



BANGALORE UNIVERSITY

Jnanabharathi, Bengaluru, Karnataka – 560 056



SCHEME AND SYLLABUS

**For the Course
BSC SEP,NEP**

**BACHELOR OF SCIENCE
(Computer Science as an Optional)**

**Academic year(2022-2026),2023-2026,2024-2026) Batch
State Education Policy2023, 2024[1ST-4THSEM]
SCHEME NEP SCHEME 2022**

Members of the BOS in Computer Science

Sl. No.	Name	Designation
1.	Dr. Hanumanthappa M Senior Professor & Co-ordinator, Dept. of Computer Science, Bangalore University, Bangalore.	Chairperson
2.	Dr. Guru D.S Senior Professor, Dept. of Computer Science, University of Mysore, Mysore.	Member
3.	Dr. Ravikumar M Professor, Dept. of Computer Science, Kuvempu University, Shimoga.	Member
4.	Dr. Muralidhara B L Senior Professor, Dept. of Computer Science, Bangalore University, Bangalore.	Member
5.	Dr. Somashekara M.T Associate Professor, Dept. of Computer Science, Bangalore University, Bangalore.	Member
6.	Dr. Suresh R Assistant Professor, Dept. of Statistics, Bangalore University, Bangalore.	Member
7.	Mr. Hemanth Uppal Assistant Professor, Dept. of Master of Computer Science Application, Dayananda Sagar College of Arts, Science and Commerce, Bangalore.	Member
8.	Ms. Jayalaxmi R Assistant Professor, Dept. of Computer Science, St. Claret College, Bangalore.	Member
9.	Mr. Dadavali S.P Assistant Professor, Government First Grade College, Kengeri.	Member
10.	Ms. Ranjana Assistant Professor, Dept. of Computer Science, Global Institute of Management Science, Bangalore.	Member
11.	Mr. Praveen Kumar V Assistant Professor, Dept. of Computer Science, Acharya B School, Bangalore.	Member
12.	Mr. Shashidhara K.G Project Manager, IBM India, Bangalore.	Member
13.	Mr. Raveesha T.C CEO, Pearl Arc Systems Pvt. Ltd., Bangalore.	Member

BANGALORE UNIVERSITY
Department of Computer Science and Applications
B.Sc. (Computer Science as an Optional)
AS PER STATE EDUCATION POLICY[1ST TO 4TH SEM]
5TH SEM AND 6TH SEM [NEP]

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours/week	Semester End Exam	Internal Assessment	Total Marks	Credits
1	25CS11T	Programming in C	3	80	20	100	3
	CS11P	Programming in C Lab	3	40	10	50	2
2	25CS21T	Data Structures	3	80	20	100	3
	CS21P	Data Structures Lab	3	40	10	50	2
3	CS31T	Database Management System	3	80	20	100	3
	CS31P	Database Management System Lab	3	40	10	50	2
4	CS41T	Object Oriented Programming using Java	3	80	20	100	3
	CS41P	Object Oriented Programming using Java Lab	3	40	10	50	2
5	DSCS51	Computer Networks	3	60	40	100	3
	DSC51P	Computer Networks Lab	3	25	25	50	2
	DSCS52	Python	3	60	40	100	2
	DSCS52P	PYTHONLAB	3	25	25	50	2
6	DSCS62(B)	Artificial Intelligence	3	80	20	100	4
	DSCS62P	Artificial Intelligence LAB	2	25	25	50	2
	DSCS61	Web programming	3	80	20	100	4
	DSCS61P	Web programming LAB	3	25	25	50	2
		INTERNSHIP	2	30	20	50	2

Department of Computer Science and Applications

BANGALORE UNIVERSITY, BANGALORE

Program Outcome

PO1	Computational Knowledge	Acquire in-depth computational and mathematical knowledge with an ability to abstract and conceptualise from defined problems and requirements.
PO2	Dynamic Problem-Solving Skill	Identify, formulate, and exhibit strong analytical and dynamic problem-solving skills to address evolving computational challenges.
PO3	Innovative System Analysis and Design/ Development	Design and evaluate solutions for complex problems in Data Science, AI & ML, and Full Stack Development, considering societal, cultural, and environmental factors.
PO4	Investigate complex computing problem	Conduct literature surveys, analyze information, and design experiments using appropriate research methods to derive valid conclusions in relevant domains.
PO5	Use of modern tools/ Adaptive programming proficiency	Select, adapt, and apply modern IT tools and programming languages effectively in Data Science, AI & ML, and Full Stack Development to solve diverse computing challenges.
PO6	Knowledge Optimization	Modify algorithms or software systems to improve efficiency or resource utilization.
PO7	Life Long Continuous learning and Technology Adaptability	Pursue lifelong learning to stay updated with emerging technologies in Data Science, AI & ML, and Full Stack Development for sustained employability.
PO8	Soft skills and collaborative teamwork	Communicate effectively, enhance interpersonal skills, and collaborate in multidisciplinary teams essential for success in professional environments.
PO9	Cyber Security Proficiency	Understand cyber threats, develop secure software, and protect sensitive data in Data Science, AI & ML, and Full Stack Development applications.
PO10	Ethical and Professional Conduct	Adhere to ethical standards and professional practices in Data Science, AI & ML, and Full Stack Development roles and responsibilities.
PO11	Employability	Identify market trends, upgrade skills accordingly, and enhance employability in Data Science, AI & ML, and Full Stack Development careers.
PO12	Innovation and Entrepreneurship	Identify opportunities, innovate, and create value through Data Science, AI & ML, and Full Stack Development projects for personal growth and societal impact.

