KARNATAKA STATE AKKAMAHADEVI WOMEN'S UNIVERSITY, VIJAYAPURA

Syllabus for Bachelor of Computer Application (III & IV Semester)

BCA

2021-22

National Education Policy - 2020 (NEP-2020)

SMT. A.S.M. COLLEGE FOR WOMEN * BELLARY

Curriculum for BCA

Sem	Core Courses	Hour / Week		DS Elective Courses	Hous / Week
		Theory	Lab		

1	**21BCADSC1:** Computers Fundamentals
	21BCADSC2: C Programming
	21BCADSC3: Digital Logic and Computer Design
	21BCADSC2P: C Programming Lab
	21BCADSC3P: Digital Logic and Computer Design Lab
		3
		3
		3
			3
			3
				OE1: (Open Elective)
				L1(Language 1)
				L2(Language 2)
				21SEC1: Digital Fluency
					3
					3
					3
					2

2	**21BCADSC4:** Discrete Mathematical Structures **21BCADSC5:** Data Structure **21BCADSC6:** Database Management System (DBMS) **21BCADSC5P:** Data Structure Lab **21BCADSC6P:** DBMS Lab	3 3 3	3 3	OE2: (Open Elective) L1(Language 1) L2(Language 2) AECC1: Environmental Studies Physical Education	3 3 3 2 1

3	**21BCADSC7:** Operating system Concepts
	21BCADSC8: C# and DOT NET Framework
	21BCADSC9: Python Programming
	21BCADSC8P: C# and DOT NET Framework Lab
	21BCADSC9P: Python Programming Lab
		3
		3
		3
			3
			3
				OE3: (Open Elective)
				L1(Language 1)
				L2(Language 2)
				21SEC2: Artificial intelligence
					3
					3
					3
					2

4	**21BCADSC10:** Software Engineering
	21BCADSC11: Object Oriented Concepts Using JAVA
	21BCADSC12: Design Analysis of Algorithms (DAA)
	21BCADSC11P: Java Programming Lab
	21BCADSC12P: DAA Lab
		3
		3
		3
			3
			3
				OE4: (Open Elective)
				L1(Language 1)
				L2(Language 2)
				AECC2: Constitution of India
					3
					3
					3
					2

Model Curriculum for BCA

Sem	Core Courses	Hour / Week		DS Elective Courses	Hours/ Week
		Theory	Lab		
III	Operating system Concepts	3			
	C# and DOT NET Framework	3			
	Python Programming	3			
	LAB: C# and DOT NET Framework Lab		4		
	LAB: Python Programming Lab		4		
IV	Software Engineering	3			
	Object Oriented Concepts Using JAVA	3			
	Design Analysis of Algorithms (DAA)	3			
	LAB: Java Programming Lab		4		
	LAB: DAA Lab		4		

Model Course Content for BCA, Semesters III and IV

Semester: III

Course Title: **Operating System Concepts**	Course code: 21BCADSC7
Total Contact Hours: 42	Course Credits: 03
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (COs):

At the end of the course, students will be able to:

- Explain the fundamentals of the operating system.
- Comprehend multithreaded programming, process management, process synchronization, memory management and storage management.
- Compare the performance of Scheduling Algorithms
- Identify the features of I/O and File handling methods.

Unit	Description	Hours
1	**Introduction to Operating System:** Definition, History and Examples of Operating System; Computer System organization; Types of Operating Systems; Functions of Operating System; Systems Calls; Operating System Structure.	08
2	**Process Management:** Process Concept- Process Definition, Process State, Process Control Block, Threads; Process scheduling- Multiprogramming, Scheduling Queues, CPU Scheduling, Context Switch; Operations on Processes- Creation and Termination of Processes; Inter process communication (IPC)- Definition and Need for Inter process Communication; IPC Implementation Methods- Shared Memory and Message Passing.	08
3	**Multithreaded Programming**: Introduction to Threads; Types of Threads; Multithreading- Definition, Advantages; Multithreading Models. **CPU Scheduling**: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling; Multiprocessor Scheduling; Real-Time CPU Scheduling.	08
4	**Process Synchronization:** Introduction; Race Condition; Critical Section Problem and Peterson's Solution; Synchronization Hardware, Semaphores; Classic Problems of Synchronization- Readers and Writers Problem, Dining Philosophers Problem; Monitors. **Deadlocks:** System Model; Deadlocks Characterization; Methods for Handling Deadlocks; Deadlock Prevention; Deadlock Avoidance; Deadlock Detection; and Recovery from Deadlock.	10

5	**Memory Management:** Logical and Physical Address Space; Swapping; Contiguous Allocation; Paging; Segmentation; Segmentation with Paging. **Virtual Memory:** Introduction to Virtual Memory; Demand Paging; Page Replacement; Page Replacement Algorithms; Allocation of frames, Thrashing.	08

References:

1. Operating System Concepts, Silberschatz' et al., 10thEdition, Wiley, 2018.
2. Operating System Concepts - Engineering Handbook, Ghosh PK, 2019.
3. Understanding Operating Systems, McHoes A et al., 7th Edition, Cengage Learning, 2014.
4. Operating Systems - Internals and Design Principles, William Stallings, 9th Edition, Pearson.
5. Operating Systems – A Concept Based Approach, Dhamdhere, 3rd Edition, McGraw Hill Education India.
6. Modern Operating Systems, Andrew S Tanenbaum, 4th Edition, Pearson.

Course Title: C# and Dot Net Framework	Course code: 21BCADSC8
Total Contact Hours: 42	Course Credits: 03+02
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (COs):

At the end of the course, students will be able to:

- Describe Object Oriented Programming concepts like Inheritance and Polymorphism in C# programming language.
- Interpret and Develop Interfaces for real-time applications.
- Build custom collections and generics in C#.

