TUMKUR UNIVERSITY

Board of Studies in Computer Science

Curriculum Structure and Syllabus for I and II Semester

Bachelor of Computer Applications

Choice Based Credit System

2024-25 Onwards

BOS Computer Science Committee Members

1.	**Dr. Kusuma Kumari B.M** MCA Coordinator & Assistant Professor Department of Studies and Research in Computer Applications Tumkur University, Tumakuru	Chairperson
2.	**Dr. Ramani. R** Assistant Professor Department of Computer Science University College of Science Tumkur University, Tumakuru	Member
3.	**Dr. Chandrali Baishya** Associate Professor Department of Studies and Research in Mathematics Tumkur University, Tumakuru	Member
4.	**Dr. Prakash B.R** Assistant Professor Department of Computer Science Government First Grade College, Tipatur.	Member
5.	**Sri. Mohan Kumar N** Assistant Professor Department of Computer Science Y.E.R Government First Grade College, Pavagada.	Member
6.	**Capt. Ramalinga Reddy S** Assistant Professor Sri Siddaganga Arts and Commerce Evening College Tumkur.	Member
7	**Dr. Manjunath S** Assistant Professor, Department of Computer Science Y.E.R Government First Grade College, Pavagada.	Member
8	**Dr. Nagamani H.S** Associate Professor, Department of Computer Science Maharani Claster University, Bengaluru.	Member
9	**Dr. Asha Gowda Karegowda** Associate Professor , Department of MCA Siddaganga Institute of Technology, Tumkur.	Member
10	**Dr. Sumathi R Gowda** Assistant Professor MCA Department, Karnataka State Open University, Muktha Gangotri, Mysore.	Member
11.	**Dr. Haridas S.** Associate Professor Department of Computer Science Government First Grade College, Tumkur.	Member

SEMESTER – I

Sl. No	Paper	Title of the Paper	Instruction Hrs. per Week	No. of Credits	Duration of the Exam.	Marks		
						Internal Assessment	Semester End Exam.	Total
1	BCAT 101	Computer Organization and Logic Design	4	4	3 Hrs.	20	80	100
2	BCAP 102	Logic Design Lab	4	2	3 Hrs.	10	40	50
3	BCAT 103	Principles of Programming Languages	4	4	3 Hrs.	20	80	100
4	BCAP 104	MS Office and PPL Lab	4	2	3 Hrs.	10	40	50
5	BCAT 105	Programming in C	4	4	3 Hrs.	20	80	100
6	BCAP 106	Programming in C Lab	4	2	3 Hrs.	10	40	50
TOTAL				**18**				**450**

NOTE:
Theory: 1 Hr. = 1 Credit **Practical: 2 Hrs. = 1 Credit**
BCAT: BCA Core Paper Theory
BCAP: BCA Core Paper Practical

SEMESTER – II

Sl. No	Paper	Title of the Paper	Instruction Hrs. per Week	No. of Credits	Duration of the Exam.	Marks		
						Internal Assessment	Semester End Exam.	Total
1	BCAT 201	Data Structure Using C	4	4	3 Hrs.	20	80	100
2	BCAP 202	Data Structure Using C Lab	4	2	3 Hrs.	10	40	50
3	BCAT 203	Java Programming	4	4	3 Hrs.	20	80	100
4	BCAP 204	Java Programming Lab	4	2	3 Hrs.	10	40	50
5	BCAT 205	Discrete Mathematics	4	4	3 Hrs.	20	80	100
6	BCAP 206	Soft Skill Enhancement Lab	4	2	3 Hrs.	10	40	50
TOTAL				**18**				**450**

NOTE:
Theory: 1 Hr. = 1 Credit **Practical: 2 Hrs. = 1 Credit**
BCAT: BCA Core Paper Theory
BCAP: BCA Core Paper Practical

Internal Assessment Marks for Theory and Practical

Internal Assessment Marks Allotment for Theory	
Internal Test 1	05
Internal Test 2	05
Assignment	05
Seminar/Quiz	05
TOTAL	**20**

Internal Assessment Marks Allotment for Practical	
Internal Test	05
Record and Attendance	05
TOTAL	**10**

Evaluation Scheme for Lab Examination

Assessment Criteria	
Writing 2 Programs	**10**
Execution of 2 Programs	**20**
Viva	**10**
Total	**40 Marks**

First Semester			
Subject Name: Computer Organization and Logic Design			
Subject Code	BCAT 101	CIE Marks	20
No of Hours/Week:	4	SEE Marks	80
Total Hours:	60	Credits	4

COURSE OBJECTIVES:

1. To prepare students understand various number systems
2. To acquire the basic knowledge of digital logic levels and application of knowledge to understand digital electronics circuits.
3. To prepare students to perform the analysis and design of various digital electronic circuits.
4. To acquire basic concepts of computer organization.