25CS11T:Programming in C

Course Outcomes

Upon successful completion of the course, the student will be able:

- CO1 To understand algorithmic strategies for enhancing problem-solving proficiency
- CO2 Demonstrate problem-solving tools and techniques using C.
- CO3 To analyze the given problems and use appropriate algorithms.
- CO4 To implement sorting and searching techniques to develop programs.

UNIT –I 12 Hours

Introduction: The Role of Algorithms in computing, Algorithms as a technology, analyzing algorithms, Designing algorithms, Flow charts. Fundamental Algorithms: Exchanging the values of two variables, Counting, Summation of a set of numbers, Factorial Computation, Generating of Fibonacci sequence, Reversing the digits of an integer, Character to number conversion.

UNIT-II 11 Hours

C Programming: Getting Started, Variables, Operators and Arithmetic expressions. Input and Output: Standard input and output, formatted input and output. Selection statements: Statements and Blocks, If, If-else, if-else-if ladder, nested if, switch. Control Structure: while loop, for loop, do-while loop, break and continue, goto and labels. Pointers and Arrays: Pointers and address, Pointers and function arguments, One Dimensional array, Two-Dimensional array, Structures and Union, Command line arguments.

UNIT - III 11 Hours

Factoring Methods: Finding the square root of a number, the smallest Divisor of an integer, the greatest common divisor of two integers, computing the prime factor of an integer, raising a number to a large power. Array Techniques: Array order reversal, Array counting, Finding the maximum number in a set, removal of duplicates from an ordered array, partitioning an array, finding the kth smallest element, multiplication of two matrices.

UNIT - IV 11 Hours

Sorting: Sorting by selection, sorting by exchange, sorting by insertion, sorting by diminishing increment, sorting by partitioning. Searching: Linear Search, Binary search. Text processing and Pattern searching: Text line length adjustment, keyboard searching in text, text line editing, linear pattern searching.

Text Book

- 1 R. G. Dromey, “How to Solve it by Computer”, Person Education India, 2008.
- 2 Brain M. Kernighan and Dennis M. Ritchie, “ The C Programming Language”, 2nd edition, Princeton Hall Software Series, 2012.
- 3 Thomas H Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, “Introduction to Algorithms”, 3rd Edition, The MIT Press Cambridge, Massachusetts London, England, 2008.

Reference Books

- 1 E. Balaguruswamy, “Programming In ANSI C”, 4th edition, TMH Publications, 2007
- 2 Greg Perry and Dean Miller, “C programming Absolute Beginner’s Guide”, 3rd edition, Pearson Education, Inc, 2014.
- 3 Donald E. Knuth, “The Art of Computer Programming”, Volume 2: Seminumerical Algorithms, 3rd Edition, Addison Wesley Longman, 1998.

Course Articulation Matrix: Mapping of Course Outcomes(COs) with Program Outcomes(POs1-12)

Course Outcome(COs)	Program Outcomes(POs)											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	3	1	1	1	1	2	1	1	1	2	1
CO2	3	3	3	2	1	1	1	1	1	1	2	1
CO3	3	3	1	1	1	1	1	1	1	1	2	1
CO4	3	3	2	2	1	1	1	1	1	1	2	1

Pedagogy: Lecture with the use of ICT/ Field Study / Assignment

Formative Assessment for Theory	
Assessment Occasion Type	Marks
C-1 Sessional Tests	5
C-1 Seminars/ Presentations	5
C-2 Sessional Tests	5
Case Study / Assignment / Project work etc.	5
Total	20 Marks
Formative Assessments as per SEP guidelines are compulsory	

CS11P:Programming in C

Write, and execute C Program for the following:

1. To read the radius of the circle and to find area and circumference.
2. To read the numbers and find the biggest of three.
3. To check whether the number is prime or not.
4. To find the root of quadratic equation.
5. To read a number, find the sum of the digits, reverse the number and check it for palindrome.
6. To read the numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers.
7. To read percentage of marks and to display appropriate message. If a percentage is 70 and above- Distinction, 60-69 – First Class, 50-59 – Second Class, 40-49 Pass, below 40 – Fail.
(Demonstrate of if-else ladder)

8. To simulate a simple calculator with addition, subtraction, multiplication, Division and it should display the error message for division of zero using switch case.
9. To read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
10. To remove duplicate elements in a single dimensional array.
11. To find the factorial of a number.
12. To generate Fibonacci series.
13. To design the following pattern using nested for loop:

```

      *
     * *
    * * *
   * * * *
  * * * * *

```

14. To find the length of the string without using built-in function.
15. To demonstrate string functions. (String Length, String Copy, String Concatenate, String Comparison)
16. To read, display and add two n x m matrices using function.
17. To read a string and to find the number of alphabets, digits, vowels, consonants, space and special characters.
18. To swap two numbers using pointers.
19. To demonstrate student structure to read & display records of n students.
20. To demonstrate the difference between structure and union for the following
Student name (String), Student roll no(integer), Student mark(float)

25CS21T:DATA STRUCTURES

Course Outcome

- | | |
|-----|---|
| CO1 | Understand basic concepts of data structures. |
| CO2 | Analyzing and exploring various ways of storing data using Array and Linked list. |
| CO3 | Demonstrate stack and queue data structures and their applications |
| CO4 | Analyze and implement various non-linear data structures. |

UNIT I

11 Hours

Introduction and Overview: Definition, Elementary data organization, Data Structures, data Structures operations, Abstract data types, algorithms complexity, time-space trade off. Preliminaries: Mathematical notations and functions, Algorithmic notations, control structures, Complexity of algorithms, asymptotic notations for complexity of algorithms.