Unit	Description	Hours
1	**Introduction to .Net Technologies:** Introduction to Web Technologies. HTML Basics, Scripts. Sample Programs. Advantages and Disadvantages of Client-side and Server-side Scripts. Overview of Client-side Technologies and Server-side Technologies.	08
2	**Introduction to C#:** Overview of C#, Literals, Variables, Data Types, Operators, Expressions, Control Structures-Methods, Arrays, Strings, Structures, Enumerations. **OOPS with C#:** Classes, Objects, Inheritance, Polymorphism, Interfaces, Operator Overloading Delegates, Events, Errors and Exceptions.	08
3	**Introduction to VB.NET**: Introduction VB.NET -IDE – Creating a shortcut to start VB.NET. Maneuverings the Toolbar Auto-hide, Docking and Undocking, Placing and Resizing the Windows, Forms, Properties Window and Solution Explorer. Writing and Event Procedure. Execution Basic Keywords. Data Types. VB.NET statements. Conditional statements: If Else, Select Case, Switch and Choose Loops: Do, For Next, For Each Next, While loop. Arrays.	08
4	**Application Development on .NET:** C#.NET: Building Windows Applications, VB.NET: Windows Forms. Working with Controls, Timer, Picture-box, Group-box, Combo-box, Horizontal and Vertical Scrollbar, Numeric-up-down, Track-bar, and Progress-bar. Subroutines and Functions in VB.NET. Database applications	10
5	**ADO .NET Connectivity:** Introduction to ADO.NET, ADO vs ADO.NET. Architecture: Data reader, Data adopter, Accessing Data with ADO.NET. Programming Web Applications with Web Forms. ASP .NET applications with ADO.NET	08

References:

1. "Programming in C#", E. Balagurusamy, 4th Edition, Tata McGraw-Hill, 2017.
2. “Visual Basic.NET”, Shirish Chavan, 3rd Edition, Pearson Education, 2009.
3. “ASP.NET and VB.NET Web Programming”, Matt J. Crouch, Edition 2012.
4. "Computing with C# and the .NET Framework", Arthur Gittleman, 2nd Edition, Jones & Bartlett Publishers, 2011

Course Title: Python Programming	Course code: 21BCADSC9
Total Contact Hours: 42	Course Credits: 03+02
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (COs):

At the end of the course, students will be able to:

- Explain the basic concepts of Python Programming.
- Demonstrate proficiency in the handling of loops and creation of functions.
- Identify the methods to create and manipulate lists, tuples and dictionaries.
- Discover the commonly used operations involving file handling.
- Interpret the concepts of Object-Oriented Programming as used in Python.
- Develop the emerging applications of relevant fields using Python.

Unit	Description	Hours
1	**Introduction** to Features and Applications of Python; Python Versions; Installation of Python; Python Command Line mode and Python IDEs; Simple Python Program. **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.	08
2	**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. **Exception Handling:** Types of Errors; Exceptions; Exception Handling using try, except and finally.	08
3	**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. **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.	08
4	**Lists:** Creating Lists; Operations on Lists; Built-in Functions on Lists; Implementation of Stacks and Queues using Lists; Nested Lists. Dictionaries: Creating Dictionaries; Operations on Dictionaries; Built-in Functions on Dictionaries; Dictionary Methods; Populating and Traversing Dictionaries. **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. **File Handling:** File Types; Operations on Files– Create, Open, Read, Write, Close Files; File Names and Paths; Format Operator.	

5	**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. **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. **Data Visualisation:** Introduction to Data Visualisation; Matplotlib Library; Different Types of Charts using Pyplot- Line chart, Bar chart and Histogram and Pie chart.	08

References:

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

Course Title: C# and DOT NET Framework Lab	Course code: 21BCADSC8P
Total Contact Hours: 42	Course Credits: 02
Formative Assessment Marks: 25	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 25	

Course Outcomes (COs):
At the end of this Lab course students will be able to:

- Create user interactive web pages using C#.Net.
- Create simple data binding applications using ADO.Net connectivity.
- Performing Database operations for Windows Form and web applications

Program List:

1. Develop a C# .NET console application to demonstrate the conditional statements.
2. Develop a C# .NET console application to demonstrate the control statements.
3. Develop an application in C#.NET that demonstrates the windows controls
4. Demonstrate Multithreaded Programming in C#.NET
5. Demonstrate subroutines and functions in C#.NET
6. Develop an application for deploying various built-in functions in VB.NET
7. Develop an MDI application for Employee Pay-roll transactions in VB.NET
8. Construct a console application to demonstrate the OOP Concepts
9. Develop a web application in VB.NET for dynamic Login Processing
10. Develop a Windows application with database connectivity for core-banking transactions

Course Title: Python Programming Lab	Course code: 21BCADSC9P
Total Contact Hours: 42	Course Credits: 02
Formative Assessment Marks: 25	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 25	

Course Outcomes (COs):

At the end of this Lab course students

- Write, test, and debug simple Python programs.
- Implement Python programs with conditionals and loops.
- Develop Python programs step-wise by defining functions and calling them.
- Use Python lists, tuples, dictionaries for representing compound data.
- Read and write data from/to files in Python

Programs for Practical Component:

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. Write a python program to print a number is positive/negative using if-else.
5. Write a python program to find largest number among three numbers.
6. Write a python program to check whether the given string is palindrome or not.
7. Write a python program to find factorial of a given number using functions.
8. Write a Python function that takes two lists and returns True if they are equal otherwise false.
9. Using a numpy module create an array and check the following:
   1. Type of array 2. Axes of array 3. Shape of array 4. Type of elements in array.
10. Write a python program to open a file and check what are the access permissions acquired by that file using os module?
11. Write a python program which accepts the radius of a circle from user and computes the area (use math module).
12. Write a program to read 3 subject marks and display pass or failed using class and object.

Semester: IV

Course Title: Software Engineering	Course code: 21BCADSC10
Total Contact Hours: 42	Course Credits: 03+02
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (COs):

At the end of the course, students will be able to:

- How to apply the software engineering lifecycle by demonstrating competence in communication, planning, analysis, design, construction, and deployment
- An ability to work in one or more significant application domains
- Work as an individual and as part of a multidisciplinary team to develop and deliver quality software
- Demonstrate an understanding of and apply current theories, models, and techniques that provide a basis for the software lifecycle
- Demonstrate an ability to use the techniques and tools necessary for engineering practice

Unit	Description	Hours
1	**OVERVIEW:** Introduction: FAQ's about software engineering, Professional and ethical responsibility. Socio-Technical systems: Emergent system properties; Systems engineering; Organizations, people and computer systems; Legacy systems.	08
2	**CRITICAL SYSTEMS, SOFTWARE PROCESSES**: Critical Systems: A simple safety-critical system; System dependability; Availability and reliability. Software Processes: Models, Process iteration, Process activities; The Rational Unified Process; Computer-Aided Software Engineering.	08
3	**REQUIREMENTS:** Software Requirements: Functional and Non-functional requirements; User requirements; System requirements; Interface specification; The software requirements document. Requirements Engineering Processes: Feasibility studies; Requirements elicitation and analysis; Requirements validation; Requirements management.	08
4	**SYSTEM MODELS, PROJECT MANAGEMENT**: System Models: Context models; Behavioral models; Data models; Object models; Structured methods. Project Management: Management activities; Project planning; Project scheduling; Risk management.	
5	**SOFTWARE DESIGN**: Architectural Design: Architectural design decisions; System organization; Modular decomposition styles; Control styles. Object-Oriented design: Objects and Object Classes; An Object-Oriented design process; Design evolution. **DEVELOPMENT:** Rapid Software Development: Agile methods; Extreme programming; Rapid application development. Software Evolution: Program evolution dynamics; Software maintenance; Evolution processes; Legacy system evolution.	08

References:

TEXT BOOKS:

1. Ian Somerville, “Software Engineering” 8th Edition, Pearson Education, 2007.

REFERENCES BOOKS:

1.Waman S Jawadekar, “Software Engineering Principles and Practice”, Tata McGraw Hill, 2004.