COURSE OUTCOMES:

After studying this course, the students would gain enough knowledge on

1. Have a thorough understanding of the fundamental concepts and techniques used in digital electronics.
2. Understand and examine the structure of various number systems and its application in digital design.
3. The ability to understand, analyze and design various combinational and sequential circuits.
4. Ability to identify basic requirements for a design application
5. The ability to understand interfacing techniques of slower devices with faster devices

Unit 1	**Computer Organization**	**Hours 15**
	Introduction to CPU: Block diagram of computer, characteristics. General register organization of Computer (block diagram, explanation) Bus organization: Address bus, Data bus, Control bus, direction of data flow, Internal data operations in CPU: memory read write, IO read & write. Instruction: definition, format & Classification of instructions. Operation code formats and operand, Instruction fetch, Execute cycle, fetch cycle. Addressing modes: definition, types and examples.	
Unit 2	**Number Representation Techniques**	**Hours 12**
	Introduction, Digital and analog representation, comparison. Number system: Base or radix, Binary, Octal, Decimal, Hexadecimal. Inter conversion techniques Signed and un signed representation (positive and negative representation with 8-bit representation). Binary Arithmetic: Addition, 1s and 2 s compliment, subtraction using 1's and 2's compliment. Binary Codes: Weighted codes BCD, 8421, interconversion applications. Non weighted codes: Excess-3, Gray code. Alphanumeric codes: ASCII, EBCDIC, UNICODE	
Unit 3	**Boolean algebra and Logic gates**	**Hours 13**
	Fundamental Boolean operations and basic logic gates. Derived Gates, Universal Gates	

	NAND and NOR Gates, Universal property. Boolean algebra laws, Demorgan's Theorems. Logic Expression: SOP, POS, canonical representation, SOP and POS to Canonical form conversion. Simplification using Boolean algebra laws Simplification of logic expression using KMAP (without Don't care condition)	
Unit 4	**Combinational logic circuits**	**Hours 10**
	Combinational logic circuits: features, examples, applications. Adder: Half-Adder, Full Adder, implementation of full adder using two Half adders. Subtractor: Half-Subtractor, Full subtractor implementation of full subtractor using two Half subtractors. Encoder: definition, Decimal to BCD encoder, Decoder: definition, BCD to Decimal Decoder Multiplexers: Definition, 4:1 MUX truth table logic diagram. Demultiplexers: Definition, 1:4 DMUX truth table logic diagram. Sequential logic Circuits: features, examples, applications. Latch, concept of clock, Flip-flops conversion of RS flip flop to D flip flop and JK flipflop, JK to T flip flop, Application of Flipflops.	
Unit 5	**Interfacing and Memory Organization**	**Hours 10**
	Introduction, IO interfacing, Data transfer schemes: Synchronous, Asynchronous, DMA. Memory hierarchy: Primary memory, Secondary memory. Memory organization: Cache memory, virtual memory, memory management hardware Memory stack. Memory address decoding (3bit address).	

TEXT BOOKS:

1. M. Moris Mano, Computer Systems Architecture, 3rd Edition, Pearson Education, 2007.
2. Digital Fundamentals, Thomas L. Floyd, 11E Pearson Education

REFERENCE BOOKS:

1. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012
2. Digital fundamental by Thomas L. Floyd
3. Digital Electronics by A K Maini
4. Computer Organization and architecture by William Stallings
5. Patterson and Hennessy, Computer Organization and Design, Morgan Kaufmann, ARM Edition, 2011
6. R P Jain, Modern Digital Electronics, Tata McGraw Hill Education Pvt. Ltd. , 4th Edition, 2010

First Semester			
Subject Name: Logic Design Lab			
Subject Code	BCAP 102	CIE Marks	10
No of Hours/Week:	4	SEE Marks	40
Total Hours:	60	Credits	2

LIST OF PRACTICAL PROGRAMS

PART A:

1. Verification of truth tables for AND, OR, NOT gates.
2. Verification of truth tables for EXOR and EXNOR gates
3. Verification of truth tables for NAND and NOR gates.
4. Realization of AND, OR, NOT gates using only NAND gates.
5. Realization of AND, OR, NOT gates using only NOR gates.
6. Verification truth table for Half adder using NAND gates.
7. Verification truth table for Full adder using NAND gates.
8. Verification of truth table for Full subtractor using XOR gates and Basic gates
9. Verification truth table for Half subtractor using NAND gates.
10. Verification of truth table for Decimal to BCD Encoder