UNIT II

11 Hours

Arrays: Definition, Linear arrays, arrays as ADT, Representation of Linear Arrays in Memory, Traversing Linear arrays, Inserting and deleting, multi-dimensional arrays, Matrices and Sparse matrices, searching and sorting techniques using array.

Linked list: Definition, Representation of Singly Linked List in memory, Traversing a Singly linked list, Searching in a Singly linked list, Memory allocation, Garbage collection, Insertion into a singly linked list, Deletion from a singly linked list; Doubly linked list, Header linked list, Circular linked list.

Stacks: Definition, Array representation of stacks, Linked representation of stacks, Stack as ADT, Arithmetic Expressions: Polish Notation, Conversion of infix expression to postfix expression, Evaluation of Postfix expression, Application of Stacks, Recursion, Towers of Hanoi, Implementation of recursive procedures by stack. Queues: Definition, Array representation of queue, Linked list representation of queues. Types of queue: Simple queue, Circular queue, Double-ended queue, Priority queue, Operations on Queues, Applications of queues.

Binary Trees: Definitions, Tree Search, Traversal of Binary Tree, Tree Sort, Building a Binary Search Tree, Height Balance: AVL Trees, Contiguous Representation of Binary Trees: Heaps, External Searching: Applications of Trees. Graphs: Mathematical Back ground, Computer Representation, Graph Traversal. Hashing: Hash Table ADT, understanding Hashing, Components of Hashing, Hash Table, Hash Function, Hashing Techniques, collisions, collision resolution techniques.

Text Book

- 1 Seymour Lipschutz, "Data Structures with C", Schaum's outLines, Tata Mc Graw Hill, 2011.
- 2 Robert Kruse, C.L.Tondo, Bruce Leung, Shashi Mogalla, "Data Structures and Program Design using C", Pearson Education, 2009

Reference Books

- 1 Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Second Edition, Pearson Education, 2013
- 2 Forouzan, "A Structured Programming Approach using C", 2nd Edition, Cengage Learning India, 2008.

Course Articulation Matrix: Mapping of Course Outcomes (COs) with Program Outcomes (POs)

Course Outcome (COs)	Program Outcomes (POs)											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	4	3	4	4	4	3	2	3	1	4	4
CO2	5	5	4	4	4	4	3	2	3	1	4	4
CO3	5	5	4	4	4	4	3	2	3	1	4	5
CO4	5	5	4	4	4	4	3	2	2	1	4	4

Pedagogy: Lecture with the use of ICT/ Field Study / Assignment

Formative Assessment for Theory	
Assessment Occasion Type	Marks
C-1 Sessional Tests	5
C-1 Seminars/ Presentations	5
C-2 Sessional Tests	5
Case Study / Assignment / Project work etc.	5
Total	20 Marks
Formative Assessments as per SEP guidelines are compulsory	

CS21P:DATA STRUCTURES LAB

NOTE: For all the programs write the output, flowchart and number of basic operations performed.

1. Write a program to search for an element in an array using binary and linear search.
2. Write a program to sort list of n numbers using Bubble Sort algorithms.
3. Perform the Insertion and Selection Sort on the input {75,8,1,16,48,3,7,0} and display the output in descending order.
4. Write a program to insert the elements {61,16,8,27} into singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
5. Write a program to insert the elements {45, 34, 10, 63,3} into linear queue and delete three elements from the list. Display your list after each insertion and deletion.
6. Write a program to simulate the working of Circular queue using an array.
7. Write a program to insert the elements {61,16,8,27} into ordered singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
8. Write a program for Tower of Hanoi problem using recursion.
9. Write recursive program to find GCD of 3 numbers.
10. Write a program to demonstrate working of stack using linked list.
11. Write a program to convert an infix expression $x^y/(5*z)+2$ to its postfix expression
12. Write a program to evaluate a postfix expression $5\ 3+8\ 2 - *$.
13. Write a program to create a binary tree with the elements {18,15,40,50,30,17,41} after creation insert 45 and 19 into tree and delete 15,17 and 41 from tree. Display the tree on each insertion and deletion operation.
14. Write a program to create binary search tree with the elements {2,5,1,3,9,0,6} and perform inorder, preorder and post order traversal.
15. Write a program to Sort the following elements using heap sort {9,16,32,8,4,1,5,8,0}.
16. Given $S1=\{\text{"Flowers"}\}$; $S2=\{\text{"are beautiful"}\}$ I. Find the length of S1 II. Concatenate S1 and S2 III. Extract the substring "low" from S1 IV. Find "are" in S2 and replace it with "is" .
17. Write a program to implement adjacency matrix of a graph.
18. Write a program to insert/retrieve an entry into hash/ from a hash table with open addressing using linear probing.