2.Roger S. Pressman,“A Practitioners Approach”7th Edition, McGraw-Hill, 2007.

3. P Jalote, “An Integrated Approach to software Engineering” Narosa Publication.

Course Title: **Object Oriented Concepts Using JAVA**	**Course code: 21BCADSC11**
Total Contact Hours: 42	Course Credits: 03+02
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course outcomes: The students should be able to:

- Explain the object-oriented concepts and JAVA.
- Develop computer programs to solve real world problems in Java.
- Develop simple GUI interfaces for a computer program to interact with users

Unit	Description	Hours
1	An Overview of Java: Object-Oriented Programming, A First Simple Program, A Second Short Program, Two Control Statements, Using Blocks of Code, Lexical Issues, The Java Class Libraries, Data Types, Variables, and Arrays: Java Is a Strongly Typed Language, The Primitive Types, Integers, Floating-Point Types, Characters, Booleans, A Closer Look at Literals, Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays, A Few Words About Strings	08
2	Operators: Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses, Control Statements: Java's Selection Statements, Iteration Statements, Jump Statements.	08
3	Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection, The finalize( ) Method, A Stack Class, A Closer Look at Methods and Classes: Overloading Methods, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Recursion.	08
4	Introducing Access Control, Understanding static, Introducing final, Arrays Revisited, Inheritance: Inheritance, Using super, Creating a Multilevel Hierarchy, When Constructors Are Called, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, The Object Class.	
5	Packages and Interfaces: Packages, Access Protection, Importing Packages, Interfaces, Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Using Exceptions.	08

Text Books:

1. Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007. (Chapters 2, 3, 4, 5, 6,7, 8, 9,10, 12,13,15)

Reference Books:

1. Mahesh Bhave and Sunil Patekar, "Programming with Java", First Edition, Pearson Education,2008, ISBN:9788131720806.
2. Rajkumar Buyya,S Thamarasi selvi, xingchen chu, Object oriented Programming with java, Tata McGraw Hill education private limited.
3. E Balagurusamy, Programming with Java A primer, Tata McGraw Hill companies.
4. Anita Seth and B L Juneja, JAVA One step Ahead, Oxford University Press, 2017.

Course Title: **Design and Analysis of Algorithm**	Course code: **21BCADSC12**
Total Contact Hours: 52	Course Credits: 04 +02
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Objectives

- This course will teach paradigms and approaches used to analyze and design algorithms and the impact of algorithm design in practice.
- To make students understand how the time complexity of an algorithm is defined, how asymptotic notation is used to provide classification of algorithms.
- To explain different computational models (e.g., divide-and-conquer), order notation and various complexity measures (e.g., running time, diskspace) to analyze the complexity/performance of different algorithms.
- To teach various advance design and analysis techniques such as greedy algorithms, dynamic programming &know the concepts of tractable and intractable problems and the classes P, NP and NP-complete problems.

Unit	Description	Hours
1	**Design Of Efficient Algorithms And Elementary Data Structures:** Algorithms, Analysis of Algorithms, Time and Space Complexity, Running Time of A Program. Review of Stack, Queues, Trees.Operations on Stack, Queue and Trees.Recursion, Heaps and Heap Sort.	08
2	**Divide and Conquer:** General Method, Binary Search, Max and Min, Merge Sort, Quick Sort, Matrix Multiplication And Related Operations; Strassen's Matrix Multiplication, Inversion Matrices, LUP Decomposition And Its Application, Boolean Matrix Multiplication	08
3	**The Greedy Method:** The General Method, Knapsack Problem, Job Sequencing with Deadlines, Minimum Cost Spanning Trees: Prim's Algorithm, Kruskal's Algorithm. Optimal Storage on Tapes, Optimal Merge Patterns, Single Source Shortest Paths.	08
4	**Dynamic Programming:**The General Method, Multistage Graphs, All Pair's Shortest Paths,0/1 knapsack, Travelling Salesman Problem.	
5	**Backtracking:** General Methods, 8 – Queens Problem, Sum of Subsets, Knapsack Problem, NP – Hard and NP – Complete Problems.	08

Text Book:

1 Ellis, Horwitz, SartajSahani and Rajashekaran S., “Computer Algorithms”, (1999) Galgotia Publications Pvt.,Ltd.

Reference Books:

2 Aho A.V, Hopcroft J.E and Ullman, J.D., “The Design and Analysis of Computer Algorithms”, (1976) Addison – Wesley.

3 Sara Baase, Computer Algorithms, “An Introduction to Design and Analysis”, Addison Wesley.

Course Title: **Java Lab**	Course code: **21BCADSC11P**
Total Contact Hours: 42	Course Credits: 02
Formative Assessment Marks: 25	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 25	

Course Outcomes (COs):

At the end of the course, students will be able to:

- Identify classes, objects, members of a class and the relationships among them for a Specific problem.
- Develop programs using appropriate packages for Inter –thread Communication and Synchronization.
- Design, develop, test and debug Java programs using object-oriented principles in Conjunction with development tools including integrated development environments.

Program List:

1. Write a java program to find the Fibonacci series using recursive and non-recursive functions.
2. Write a java program to multiply two given matrices.
3. Write a java program for Method overloading and Constructor overloading.
4. Write a java program to display the employee details using Scanner class.
5. Write a java program that checks whether a given string is palindrome or not.
6. Write a java program to represent Abstract class with example.
7. Write a java program to implement Interface using extends keyword.
8. Write a java program to create user defined package.
9. Write a java program to create inner classes.
10. Write a java program for creating multiple catch blocks.
11. Write a java program for producer and consumer problem using Threads.
12. Write a Java program that implements a multi-thread application that has three threads.
13. Write a java program to display File class properties.
14. Write a java program to represent ArrayList class.
15. Write a Java program to load phone no, name from a text file using hash table.

Course Title: **Design and Analysis of Algorithm**	Course code: **21BCADSC12P**
Total Contact Hours: 42	Course Credits: 02
Formative Assessment Marks: 25	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 25	

Course Objectives

- This Lab will train students to write programs and solve problems using algorithm design techniques such as Divide and Conquer, Greedy, Dynamic programming, Back tracking and to analyze the running time of algorithms.
- To study about various designing paradigms of algorithms for solving real world problems.