PART B:

1. Verification of truth table for BCD to Seven segment display decoder.
2. Verification of Distributive property.
3. Verification of truth table for BINARY TO GRAY conversion using XOR gates
4. Verification of truth table for GRAY TO BINARY conversion using XOR gates
5. Verification of truth table for SR and D Flip-Flop.
6. Verification of truth table for JK and T Flip-Flop.
7. Design of logic circuit for simple 2 variable SOP expression.
8. Design of logic circuit for simple 2 variable POS expression.
9. Design of logic circuit of De-Morgans' Theorem and realize.
10. Simplification of given expression using KMAP and designing the logic circuit. (SOP only)

Practical can be performed using any open-source simulator (like Logisim)
(Download it from https://sourceforge.net/projects/circuit/)

First Semester			
Subject Name: Principles of Programming Languages			
Subject Code	BCAT 103	CIE Marks	20
No of Hours/Week:	4	SEE Marks	80
Total Hours:	60	Credits	4

COURSE OBJECTIVES:

1. For the beginner students learning by using text based is difficult to programming concepts.
2. By using flowcharts, Students can concentrate on programming concepts rather than all the nuances of a typical programming language

COURSE OUTCOMES:

After studying this course, the students able to,

1. To introduce the various programming paradigms.
2. To understand the evolution of programming languages.
3. To understand the concepts of OO languages, functional languages, logical, scripting Languages and modern programming languages.
4. To introduce the notations to describe the syntax and semantics of programming languages.

Unit 1	**Preliminary Concepts**	**Hours 15**
	Reasons for studying concepts of programming languages, programming domains, language evaluation criteria, influences on language design, language categories, language design trade-offs, implementation methods, programming environments. Syntax and Semantics: General problem of describing syntax, formal methods of describing syntax, attribute grammars, describing the meanings of programs, Main Programming Structures, Algorithm, Flow Charts & Pseudocode, documenting a Program, understand programming logic through Flogorithm	
Unit 2	**Programming Language**	**Hours 15**
	Introduction, Key Concepts and overview of: Functional Programming Languages, Logic Programming Language, Scripting Languages. Object-Oriented Programming: Design issues for OOP, Using the Unified Modelling Language to Design Classes, Inheritance, Polymorphism GUI Programming Languages: GUI Applications, Graphical User Interfaces, Designing the User Interface and Event Handler for a GUI Program.	
Unit 3	**Basic concepts of Program**	**Hours 10**
	Variables and scopes: Introduction, names, variables, concept of binding, scope, scope and lifetime, referencing environments, named constants, Operators Data types: Introduction, primitive, character, string types, user defined ordinal types, array, associative arrays, record, tuple types, list types, union types, pointer and reference types, type checking, strong typing, type equivalence.	
Unit 4	**Statements and Control Structures**	**Hours 10**
	Expressions and Statements: Arithmetic expressions, overloaded operators, type conversions, relational and Boolean expressions, short- circuit evaluation, assignment statements, mixed mode assignment. Control Structures – introduction, selection statements, iterative statements,	

	unconditional branching, guarded commands.	
Unit 5	**Sub Programs**	**Hours 10**
	Introduction to Sub-Programs (Functions): Defining and Calling a Module, Local Variables, Passing Arguments to Modules, Global and Local Variables. Calling subprograms indirectly, implementing subprograms, General semantics of calls and returns, implementing subprograms with stack-dynamic local variables, nested subprograms, blocks. Concept of Abstract Data types. Library Functions, Introduction to Recursion and Problem Solving with Recursion. Introduction to Menu-Driven Programs: Using a Loop to Repeat the Menu, Modularizing a Menu-Driven Program, Multiple-Level Menus	

TEXT BOOKS:

1. Concepts of Programming Languages, 11E, Robert Sebesta, Pearson, Global Edition.

REFERENCE BOOKS:

1. Starting out with Programming Logic & Design, 3E, Tony Gaddis, Pearson Education
2. Programming language design concepts, Watt, David A. John Wiley & Sons Ltd.
3. The Art of Programming through Flowcharts & Algorithms, 2E, A. B. Chaudhuri, Firewall Media

First Semester			
Subject Name: MS Office and PPL Lab			
Subject Code	BCAP 104	CIE Marks	10
No of Hours/Week:	4	SEE Marks	40
Total Hours:	60	Credits	2

LIST OF PRACTICAL PROGRAMS

PART A:

1. Design a visiting card for Managing Director of a company as per the following specification.
   - Size of visiting card is 3½×2
   - Name of the company with big font
   - Phone number, Fax number and E-mail address with appropriate symbols.
   - Office and Residence address separated by a line.