III SEMESTER	
PROGRAM NAME	B.Sc
TITLE OF THE COURSE	DATABASE MANAGEMENT SYSTEMS
COURSE CODE	CS31T
CREDITS	03
TOTAL NO. OF TEACHING HOURS	45

Course Objectives:
<p>To enable the students</p> <ul style="list-style-type: none"> • Introduce fundamental concepts and architecture of database systems and DBMS. • Develop understanding of data modelling using the Entity-Relationship (ER) model. • Teach core principles of the relational model, relational algebra, and SQL for database manipulation • Explain normalization techniques and database design for consistency and efficiency. • Explore transaction management, concurrency control, and recovery mechanisms.

Course Outcomes:
<p>By the end of this course student would be able to:</p> <p>CO1: Understand and explain the architecture, components, and functions of a DBMS.</p> <p>CO2: Design Entity-Relationship (ER) diagrams and convert them into relational schemas.</p> <p>CO3: Formulate and execute queries using relational algebra and SQL</p> <p>CO4: Apply normalization techniques to optimize database design.</p> <p>CO5: Demonstrate knowledge of transactions, concurrency control, and recovery mechanisms.</p>

COURSE CONTENT	
UNIT 1:	10 Hours
<p>Introduction to Databases: Definition of Data, Database, and DBMS, Overview of Database Applications, Advantages and Disadvantages of DBMS, Roles of Database Users and Administrators</p> <p>Data Models: Introduction to Data Models, Types of Data Models (Hierarchical, Network, Relational, Object-oriented), Importance of Data Models in DBMS</p> <p>Database Design: Keys: Primary Key, Candidate Key, Super Key, Foreign Key, Composite Key, Alternate Key, Unique Key, Surrogate Key, Constraints in a table: Primary Key, Foreign Key, Unique Key, NOT NULL, CHECK, Entity-Relationship (ER) Model, Entities and Entity Sets, Attributes and Relationships, ER Diagrams, Key Constraints and Weak Entity Sets, Extended ER Features, Introduction to the Relational Model and Relational Schema</p>	
UNIT 2:	13 Hours
<p>Relational Algebra and Calculus: Introduction to Relational Algebra, Operations: Selection, Projection, Set Operations, Join Operations, Division, Tuple and Domain Relational Calculus.</p> <p>Structured Query Language (SQL): SQL Basics: DDL and DML, Aggregate Functions (Min(), Max(), Sum(), Avg(), Count()), Logical operators (AND, OR, NOT), Predicates (Like, Between, Alias, Distinct), Clauses(Group By, Having, Order by, top/limit), Inner Join, Natural Join, Full Outer Join, Left Outer Join, Right outer Join, Equi Join</p>	

Advanced SQL: Analytical queries, Hierarchical queries, Recursive queries, Views, Cursors, Stored Procedures and Functions, Packages, Triggers, Dynamic SQL
Normalization and Database Design: Functional Dependencies: Armstrong's Axioms, Definition, Properties (Reflexivity, Augmentation, Transitivity), Types (Trivial, Non-Trivial, Partial and Full Functional Dependency), Closure of Functional Dependencies, Normal Forms (1NF, 2NF, 3NF, BCNF), Denormalization.

UNIT 3:

12 Hours

Transaction Management: ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions, Lock-Based Concurrency Control, Performance of Locking, Transaction Support in SQL, Introduction to Crash Recovery, 2PL, Serializability, and Recoverability, Introduction to Lock Management, Dealing with Deadlocks

Database Storage and Indexing: Data on External Storage, File Organizations and Indexing, Index Data Structures, Comparison of File Organizations, Indexes and Performance Tuning, Guidelines for Index Selection, Basic Examples of Index Selection.

UNIT 4:

10 Hours

NoSQL Databases and Big Data: Introduction to NoSQL, Data Models: Document, Key value, Column family, Graph. Uses and Features of NO/SQL document databases. CAP theorem, BASE vs ACID, CRUD operations, MongoDB operators, Overview of Big Data Technologies: Hadoop, MongoDB, Cassandra.

Database Security and Advanced Topics: Introduction to Database Security, Access Control, Discretionary Access Control, Introduction to Data Warehousing, OLAP, Data Mining.

Text Books:

1. Raghu Ramakrishnan, Johannes Gehrke, "Database Management Systems", third edition, McGraw – Hill, 2018
2. Benjamin Rosenzweig, Elena Rakhimov, "Oracle PL/SQL by Example", fifth edition, Prentice Hall, 2015
3. Brad Dayley, "NoSQL with MongoDB in 24 Hours", 1st edition, Sams Publishing, 2024.

Reference Books:

1. Korth, Silbertz, Sudarshan," Database System Concepts", Seventh Edition, McGraw - Hill.(2019)
2. R.P. Mahapatra, Govind Verma, "Database Management Systems", Khanna Publishing House, 2025.