Program List:

1. Program to construct a stack of integers and to perform the following operations on it:
   a. PUSH b. POP c. DISPLAY
2. Program to construct a queue of integers and to perform following operations on it:
   a. INSERT b. DELETE c. DISPLAY
3. Program to find factorial of a given number using recursion
4. Program to sort N numbers using Heap Sort
5. Program to search an element from a list using Binary Search Technique
6. Program to sort N numbers using Merge Sort
7. Program to implement Strassen's matrix multiplication
8. Program to find Maximum and Minimum elements in a matrix
9. Program to compute minimum spanning tree using Prim's Algorithm
10. Program to find shortest distance and shortest path from given source to all other nodes using Dijkstra's Algorithm
11. Program to implement all pair shortest problem using dynamic programming technique
12. Program to find GCD of two integers using Euclid's Algorithm

13. Evaluation Scheme for Lab Examination:

Assessment Criteria		**Marks**
Program – 1 from Part A	Writing the Program	03
	Execution and Formatting	07
Program -2 from Part B	Writing the Program	03
	Execution and Formatting	07
Viva Voice based on Python Programming and Practical Record		10
Total		30

Open Source Tools
(Skill Enhancement Course: SEC for BCA Course)

Semester: III

Course Title: **Open Source Tools**	Course Code: 21BCASEC1	Course Credits: 2 (1L+0T+2P)
Semester: III		Duration of SEE: 01 Hour
Total Contact Hours: 13 hours of theory and 26-28 hours of practicals		SEE: 30 Marks IA: 20 Marks

Course Outcomes (COs):

- Recognize the benefits and features of Open Source Technology and to interpret, contrast and compare open source products among themselves
- Use appropriate open source tools based on the nature of the problem
- Write code and compile different open-source software.

Course Content (Open Source Tools)

Module	Details of topic	Duration
Module 1: Open Source Softwares	i. Introduction to Open sources, Need of Open Sources, Open Source –Principles, Standard Requirements, Advantages of Open Sources – ii. Free Software – FOSS iii. Licenses – GPL, LGPL, Copyrights, Patents, Contracts & Licenses and Related Issues iv. Application of Open Sources. Open-Source Operating Systems: FEDORA, UBUNTU	05 hours
Module 2: Programming Tools And Techniques	i. Usage of design Tools like Argo UML or equivalent ii. Version Control Systems like Git or equivalent iii. Bug Tracking Systems (Trac, BugZilla) iv. BootStrap	04 hours
Module 3: Case Studies	**i.** Apache **ii.** Berkeley Software Distribution **iii.** Mozilla (Firefox) **iv.** Wikipedia **v.** Joomla **vi.** GNU Compiler Collection **vii.** Libre Office	04 hours

Text Book:

1. KailashVadera, Bhavyesh Gandhi, "Open Source Technology", Laxmi Publications Pvt. Ltd 2012, 1st Edition.

Reference Book:

1. Fadi P. Deek and James A. M. McHugh, "Open Source: Technology and Policy", Cambridge Universities Press 2007.

SMT. A.S.M. COLLEGE FOR WOMEN * BELLARY

KARNATAKA STATE AKKAMAHADEVI WOMEN'S UNIVERSITY, VIJAYAPURA

Curriculum Structure for

Bachelor of Computer Applications (BCA) Programme

Model Syllabus for I to II Semesters
and
Open Elective Courses in Computer Science

HOD,
Dept. of Computer Application,
Smt. A.S.M. College for Women,
BELLARY-583 103.

IQAC
CO-ORDINATOR
Smt. Allum Sumangalamma Memorial
College For Women, BELLARY.

PRINCIPAL
Smt. ASM College
For Women, BALLARI

2021-22

Preamble

Computer Application (CA) has been evolving as an important branch of science and technology in last two decade and it has carved out a space for itself like computer science and engineering. Computer application spans theory and more application and it requires thinking both in abstract terms and in concrete terms.

The ever -evolving discipline of computer application has strong connections to other disciplines. Many problems in science, engineering, health care, business, and other areas can be solved effectively with computers and its applications, but finding a solution requires both computer science expertise and knowledge of the particular application domain.

Computer science has a wide range of specialties. These include Computer Architecture, Software Systems, Graphics, Artificial Intelligence, Mathematical and Statistical Analysis, Data Science, Computational Science, and Software Engineering.

Universities and other HEIs introduced programmes of computer application. Information Technology is growing rapidly. Increasing applications of computers in almost all areas of human endeavour has led to vibrant industries with concurrent rapid change in technology. Unlike other basic disciplines, developing core competency in this discipline that can be reasonably stable becomes a challenge.

In India, it was initially introduced at the Master (postgraduate) level as MCA and M.Tech. Later on, engineering programmes such as B.Tech and B.E in Computer Science & Engineering and in Information Technology were introduced in various engineering College/Institutions to cater to the growing demand for trained engineering manpower in IT industries. Parallelly, BCA, BSc and MSc programmes with specialisation in Computer Science were introduced to train manpower in this highly demanding area.

BCA and BCA (Hons) are aimed at undergraduate level training facilitating multiple career paths. Students so graduated, can take up postgraduate programmes in CS or MCA leading to research as well as R&D, can be employable at IT industries, or can pursue a teaching profession or can adopt a business management career.

BCA and BCA (Hons) aims at laying a strong foundation of computer application at an early stage of the career. There are several employment opportunities and after successful completion of BCA, graduating students can fetch employment directly in companies as programmer, Web Developer, Software Engineer, Network Administrator, Data Scientist, or AI/ML personnel.

The Program outcomes in BCA are aimed at allowing flexibility and innovation in design and development of course content, in method of imparting training, in teaching learning process and in assessment procedures of the learning outcomes. The emphasis in BCA courses, in outcome-based curriculum framework, help students learn solving problems, accomplishing IT tasks, and expressing creativity, both individually and collaboratively. The proposed framework will help Students learn programming techniques and the syntax of one or more programming languages.

All students must, therefore, have access to a computer with a modern programming language installed. The computer science framework does not prescribe a specific language. The teacher and students will decide which modern programming languages students will learn. More importantly, students will learn to adapt to changes in programming languages and learn new languages as they are developed.

The present Curriculum Framework for BCA degrees is intended to facilitate the students to achieve the following.

