the importance of normalization in databases.

SQL Programming

- Design, develop, and implement the specified queries for the following problems using Oracle, MySQL, MS SQL Server, or any other DBMS under LINUX/Windows environment.
- Create Schema and insert at least 5 records for each table. Add appropriate database constraints.

1. Perform the following:
   - i) Creating a Database,
   - ii) Viewing all databases,
   - iii) Viewing all Tables in a Database,
   - iv) Creating Tables (With and Without Constraints),
   - v) Delete Table
   - vi) Rename Table.
2. Write SQL Queries involving:
   1. Date Functions,
   2. String Functions and
   3. Math Functions.
3. Create a table STATION to store information about weather observation stations with fields: ID(Primary Key), CITY, STATE, LAT, LONG and populate the table STATION with a few rows:
   - i) Write a SQL query to look at table STATION in undefined order
   - ii) Write a SQL query to select Northern stations (latitude > 39.7)
   - iii) Write a SQL query to select only ID, CITY, and STATE columns
   - iv) Write a SQL query to select only ID, CITY, and STATE columns where Longitude >45.

4. Create a table STATION to store information about weather observation stations with fields: ID (Primary Key), CITY, STATE, LAT, LONG. Duplicate ID fields are not allowed. Populate the table STATION with a few rows.
   Create another table called STATS to store normalized temperature and precipitation data:
    - ID field must match some STATION table ID (so that name and location will be known).
    - Allowable ranges will be enforced for other values.
    - No duplicate ID and MONTH combinations.
    - Temperature is in degrees Fahrenheit.
    - Rainfall is in inches.

   i) Populate the table STATS with some statistics for January and July.
   ii) Write a SQL query to look at table STATS in undefined order.
   iii) Write a SQL query to look at table STATS, picking up location information by joining with table STATION on the ID column.
   iv) Write a SQL query to look at the table STATS, ordered by month and greatest rainfall, with columns rearranged.
5. Create a table STATION to store information about weather observation stations with fields: ID(Primary Key), CITY, STATE, LAT, LONG. No duplicate ID fields allowed. Populate the table STATION with a few rows. Create another table called STATS to store normalized temperature and precipitation data.
    - ID field must match some STATION table ID (so name and location will be known).
    - Allowable ranges will be enforced for other values.
    - No duplicate ID and MONTH combinations.
    - Temperature is in degrees Fahrenheit.
    - Rainfall is in inches.

   i) Write a SQL query to look at temperatures for July from table STATS, lowest temperatures first, picking up city name and latitude by joining with table STATION on the ID column
   ii) Write a SQL query to show MAX and MIN temperatures as well as average rainfall for each station
   iii) Write a SQL query (with sub query) to show stations with year-round average temperature above 50 degrees.
       - Rows are selected from the STATION table based on related values in the STATS table.
6. Create table called STATS to store normalized temperature and precipitation data.
    - Allowable ranges will be enforced for other values.
    - No duplicate ID and MONTH combinations.
    - Temperature is in degrees Fahrenheit.
    - Rainfall is in inches.

   i) Create a view (derived table or persistent query) to convert Fahrenheit to Celsius and inches to centimeters
   ii) Add new column rainfall_centimeter to the table STATS.
   iii) Insert values into rainfall_centimeter from the view

iv) Delete Column rainfall from table STATS

7. Create table called STATS to store normalized temperature and precipitation data:
    - Allowable ranges will be enforced for other values.
    - No duplicate ID and MONTH combinations.
    - Temperature is in degrees Fahrenheit.
    - Rainfall is in inches.

   i) Write a SQL query to look at table STATS in a metric light (through the new view).
   ii) Write a SQL metric query restricted to January below-freezing (0 Celsius) data, sorted on rainfall.

8. Create table called STATS to store normalized temperature and precipitation data.
   i) Update all rows of table STATS to compensate for faulty rain gauges known to read 0.01 inches low
   ii) Update one row, ID 44's July temperature reading, to correct a data entry error
   iii) Make the above changes permanent
   iv) Undo that update

9. Create a table STATION to store information about weather observation stations with fields: ID(Primary Key), CITY, STATE, LAT, LONG. No duplicate ID fields allowed. Populate the table STATION with a few rows.
   Create another table called STATS to store normalized temperature and precipitation data.
    - ID field must match some STATION table ID (so name and location will be known).
    - Allowable ranges will be enforced for other values.
    - No duplicate ID and MONTH combinations.
    - Temperature is in degrees Fahrenheit.
    - Rainfall is in inches.

   i) Delete data from STATION table where longitude is >90
   ii) Delete July data from STATS table where longitude is >90
   iii) Increase the size for the column CITY with the following information:-

COLUMNNAME	DATATYPE(SIZE)
CITY	VARCHAR (25)

   iv) Modify the column name of LONG to LONGITUDE present in the STATION table and verify the result.

10. Create Table INSTRUCTOR with the following fields: InstuctID, InstructName, Department, Salary.
    i) Find instructors whose salary is more than the salary of any employee from department 'Physics'.

ii) Find the instructor name and department name of all instructors working in a department with any instructor whose name contains the letter “S”.

iii) Find the name and department name, of instructors whose salary is more than all their colleagues salaries in the same department.

11. Create table INSTRUCTOR with the following fields: InstuctID, InstructName, Department, Salary.

Create another table called STUDENT with the following fields: StudentID, StudentName, Department, InstuctID

i) Find the names of all instructors whose salary is greater than at least one instructor in the Finance department.

ii) Find all instructors whose salary is less than the salary of all instructors in the Computer Science department and whose department name is not Computer Science.

iii) Find the student name and department name of all student who study in a department with any student whose name contains the letter “S”.

12. Consider the following schema for a Library Database:

BOOK (Book_id, Title, Publisher_Name, Pub_Year)

BOOK_AUTHORS (Book_id, Author_Name)

PUBLISHER (Publisher_Name , Address, Phone)

BOOK_COPIES (Book_id, Branch_id, No-of_Copies)

BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out, Due_Date)

LIBRARY_BRANCH (Branch_id, Branch_Name, Address).

i) Draw the E_R Diagram for the Library database.

ii) Write SQL query to Retrieve details of all books in the library: id, title, name of publisher, authors, number of copies in each branch.

iii) Write an SQL query to compute the total number of books based on publisher name order by year.

13. Consider the schema for Movie Database:

ACTOR(Act_id, Act_Name, Act_Gender)

DIRECTOR(Dir_id, Dir_Name, Dir_Phone)

MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)

MOVIE_CAST(Act_id, Mov_id, Role)

RATING(Mov_id, Rev_Stars)

i) Draw the E_R Diagram for the Movie database.

ii) Write an SQL query to find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title. (Use Inner Join)

iii) Write an SQL query to retrieve all the actors and any movies they have acted in (Use left Join)

iv) Write an SQL query to retrieve all the directors and any movies they have directed.(Use right outer join)

v) Write an SQL query to retrieve all the directors name and all the movies (use Full join)

Evaluation Scheme for Lab Examination

Assessment Criteria	**Marks**
Creation of Table	2
Insertion of Tuples	3
SQL Queries	10
Viva Voice	5
Practical Record	5
Total	**25**