Web Resources

1. <https://oracle-base.com/articles>
2. https://forums.oracle.com/ords/apexds/domain/dev-community/category/sql_and_pl_sql
3. <https://asktom.oracle.com/ords/f?p=100:1:0>

Pedagogy: Lecture with the use of ICT/ Field Study / Assignment

Course Articulation Matrix: Mapping of Course Outcomes(COs) with Program

Outcomes(POs1-12)

Course Outcome (COs)	Program Outcomes (POs)											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	4	4	5	4	4	3	2	3	1	4	5
C02	5	5	4	5	4	4	3	2	3	1	4	5
C03	5	5	4	5	4	4	3	3	3	1	4	5
C04	5	5	4	5	4	4	3	2	3	1	5	4

Formative Assessment for Theory	
Assessment Occasion Type	Marks
C-1 Sessional Tests	5
C-1 Seminars/ Presentations	5
C-2 Sessional Tests	5
Case Study / Assignment / Project work etc.	5
Total	20 Marks
Formative Assessments as per SEP guidelines are compulsory	

Cs41p: DATABASE MANAGEMENT SYSTEMS LAB

- Draw an E-R diagram and convert entities and relationships to relation tables for a given scenario (e.g., bank, college). Consider the Company database with the following schema:
 EMPLOYEE(FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO) DEPARTMENT(DNAME, DNUMBER, MGRSSN, MSRSTARTDATE) DEPT_LOCATIONS(DNUMBER, DLOCATION) PROJECT(PNAME, PNUMBER, PLOCATION, DNUM) WORKS_ON(ESSN, PNO, HOURS) DEPENDENT(ESSN, DEPENDENT_NAME, SEX, BDATE, RELATIONSHIP)
- Create a database and perform the following operations:
 - View all databases
 - Create tables (with and without constraints)
 - Insert, update, and delete records
 - Save (commit) and undo (rollback) changes
- Perform the following operations on database tables:
 - Alter a table (add/remove columns)
 - Rename a table
 - Drop/truncate a table
 - Backup and restore a database

3. Perform the following operations on database tables:
 - a) Alter a table (add/remove columns)
 - b) Rename a table
 - c) Drop/truncate a table
 - d) Backup and restore a database
4. Create tables for a given set of relation schemas and execute the following SQL queries:
 - a) Simple SELECT queries
 - b) Aggregate functions (SUM, AVG, MIN, MAX, COUNT)
 - c) Queries using GROUP BY and HAVING clauses
5. Execute queries to:
 - a) Increase the salary of all employees in the "Research" department by 10%
 - b) Find the sum, maximum, minimum, and average salary of all employees in the "Accounts" department
6. Execute queries to:
 - a) Retrieve the names of employees controlled by Department number 5 (use EXISTS operator)
 - b) Retrieve the name of each department and the number of employees working in each department with at least 2 employees
7. Execute queries to:
 - a) Retrieve the project number, project name, and the number of employees working on each project (use GROUP BY)
 - b) Retrieve the names of employees who were born in the 1990s
8. For each department with more than five employees, retrieve the department number and number of employees earning more than 40,000
9. For each project where more than two employees work, retrieve the project number, project name, and the number of employees working on that project
10. Create views based on a given set of relational tables and perform the following operations:
 - a) Create views (with and without CHECK OPTION)
 - b) Drop views
 - c) Select data from views
11. Create the following tables with Primary keys and Foreign keys and execute SQL queries:
BRANCH(Branchid, Branchname, HOD) STUDENT(USN, Name, Address, Branchid, Sem)
BOOK(Bookid, Bookname, Authorid, Publisher, Branchid) AUTHOR(Authorid, Authurname, Country, Age) BORROW(USN, Bookid, Borrowed_Date)
 - a) List students studying in 2nd-semester BCA
 - b) List students who haven't borrowed any books
12. Execute queries to:
 - a) Display USN, Student Name, Branch Name, Book Name, Author Name, and Books Borrowed Date of 2nd-semester BCA students who borrowed books
 - b) Display the number of books written by each author
13. Execute queries to:
 - a) Retrieve details of students who have borrowed more than two books

- b) Retrieve details of students who have borrowed books from more than one author
14. Execute queries to:
- a) Display book names in descending order
 - b) List students who borrowed books from the same publisher
15. Create the following table: STUDENT(USN, Name, Date_of_Birth, Branch, Mark1, Mark2, Mark3, Total, GPA)
- a) Find the GPA of all students
 - b) Find students who were born in a specific year
 - c) List students studying in a particular branch
 - d) Find the maximum GPA score branch-wise

IV SEMESTER	
PROGRAM NAME	B.Sc.(Computer Science)
TITLE OF THE COURSE	Object Oriented Programming using Java
COURSE CODE	24BSC-CS-4
CREDITS	03
TOTAL NO. OF TEACHING HOURS	45

Course Outcomes (COs): After the successful completion of the course, the student will be able to:

CO1 - Understand object-oriented programming concepts

CO2 - Demonstrate the important feature of Object-oriented programming

CO3 - Examine event handling, String handling, and exception handling concepts

CO4 - Implement concepts to solve real-world problems

UNIT-1 [12 Hours]

Introduction: Basics of object-oriented programming, comparison of procedure-oriented and object oriented programming paradigms; Difference between C and Java Programming languages; Features of Java; Objects and classes in Java, Structure of a Java program; Data Types, variables and operators in java; Control structures- Branching and looping; Methods & Constructors in java; Java Development Kit (JDK); Built-in classes in Java; Math, Character, String, String Buffer and Scanner; Wrapper classes; The abstract, static and final classes; Casting objects; The instance of operator; Usage of this keyword; Arrays in Java.