- To develop an understanding and knowledge of the basic theory of Computer Science and Information Technology with good foundation on theory, systems and applications such as algorithms, data structures, data handling, data communication and computation
- To develop the ability to use this knowledge to analyse new situations in the application domain
- To acquire necessary and state-of-the-art skills to take up industry challenges. The objectives and outcomes are carefully designed to suit to the above-mentioned purpose.
- The ability to synthesize the acquired knowledge, understanding and experience for a better and improved comprehension of the real-life problems
- To learn skills and tools like mathematics, statistics and electronics to find the solution, interpret the results and make predictions for the future developments
- To formulate, to model, to design solutions, procedure and to use software tools to solve real world problems and evaluate

The objectives of the Programme are:

1. The primary objective of this program is to provide a foundation of computing principles and business practices for effectively using/managing information systems and enterprise software
2. It helps students analyze the requirements for system development and exposes students to business software and information systems
3. This course provides students with options to specialize in legacy application software, system software or mobile applications
4. To produce outstanding IT professionals who can apply the theoretical knowledge into practice in the real world and develop standalone live projects themselves
5. To provide opportunity for the study of modern methods of information processing and its applications.
6. To develop among students the programming techniques and the problem- solving skills through programming
7. To prepare students who wish to go on to further studies in computer science and related subjects.
8. To acquaint students to Work effectively with a range of current, standard, Office Productivity software applications

Program Outcomes: **BCA** (3 Years) **Degree**

1. **Discipline knowledge:** Acquiring knowledge on basics of Computer Science and ability to apply to design principles in the development of solutions for problems of varying complexity
2. **Problem Solving:** Improved reasoning with strong mathematical ability to Identify, formulate and analyze problems related to computer science and exhibiting a sound knowledge on data structures and algorithms.
3. **Design and Development of Solutions:** Ability to design and development of algorithmic solutions to real world problems and acquiring a minimum knowledge on statistics and optimization problems. Establishing excellent skills in applying various design strategies for solving complex problems.
4. **Programming a computer:** Exhibiting strong skills required to program a computer for various issues and problems of day-to-day applications with thorough knowledge on programming languages of various levels.
5. **Application Systems Knowledge**: Possessing a sound knowledge on computer application software and ability to design and develop app for applicative problems.
6. **Modern Tool Usage:** Identify, select and use a modern scientific and IT tool or technique for modeling, prediction, data analysis and solving problems in the area of Computer Science and making them mobile based application software.
7. **Communication:** Must have a reasonably good communication knowledge both in oral and writing.
8. **Project Management:** Practicing of existing projects and becoming independent to launch own project by identifying a gap in solutions.
9. **Ethics on Profession, Environment and Society:** Exhibiting professional ethics to maintain the integrality in a working environment and also have concern on societal impacts due to computer-based solutions for problems.
10. **Lifelong Learning:** Should become an independent learner. So, learn to learn ability.
11. **Motivation to take up Higher Studies:** Inspiration to continue educations towards advanced studies on Computer Science.

Additional Program Outcomes: **BCA Degree** (Hons)

The Bachelor of Computer Application (BCA (Hons)) program enables students to attain following additional attributes besides the afore-mentioned attributes, by the time of graduation:

1) Apply standard Software Engineering practices and strategies in real -time software project development
2) Design and develop computer programs/computer -based systems in the areas related to AI, algorithms, networking, web design, cloud computing, IoT and data analytics.
3) Acquaint with the contemporary trends in industrial/research settings and thereby
4) innovate novel solutions to existing problems
5) The ability to apply the knowledge and understanding noted above to the analysis of a
6) given information handling problem.
7) The ability to work independently on a substantial software project and as an effective
8) team member.

Curriculum for BCA

Sem	Core Courses	Hour / Week		DS Elective Courses	Hous / Week
		Theory	Lab		

1	**21BCADSC1:** Computers Fundamentals
	21BCADSC2: C Programming
	21BCADSC3: Digital Logic and Computer Design
	21BCADSC2P: C Programming Lab
	21BCADSC3P: Digital Logic and Computer Design Lab
		3
		3
		3
			3
			3
				OE1: (Open Elective)
				L1(Language 1)
				L2(Language 2)
				21SEC1: Digital Fluency
					3
					3
					3
					2

2	**21BCADSC4:** Discrete Mathematical Structures **21BCADSC5:** Data Structure **21BCADSC6:** Database Management System (DBMS) **21BCADSC5P:** Data Structure Lab **21BCADSC6P:** DBMS Lab	3 3 3	3 3	OE2: (Open Elective) L1(Language 1) L2(Language 2) **AECC1**: Environmental Studies Physical Education	3 3 3 2 1

3	**21BCADSC7:** Operating system Concepts
	21BCADSC8: C# and DOT NET Framework
	21BCADSC9: Python Programming
	21BCADSC8P: C# and DOT NET Framework Lab
	21BCADSC9P: Python Programming Lab
		3
		3
		3
			3
			3
				OE3: (Open Elective)
				L1(Language 1)
				L2(Language 2)
				21SEC2: Open-Source tools
					3
					3
					3
					2

4	**21BCADSC10:** Software Engineering
	21BCADSC11: Object Oriented Concepts Using JAVA
	21BCADSC12: Design Analysis of Algorithms (DAA)
	21BCADSC11P: Java Programming Lab
	21BCADSC12P: DAA Lab
		3
		3
		3
			3
			3
				OE4: (Open Elective)
				L1(Language 1)
				L2(Language 2)
				AECC2: Constitution of India
					3
					3
					3
					2

Model Course Content for BCA, Semesters I and II

Semester: I

Course Code: **21BCADSC1**	**Course Title:** Computer Fundamentals
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 03

Course Outcomes (COs):

- Introduction to computers, classification of computers, anatomy of computer, constituents and architecture, microcontrollers
- Operating systems, functions of operating systems, classification of operating systems, kernel, shell, basics of Unix, shell programming, booting
- Databases, why databases are used, users, SQL, data types in SQL, introduction of queries - select, alter, update, delete, truncate, using where, and or in not in
- Internet basics, features, applications, services, internet service providers, domain name system, browsing, email, searching
- Web Programming basics, introduction of HTML and CSS programming
- Introduction of computers, classification of computers, anatomy of computer, constituents and architecture, microcontrollers.

Course Content

Content	**Hours**
Unit – 1	
Fundamentals of Computers: Introduction to Computers - Computer Definition, Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organization of a Digital Computer; 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.	10

Unit-2	
Introduction to computers: Characteristics of computers, Classification of Digital Computer Systems: Microcomputers, Minicomputers, Mainframes, Super computers. Anatomy of Computer: Introduction, Functions & Components of a Computer, Central Processing Unit, Microprocessor, Storage units, Input and output Devices. How CPU and memory works. Program execution with illustrative examples. Introduction to microcontrollers.	08
Unit-3	
Operating System Fundamentals: Operating Systems: Introduction, Functions of an operating System, Classification of Operating Systems, System programs, Application programs, Utilities, The Unix Operating System, Basic Unix commands, Microkernel Based Operating System, Booting.	08
Unit-4	
Introduction to Database Management Systems: Database, DBMS, Why Database -File system vs DBMS, Database applications, Database users, Introduction to SQL, Data types, Classification of SQL-DDL with constraints, DML, DCL, TCL	08
Unit-5	
Internet Basics: Introduction, Features of Internet, Internet application, Services of Internet, Logical and physical addresses, Internet Service Providers, Domain Name System. **Web Basics:** Introduction to web, web browsers, http/https, URL, HTML5, CSS	08

Text Books:

1. Pradeep K. Sinha and Priti Sinha: Computer Fundamentals (Sixth Edition), BPB Publication, 2010.
2. David Riley and Kenny Hunt, Computational thinking for modern solver, Chapman & Hall/CRC, March 2014, ISBN: 9781466587793

Reference:

1. J. Glenn Brook shear," Computer Science: An Overview", Addision-Wesley, Twelfth Edition, June 2017.
2. R.G. Dromey, "How to solve it by Computer", Prentice-Hall International Series in computer science, C.A.R. HOARE Series Editor, PHI,ISBN: 0-13-433995-9.

Course Code: **21BCADSC2**	**Course Title: C Programming**
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 03

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

Content	Hours
Unit – 1	
Overview of C: History and importance of C; Structure of a C Program with Examples; Creating and Executing a C Program; Compilation process in C. **Constants, variables and data types:** Character Set; C tokens – keywords and identifiers, constants, and variables; Data types; Declaration & initialization of variables. **Input and output with C:** Formatted I/O functions - *printf* and *scanf,* Unformatted I/O functions to read and display single character and a string - *getchar, putchar, gets* and *puts* functions.	10
Unit – 2	
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. **Control Structures:** Decision making Statements - *Simple if, if_else, nested if_else, else_if ladder, Switch Case, goto, break* & *continue* statements; decision making and Looping Statements - Entry controlled and exit controlled statements, *while, do-while, for* loops, Nested loops.	08

Unit – 3	
Arrays: One Dimensional arrays - Declaration, Initialization; Two Dimensional arrays - Declaration, Initialization, Multidimensional array. **Strings:** Declaring & Initializing string variables; String handling functions - *strlen, strcmp, strcpy and strcat;* Character handling functions - *toascii, toupper, tolower, isalpha, isnumeric* etc.	08
Unit – 4	
User Defined Functions: Need for user defined functions; elements of user defined functions – return values and their types; function calls and declaration, category of function, recursion.	08
Unit – 5	
Pointers in C: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer expressions; Advantages and disadvantages of using pointers. **Structures and Unions** - Structure Definition, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.	08

Text Books:

1. E. Balaguruswamy: Programming in ANSI C (TMH), 6th edition 2012.
2. C: The Complete Reference, By Herbert Schildt. 4th Edition By Herbert Schildt 2000 | Published: October 6, 2000.
3. C Programming Language, By Brain W. Kernighan, 2nd Edition AT & T Bell Laboratories Murray Hill, New Jersey.

Reference Books:

1. P. K. Sinha & Priti Sinha: Computer Fundamentals (BPB)
2. Kamthane: Programming with ANSI and TURBO C (Pearson Education)
3. V. Rajaraman: Programming in C (PHI – EEE)
4. S. Byron Gottfried: Programming with C (TMH)
5. Yashwant Kanitkar: Let us C
6. P.B. Kottur: Programming in C (Sapna Book House)

Course Code: **21BCADSC3**	**Course Title: Digital Logic and Computer Design**
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 03

Course Outcomes (COs):

- Minimize functions using any type of minimizing algorithms (Boolean algebra, Karnaugh map or Tabulation Method).
- Implement functions using digital circuit (Combinational or Sequential)
- Have knowledge in analyzing and designing procedures of Combinational and Sequential circuits.
- Work effectively with others. - Use simulation software, for testing the designed circuit.

Content	**Hours**
Unit – 1	
Digital computers and digital systems: binary numbers, Number base conversion, Octal and Hexadecimal number system, Compliments, Binary codes, Binary storage and Registers, binary logic,IC.	08
Unit – 2	
Boolean Algebra: Definition of Boolean Algebra, Laws and Theorems and properties of Boolean algebra, Boolean functions, Canonical and standard forms, digital logic gates. **Simplifications of Boolean functions:** functions for 2, 3 and 4 variable maps, Sum-of-products, product of sums, Karnaugh Map, NAND and NOR implementation, Don'tCare conditions.	10
Unit – 3	
Combinational logic Circuits: Design procedure, half adders, full adder, half Subtractors, full subtractors, code conversion, multilevel NAND circuits, multilevel NOR circuits, Binary parallel adder, decimal adder, decoders, multiplexers.	08
Unit – 4	
Sequential Logic: Introduction, flip-flops, Clocked RS flip-flop, D flip-flop, JK flip-flop, clocked T flip-flop, triggering of flip-flops, Master-slave flip-flop, edge triggered flip-flop , analysis of clocked sequential circuits, design of counters.	08
Unit – 5	
Registers: Registers, Types of registers, Serial In-Serial Out, Serial in Parallel Out, Parallel In-Serial Out, Parallel- in- Parallel Out, Application of Shift Register, Ripple counters, Synchronous Counters, Timing sequence, The memory unit.	08

Text-book:

1. Morris Mano M., "Digital logic and Computer Design", PHI, 3rd edition , 2013.

Reference Books:

1. V. Rajaraman, T. Radhakrishnan, "An Introduction to Digital Computer Design", PHI
2. Stephen Brown, Zvonko Vranesic, "Fundamentals of Digital Logic with Verilog Design", TMH, 2006, Charles H. Roth, Jr. 5th

Course Code: **21BCADSC2P**	**Course Title: C Programming Lab**
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 42	Formative Assessment Marks:25
Exam Marks: 25	Exam Duration: 03

Program List

1) Program to read radius of a circle and to find area and circumference
2) Program to read three numbers and find the biggest of three
3) Program to demonstrate library functions in math.h
4) Program to check for prime
5) Program to read a number, find the sum of the digits, reverse the number and check it for palindrome
6) Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers
7) Program to find the roots of quadratic equation (demonstration of switch Case statement)
8) Program to read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
9) Program to remove Duplicate Element in a single dimensional Array
10) Program to find the length of a string without using built in function
11) Program to demonstrate string functions.
12) Program to demonstrate pointers in C
13) Program to read, display and to find the trace of a square matrix
14) Program to read, display and add two m x n matrices using functions
15) Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
16) Program to Reverse a String using Pointer
17) Program to Swap Two Numbers using Pointers
18) Program to demonstrate student structure to read & display records of nstudents.