UNIT-2 [11 Hours]

Inheritance: Super and subclasses; visibility modifiers; Types of Inheritance- single, multiple, hierarchical and hybrid inheritance; the interface concept in Java, Polymorphism: Compile time and run time polymorphisms – Method overloading and method overriding. Package: Types of packages; the util, awt and swing packages; Creating and importing user-defined packages. I/O programming: Standard I/O streams in Java; Types of streams – Base3d on the type of Operations and the type of file.

UNIT-3 [11 Hours]

Event handling: Major events in Java; Two Event Handling mechanisms- Event classes and Event Listener Interfaces; Mouse and keyboard events; GUI: Panels; Frames; Layout managers – Flow, border and grid layouts; Buttons; Checkboxes; Radio buttons; Labels; Text fields; Text areas; Combo boxes; Scroll bars; Sliders; Menu, Dialog boxes. Applet programming: Comparison of applets and applications; Applet life cycle; Developing and running applets. String handling: String construction, string length, special string operations, character extraction, string comparison, modifying string and string buffers.

UNIT-4 [11 Hours]

Exception handling: Types of Java exception – checked and unchecked exceptions; Usage of trycatch-finally blocks. Multithreading: comparison of multithreading and multitasking; Life cycle of a thread; two ways of creating thread – by extending the Thread class and by implementing the Runnable

Interface, Thread synchronization. Advanced concepts: Collections in Java; Introduction to JavaBeans and Java security manager, Importance of generic programming in java with examples.

Text Books

- 1 E. Balagurusamy, Programming with JAVA, McGraw Hill, New Delhi, 2007
- 2 Herbert Schildt, Java A Beginner's Guide – Create, Compile, and Run Java Programs Today, Sixth Edition, Oracle Press, 2014

Reference Books

- 1 Ken Arnold, James Gosling, "The Java Programming Language, Fourth Edition, Addison Wisely, 2005
 - 2 R Herbert Schildt, 'The Complete Reference Java, 7th Edition, McGraw Hill, 2007
- Course Articulation Matrix: Mapping of Course Outcomes(COs) with Program Outcomes(POs1-12)

Course Outcome (COs)	Program Outcomes (POs)											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	4	3	2	2	3	2	3	2	2	4	2	2
C02	4	3	3	3	4	4	3	2	3	4	4	3
C03	4	4	3	3	4	3	3	2	3	4	3	4
C04	4	4	4	3	4	4	3	2	3	4	4	4

Formative Assessment for Theory	
Assessment Occasion Type	Marks
C-1 Sessional Tests	5
C-1 Seminars/ Presentations	5
C-2 Sessional Tests	5
Case Study / Assignment / Project work etc.	5
Total	20 Marks
Formative Assessments as per SEP guidelines are compulsory	

24BSC-CS-4P : OBJECT ORIENTED PROGRAMMING LAB

1. Java program to display "Hello World" and display the size of all the data types.
2. Java program to implement the usage of static, local and global variables.
3. Java program to implement string operations string length, string concatenate, substring
4. Java program to find the maximum of three numbers
5. Java program to check whether the number is odd or even.
6. Java program to implement default and parameterized constructors.
7. Java program to implement an array of objects.
8. Java program to implement Single Inheritance
9. Java program to implement Multiple Inheritance using Interface
10. Java program to implement the Life cycle of the applet
11. Java program to demonstrate a division by zero exception

12. Java program to add two integers and two float numbers. When no arguments are supplied give a default value to calculate the sum. Use method overloading.
13. Java program that demonstrates run-time polymorphism.
14. Java program to catch negative array size Exception. This exception is caused when the array is initialized to negative values.
15. Java program to handle null pointer exception and use the “finally” method to display a message to the user.
16. Java program to import user-defined packages
17. Java program to check whether a number is palindrome or not
18. Java program to find the factorial of a list of numbers reading input as command line argument.
19. Java program to display all prime numbers between two limits.
20. Java program to create a thread using Runnable Interface.

Program Name	B.Sc.	Semester	V
Course Title	Programming in Python (Theory)		
Course Code:	DSCSC51	No. of Credits	04
Contact hours	52 Hours	Duration of SEA/Exam	2 hours
Formative Assessment Marks	40	Summative Assessment Marks	60

Course Outcomes (COs): After the successful completion of the course, the student will be able to:

CO1	Setup python to develop simple applications	
CO2	Understand the basic concepts in Python Programming	
CO3	Learn how to write, debug and execute Python programs	
CO4	Understand and demonstrate the use of advanced data types such as tuples, dictionaries and lists, Tuples and Sets	
CO5	Design solutions for problems using object-oriented concepts in Python	
CO6	Use and apply the different Python Libraries for GUI Interface, Data Analysis and Data Visualisation.	52 Hrs
	Content	
C07	Extend the knowledge of python programming to build successful career in software development.	
Intro	Introduction to Features and Applications of Python; Python Versions; Installation of Python; Python Command Line mode and Python IDEs; Simple Python Program.	
	<p>Python Basics: Identifiers; Keywords; Statements and Expressions; Variables; Operators; Precedence and Association; Data Types; Indentation; Comments; Built-in Functions- Console Input and Console Output, Type Conversions; Python Libraries; Importing Libraries with Examples.</p> <p>Python Control Flow: Types of Control Flow; Control Flow Statements- if, else, elif, while loop, break, continue statements, for loop Statement; range () and exit () functions.</p>	10
	<p>Exception Handling: Types of Errors; Exceptions; Exception Handling using try, except and finally.</p> <p>Python Functions: Types of Functions; Function Definition- Syntax, Function Calling, Passing Parameters/arguments, the return statement; Default Parameters; Command line Arguments; Key Word Arguments; Recursive Functions; Scope and Lifetime of Variables in Functions.</p> <p>Strings: Creating and Storing Strings; Accessing Sting Characters; the str() function; Operations on Strings- Concatenation, Comparison, Slicing and Joining, Traversing; Format Specifiers; Escape Sequences; Raw and Unicode Strings; Python String Methods.</p>	10

<p>Lists: Creating Lists; Operations on Lists; Built-in Functions on Lists; Implementation of Stacks and Queues using Lists; Nested Lists.</p> <p>Dictionaries: Creating Dictionaries; Operations on Dictionaries; Built-in Functions on Dictionaries; Dictionary Methods; Populating and Traversing Dictionaries.</p> <p>Tuples and Sets: Creating Tuples; Operations on Tuples; Built-in Functions on Tuples; Tuple Methods; Creating Sets; Operations on Sets; Built-in Functions on Sets; Set Methods.</p>	10
<p>File Handling: File Types; Operations on Files– Create, Open, Read, Write, Close Files; File Names and Paths; Format Operator.</p>	10
<p>Object Oriented Programming: Classes and Objects; Creating Classes and Objects; Constructor Method; Classes with Multiple Objects; Objects as Arguments; Objects as Return Values; Inheritance- Single and Multiple Inheritance, Multilevel and Multipath Inheritance; Encapsulation- Definition, Private Instance Variables; Polymorphism- Definition, Operator Overloading.</p>	
<p>GU Interface: The Tkinter Module; Window and Widgets; Layout Management- pack, grid and place.</p> <p>Python SQLite: The SQLite3 module; SQLite Methods- connect, cursor, execute, close; Connect to Database; Create Table; Operations on Tables- Insert, Select, Update. Delete and Drop Records.</p> <p>Data Analysis: NumPy- Introduction to NumPy, Array Creation using NumPy, Operations on Arrays; Pandas- Introduction to Pandas, Series and DataFrames, Creating DataFrames from Excel Sheet and .csv file, Dictionary and Tuples. Operations on DataFrames.</p> <p>Data Visualisation: Introduction to Data Visualisation; Matplotlib Library; Different Types of Charts using Pyplot- Line chart, Bar chart and Histogram and Pie chart.</p>	12

Course Title	Python Programming Lab (Practical)	Practical Credits	02
Course Code	DSCSC51P	Contact Hours	04 Hours
Formative Assessment	25 Marks	Summative Assessment	25 Marks

Practical Content

Part-A

1. Check if a number belongs to the Fibonacci Sequence
2. Solve Quadratic Equations
3. Find the sum of n natural numbers
4. Display Multiplication Tables
5. Check if a given number is a Prime Number or not
6. Implement a sequential search
7. Create a calculator program
8. Explore string functions
9. Implement Selection Sort
10. Implement Stack
11. Read and write into a file

Part-B

1. Demonstrate usage of basic regular expression
2. Demonstrate use of advanced regular expressions for data validation.
3. Demonstrate use of List
4. Demonstrate use of Dictionaries
5. Create SQLite Database and Perform Operations on Tables
6. Create a GUI using Tkinter module
7. Demonstrate Exceptions in Python
8. Drawing Line chart and Bar chart using Matplotlib
9. Drawing Histogram and Pie chart using Matplotlib
10. Create Array using NumPy and Perform Operations on Array
11. Create Data Frame from Excel sheet using Pandas and Perform Operations on DataFrames

References

1	Think Python How to Think Like a Computer Scientist , Allen Downey et al., 2 nd Edition, 2015, Green Tea Press. Freely available online @ https://www.greenteapress.com/thinkpython/thinkCSpy.pdf
2	Introduction to Python Programming , Gowrishankar S et al., 2019, CRC Press
3	Python Data Analytics: Data Analysis and Science Using Pandas, matplotlib, and the Python Programming Language , Fabio Nelli, 2015, Apress®
4	Advance Core Python Programming , Meenu Kohli, 2021, BPB Publications
5	Core PYTHON Applications Programming , Wesley J. Chun, 3 rd Edition, 2012, Prentice Hall
6	Automate the Boring Stuff , Al Sweigart, 2015, No Starch Press, Inc.
7	Data Structures and Program Design Using Python , D Malhotra et al., 2021, Mercury Learning and Information LLC
8	http://www.ibiblio.org/g2swap/byteofpython/read/
9	https://docs.python.org/3/tutorial/index.html

Program Name	B.Sc.	Semester	V
Course Title	Computer Networks (Theory)		
Course Code:	DSCSC52	No. of Credits	04
Contact hours	52 Hours	Duration of SEA/Exam	2 hours
Formative Assessment Marks	40	Summative Assessment Marks	60

Course Outcomes (COs): After the successful completion of the course, the student will be able to:

CO1	Define various data communication components in networking.
CO2	Describe networking with reference to different types of models and topologies.
CO3	Understand the need for Network and various layers of OSI and TCP/IP reference model.
CO4	Explain various Data Communications media.
CO5	Describe the physical layer functions and components
CO6	Identify the different types of network topologies and Switching methods.
CO7	Describe various Data link Layer Protocols.
CO8	Analyze and Interpret various Data Link Layer and Transport Layer protocols.
CO10	Explain different application layer protocols.