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	Writing the Program	04
	Execution and Formatting	04
Program -2 from Part B	Writing the Program	04
	Execution and Formatting	04
Viva Voice based on C Programming		05
Practical Record		04
Total		**25**

Course Code: **21BCADSC3P**	**Course Title: Digital Logics and Computer Design Lab**
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 42	Formative Assessment Marks:25
Exam Marks: 25	Exam Duration: 03

Experiments

1. Digital logic assignments shall be carried out using trainer kit or simulation software toillustrate the concepts studied in the paper Digital Logic and Computer Design.
2. Implement basic gates, universal gates and advanced gates.
3. Implementation of given Boolean function in both SOP and POS forms.
4. Verification of state tables of RS-, JK-, T-, D- flip-flops.
5. Implementation and verification of de-coder, de-multiplexer and encoder using logicgates.
6. construct half adder circuit and verify it's working.
7. construct full adder circuit and verify it's working.
8. perform Binary to Gray Code Conversion.
9. perform gray code to binary code conversion.
10. Implementation of `4x1 multiplexer using logic gates.
11. Implementation of 4-bit parallel adder using 7483 IC. Design and verification of 4-bitsynchronous counter.

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part B	Writing the Program	04
	Execution and Formatting	04
Program -2 from Part B	Writing the Program	04
	Execution and Formatting	04
Viva Voice based on C Programming		05
Practical Record		04
Total		**25**

Semester: II

Course Code: **21BCADSC4**	**Course Title: Discrete Mathematical Structures**
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 03 Hours

Course Outcomes (COs):

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

- To understand the basic concepts of Mathematical reasoning, set and functions.
- To understand various counting techniques and principle of inclusion and exclusions.
- Understand the concepts of various types of relations, partial ordering and
- equivalence relations.
- Apply the concepts of generating functions to solve the recurrence relations.
- Familiarize the fundamental concepts of graph theory and shortest path algorithm

Course Content

Content	Hours
Unit – 1	
The Foundations: Logic and proofs: Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy. **Basic Structures**: Sets, Functions, Sequences, Sums, and Matrices: Sets, set operations, Functions, Sequences and Summations, matrices.	10
Unit – 2	
Counting: Basics of counting, Pigeonhole principle, Permutation and combination, Binomial Coefficient and Combination, Generating Permutation and Combination. **Advanced Counting Techniques**: Applications of Recurrence Relations, Solving Linear Recurrence, Relations, Divide and Conquer Algorithms and Recurrence Relations, Generating functions, Inclusion-Exclusion, Applications of Inclusion-exclusion.	08
Unit – 3	
Induction and Recursion: Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Corrections.	08

Unit – 4	
Relation: Properties of relation, Composition of relation, Closer operation on relation, Equivalence relation and partition. Operation on relation, Representing relation.	08
Unit – 5	
Graphs: Graphs and Graph models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring.	08

Text Book:

1. Discrete Mathematics and Its Applications, Kenneth H. Rosen: Seventh Edition, 2012.

References:

2. Discrete Mathematical Structure, Bernard Kolman, Robert C, Busby, Sharon Ross, 2003.
3. Graph Theory with Applications to Engg and Comp. Sci: Narsingh Deo-PHI 1986.
4. Discrete and Combinatorial Mathematics Ralph P. Grimaldi, B. V. Ramatta, Pearson, Education, 5 Edition.
5. Discrete Mathematical Structures, Trembley and Manobar.

Course Code: **21BCADSC5**	**Course Title: Data Structures**
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	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 are represented 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, searching, and hashing

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 Recursion: Definition; Types of recursions; Recursion Technique Examples - GCD, Binomial coefficient ^nC_r, Towers of Hanoi; Comparison between iterative and recursive functions.	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, Selection sort, Insertion sort; Searching - Sequential Search, Binary search; Iterative and Recursive searching; Multidimensional arrays; Representation of multidimensional arrays; Sparse matrices.	08

Unit – 3	
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,	08
Unit – 4	
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. **Queues**: Basic Concepts – Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues;	08
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 representation of binary tree. Traversal of binary tree; preorder, inorder and postorder traversal;	08

Text Books

1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures (1982), ISBN: 0-914894-20X.

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))

Course Code: **21BCADSC6**	**Course Title: Database Management System**
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 03 Hours

Course Outcomes (COs):

- Understand the merits and demerits of DBMS and the components of database system.
- Evaluate business information problem and find the requirements of a problem in terms of data.
- Understand, appreciate and effectively explain the concepts of database technologies.
- Design a database using ER modeling and Normalization to given Business information problem.
- Understand issues and techniques relating to concurrency and recovery in multi-user database environments.

Couse Content

Unit	Description	Hours
1	**Databases and Database Users:** Introduction, An Example, Characteristics of the Database Approach, Actors on the Scene, Workers behind the Scene, Advantages of Using the DBMS Approach, A Brief History of Database Applications, When Not to Use a DBMS.	08
2	**Database System Concepts and Architecture:** Data Models, Schemas, and Instances, Three-Schema Architecture and Data Independence, Database Languages and Interfaces, The Database System Environment, Centralized and Client/Server Architectures for DBMSs, Classification of Database Management Systems.	08
3	**Data Modeling Using the Entity–Relationship (ER) Model:** Using High-Level Conceptual Data Models for Database Design, A Sample Database Application, Entity Types, Entity Sets, Attributes, and Keys, Relationship Types, Relationship Sets, Roles, and Structural Constraints, Weak Entity Types, Refining the ER Design for the COMPANY Database, ER Diagrams, Naming Conventions, and Design Issues.	08
4	**Relational Data Model and Relational Database Constraints:** Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update Operations, Transactions, and Dealing with Constraint Violations. **Basic SQL:** SQL Data Definition and Data Types, Specifying Constraints in SQL, Basic Retrieval Queries in SQL, INSERT, DELETE, and UPDATE Statements in SQL.	08

5	**Relational Algebra:** Unary Relational Operations: SELECT and PROJECT, Relational Algebra Operations from Set Theory, Binary Relational Operations: JOIN and DIVISION, Additional Relational Operations. **Basics of Functional Dependencies and Normalization for Relational Databases:** Informal Design Guidelines for Relation Schemas, Functional Dependencies, Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form.	10

Textbook:

1. Fundamentals of Database Systems, Ramez Elamassri, Shankant B. Navathe, 7th Edition, Pearson, 2015.
2. Database Systems Concepts, Abraham Silberschatz, Henry Korth, S.Sudarshan, 6th Edition, McGraw Hill, 2010.

References:

1. An Introduction to Database Systems, Bipin Desai, Galgotia Publications, 2010.
2. Introduction to Database System, C J Date, Pearson, 1999.
3. Database Management Systems, Raghu Rama Krishnan and Johannes Gehrke, 3rd Edition, McGraw Hill, 2002

Course Code: **21BCADSC5P**	**Course Title: Data Structures Lab**
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 42	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 03 Hours

Programming Lab

Part A:

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

Part B:

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

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part B	Writing the Program	04
	Execution and Formatting	04
Program -2 from Part B	Writing the Program	04
	Execution and Formatting	04
Viva Voice based on C Programming		05
Practical Record		04
Total		**25**

Course Code: **21BCADSC6P**	**Course Title:** DBMS Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 42	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 04 Hours

Course Outcomes (COs):

- To understand difference between storing data in FMS and DBMS and advantages of DBMS.
- To Understand conceptual and physical design of a database.
- To understand RDMBS and to design Relational database.
- To know basic database backup and recovery.
- To know basics of advances in DBMS.

Practicals:

1. Execute a single line query and group functions.
2. Execute DDL Commands.
3. Execute DML Commands
4. Execute DCL and TCL Commands.
5. Implement the Nested Queries.
6. Implement Join operations in SQL
7. Create views for a particular table
8. Implement Locks for a particular table.
9. Write a query to understand the concepts for ROLL BACK, COMMIT & CHECK POINTS.
10. Write a query for extracting data from more than one table.

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part B	Writing the Program	04
	Execution and Formatting	04
Program -2 from Part B	Writing the Program	04
	Execution and Formatting	04
Viva Voice based on C Programming		05
Practical Record		04
Total		**25**

Specific Elective Course in Computer Science:

Course Code: **21SEC1**	**Course Title: Digital Fluency**
Course Credits: 02	Hour of Teaching/Week: 03 (1 hr./Teaching and 2 hrs./ Practical)
Total Contact Hours: 45	Formative Assessment Marks: 20
Exam Duration: 01.30 Hrs	Exam Marks: 30

Course Outcomes (COs):

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

- To perform and get knowledge about applications, virtual learning and internet fundamentals.
- Develop holistically by learning essential skills such as effective communication, problem-solving, design thinking, and teamwork.

Course Content:

Content	**Hours**
Unit – 1	
Fundamentals of computer and Operating System: **Fundamentals of computer:** An Overview of Computer, Functional Components of a computer (Working of each unit), Evolution and Generations of Computers, Classification of Computers, Applications of Computers. **Operating System:** Operating Systems, types of operating systems, major functions of the operating systems, types of user interface, examples of operating systems: MS-DOS, Windows, Mac OS, Linux, Solaris, Android.	05
Unit – 2	
Office Automation Tools, ICT and Security Aspects: **Office automation tools and ICT:** word processor, power point, and spread sheet, creating an email-ID, e-mail reading, saving, printing, forwarding and deleting the mails, checking the mails, viewing and running file attachments, addressing with cc and bcc, Google forms, working with Google Meet and Zoom Meet. Introduction to e-learning platforms such as Swayam and MOOC. **Security Aspects:** Threats and Prevention, Malware - virus, Worms, Ransomware, Trojan, Spyware, adware, key loggers, Modes of Malware distribution, Antivirus, HTTP vs HTTPS, Firewall, Cookies, Hackers and Crackers.	05
Unit – 3	
E-Commence, E- payment methods and Societal impacts: **E-Commence:** Basic Web Commerce Concept, The E-Commerce Environment, Electro Marketplace Technologies, B2B, B2C and C2C. **E- payment methods:** Cash Payment System, Credit Payment System, Types of Electronic Payment Systems: Credit Card • Debit Card • Smart Card • E-Money • Electronic Fund Transfer (EFT). **Societal impacts:** Digital Foot prints, Digital Society and Netizen, Data Protection, E-waste, Impact on Health.	05

Laboratory Activities: • Identifying the configuration and version of a computer system (PC), laptop, and a mobile phone. • Create an email-ID and sending and editing with mail merge. • Creating a Google form and send it to Ten users. • Scheduling a virtual meet and invite peoples to join the Google meet and Zoom Meet. • Record the virtual Meet, chatting and sharing the documents or presentation on virtual meet. • Creating a hotspot from a mobile phone, and allowing others to use the hotspot. • Sign in and create account e-learning platforms such as Swayam and MOOC. • Creating a one-minute video of your choice in your native tongue, and upload the video to YouTube. • Creating and composing word document, creating tables, creating charts etc. • Preparing power point slides using transition and animations. • Simple computation using spread sheet. • Creating an account in the railway reservation website, IRCTC, and finding trains from Vijayapura to Bangalore. • Creating an account in Redbus.in and book bus tickets online. • Demo of online order placing for book using online e-payment methods.	30

Note: Faculty of Computer Science shall teach the Digital Fluency paperText

Books:

1. Fundamentals of computers - V. Rajaraman - Prentice- Hall of India.
2. Computer Fundamentals - P. K. Sinha Publisher: BPB Publications.

Reference Links:

- Operating Systems: https://ftms.edu.my/v2/wpcontent/uploads/2019/02/csca0101_ch06.pdf
- Gmail Creating links: https://clubrunner.blob.core.windows.net/00000000961/en-ca/files/homepage/how-to-create-a-gmail-account/HowtoCreateaGmailAccount.pdf
- Google Forms: https://pdst.ie/sites/default/files/Google%20Drive_1.pdf
- Google Meet: https://edvance.hawaii.hawaii.edu/wp-content/uploads/Google-Meet-Tutorial-Getting-Started-and-Recording-a-Lecture.pdf
- Zoom Meet: https://assets.zoom.us/docs/user-guides/zoom-rooms-full-user-guide.pdf
- Swayam: https://www.aicte-india.org/bureaus/swayam
- Security Aspects - https://ncert.nic.in/textbook/pdf/lecs112.pdf
- E-Commence: http://www.aagasc.edu.in/cs/msccs/ECommerce%20Unit%201.pdf
- E- payment methods: http://www.dspmuranchi.ac.in/pdf/Blog/e%20business%20UnitIII,%20%202020.pdf
- Societal impacts: https://ncert.nic.in/textbook/pdf/leip106.pdf

- Forms and reports
- Demo of the Application developed

Note: IA Marks shall be assigned by the concerned guide monitoring the project work of the students

Project Work		
Report (30)	Presentation (30)	Viva-Voce (20)

HOD,
Dept. of Computer Application,
Smt. A.S.M. College for Women,
BELLARY-583 103.

IQAC
CO-ORDINATOR
Smt. Allum Sumangalamma Memorial
College For Women, BELLARY.

PRINCIPAL
Smt. ASM College
For Women, BALLARI