Contents	52 Hrs
<p>Introduction: Computer Network: Definition, Goals, Structure; Broadcast and Point-To-Point Networks; Network Topology and their various Types; Types of Network, Network software, Design issues for the layers, Connection-oriented vs. Connectionless service, Applications of Computer network, Protocols and Standards, The OSI Reference Model, The TCP/IP Protocol suite, Comparison between OSI and TCP/IP Reference model.</p>	10
<p>Physical Layer: Functions of Physical Layer, Analog signals, Digital signals, Transmission Impairment, Data Rate Limits, and Performance. Data Transmission Media: Guided Transmission Media, Magnetic Media, Twisted Pairs, Coaxial Cable, Power Lines, Fiber Optics, Wireless Transmission, Electromagnetic Spectrum, Radio Transmission, Microwave Transmission, Infrared Transmission, Light Transmission, Digital Modulation and Multiplexing, Public Switched Telephone Networks. Switching: Circuit switching, Message switching & Packet switching</p>	12
<p>Data Link Layer: Functions of Data Link Layer, Data Link Control: Framing, Flow and Error Control, Error Detection and Correction, High-Level Data Link Control (HDLC) & point-to-point protocol (PPP), Channel Allocation Problem, Multiple Access: Random Access (ALOHA, CSMA, CSMA/CD, CSMA/CA), Controlled Access (Reservation, Polling, Token Passing), Channelization (FDMA, TDMA, CDMA),</p>	10

<p>Wired LAN: Ethernet Standards and FDDI, Wireless LAN: IEEE 802.11 and Bluetooth Standards.</p> <p>Transport Layer: Functions of Transport Layer, Elements of Transport Protocols: Addressing, Establishing and Releasing Connection, Flow Control & Buffering, Error Control, Multiplexing & De-multiplexing, Crash Recovery,</p>	10
<p>User Datagram Protocol (UDP): User Datagram, UDP Operations, Uses of UDP, RPC, Principles of Reliable Data Transfer: Building a Reliable Data Transfer Protocol,</p>	10
<p>Pipelined Reliable Data Transfer Protocol, Go Back-N (GBN), Selective Repeat (SR).</p> <p>Application layer : Functions of Application layer, Application Layer Protocols: DNS, DHCP, WWW, HTTP, HTTPS, TELNET, FTP, SMTP, POP, IMAP</p>	

References	
Reference Books:	
1	Andrew S Tanenbaum, David. J. Wetherall, —Computer Networks II, Pearson Education, 5th Edition,
2	Behrouz A. Forouzan, "Data Communications and Networking", Tata McGraw-Hill, Fourth Edition
3	Kurose and Ross, Computer Networking- A Top-Down approach, Pearson, 5th edition
4	William Stallings, Data and Computer Communications, 7th Edition, PHI.
4	http://higher.ed.mheducation.com/sites/0072967757/index.html
7	Larry L. Peterson, Bruce S. Davie, —Computer Networks: A Systems Approach II, Morgan Kaufmann Publishers, Fifth Edition, 2011.
8	Brijendrasingh, Data Communication and Computer Networks, PHI.

Course Title	Computer Networks Laboratory (Practical)	Practical Credits	02
Course Code	DSCSC52P	Contact Hours	04 Hours
Formative Assessment	25 Marks	Summative Assessment	25 Marks

**Practical
Content**

Part A:

1. Prepare hardware and software specification for basic computer system and Networking.
2. Study of different types of Network cables and practically implement the cross-wired cable and straight through cable using clamping tool.
3. Identifying the networking devices on a network.
4. Configure the IP address of the computer.
5. Create a basic network and share file and folders.
6. Study of basic network command and Network configuration commands.
7. Installation process of any open source network simulation software.

Part B:

1. Implement connecting two nodes using network simulator.
2. Implement connecting three nodes considering one node as a central node using network simulator. Implement a network to connect three nodes considering one node as a central node using network simulator
3. Implement bus topology using network simulator.
4. Implement star topology using network simulator.
5. Implement ring topology using network simulator.
6. Demonstrate the use of wireless LAN using network simulator.
7. Implement FTP using TCP bulk transfer using network simulator.
8. Implement connecting multiple routers and nodes and building a Hybrid topology network simulator.

Links for open source simulation software:

- NS3 software: <https://www.nsnam.org/releases/ns-3-30/download/>
- Packet Tracer Software: <https://www.netacad.com/courses/packet-tracer>
- GNS3 software: <https://www.gns3.com/>