2. Create a table with following columns and display the result in separate cells for the following
   - Emp Name, Basic pay, DA, HRA, Total salary.
   - Sort all the employees in ascending order with the name as the key
   - Calculate the total salary of the employee
   - Calculate the Grand total salary of the employee
   - Finding highest salary and
   - Find lowest salary

3. Prepare an advertisement to a company requiring software professional with the following
   - Attractive page border
   - Design the name of the company using WordArt
   - Use at least one clipart.
   - Give details of the company (use bullets etc.)
   - Give details of the Vacancies in each category of employee's (Business manager, Software engineers, System administrators, Programmers, Data entry operators) qualification required.

4. Create two pages of curriculum vitae of a graduate with the following specifications
   - Table to show qualifications with proper headings
   - Appropriate left and right margins
   - Format ½ page using two-column approach about yourself
   - Name on each page at the top right side
   - Page no. in the footer on the right side.

5. Create a letter as the main document and create 10 records for the 10 persons use mail merge to create letter for selected persons among 10.

6. Create a excel spreads sheet to calculate the net pay of the employees following the conditions below.

	A	B	C	D	E	F	G	H	I
1	EMP ID	EMP NAME	BASIC	DA	HRA	GPF	INCOME TAX	GROSS SAL	NET SAL
2									

- DA: -56% of the basic pay if Basic pay is greater than 20000 or else 44%.
- HRA: -15% of the Basic pay subject to maximum of Rs. 4000.
- GPF: -10% of the basic pay.
- INCOMETAX: - 10% of basic if Basic pay is greater than 20000.
- Find who is getting highest salary & who is get lowest salary?

7. The ABC Company shows the sales of different product for 5 years. Create BAR Graph, 3D and Pie chart for the following.

A	B	C	D	E	F
SL.NO.	YEAR	PROD1	PROD2	PROD3	PROD4
1	2019	1000	800	90	1000
2	2020	800	90	70	1100
3	2021	1200	190	100	900
4	2022	600	260	58	1400
5	2023	1800	510	80	800

8. Create a suitable examination data excel sheet and find the sum of the marks (total) of each student and respective, class secured by the student.

- Pass: if marks in each subject>=35
- Distinction: if average>=75
- First class: If average>=60 but <75
- Second Class: if average >=50 but less than 60
- Third class: if average>=35 but less than 50
- Fail: if marks in any subject<35

9. Enter the following data into the sheet.

NAME	DEPARTMENT	SALARY
Anusha	Accounts	30000
Ramesh	Marketing	20000
Tejaswi	Engineering	44000
Harika	Accounts	35000
Poornima	Engineering	56000
Vijay	Marketing	22000
Prasad	Accounts	48000
Swetha	Engineering	60000
Rajesh	Marketing	19000

- Extract records for department Accounts and Salary > 10000
- Sort the data by salary with the department using “sort commands”.
- Calculate total salary for each department using Subtotals

10. Create a presentation with slide transitions and animation effects.

PART B:

Design Flowchart and submit to generate pseudo code and run the program for the following problems (Any 12):

1. Read radius to calculate and display area of circle.
2. Find maximum of three number
3. Count no of digits in a number.
4. Reverse a given number.
5. Print odd number from given numbers.
6. Find minimum, maximum and average score of labs.
7. Print even number between 2 to n
8. Determine the give year is leap or not
9. Finding Duplicates in an Array.
10. Defining and calling a function.
11. To perform bubble sort.
12. Print triangle pattern
13. To search an element using linear search.
14. Solve quadratic equation.
15. Reverse a string.

First Semester			
Subject Name: Programming in C			
Subject Code	BCAT 105	CIE Marks	20
No of Hours/Week:	4	SEE Marks	80
Total Hours:	60	Credits	4

COURSE OBJECTIVES:

1. The course aims to provide exposure to problem-solving through programming.
2. It aims to train the student to the basic concepts of the C-programming language.
3. This course involves a lab component which is designed to give the student hands-on experience with the concepts.

COURSE OUTCOMES:

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

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

Unit 1	**C Programming Basic**	**Hours 10**
	Introduction and Overview: 'C' History and Background, Example 'C' Program, C Program Structure. C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants. Input and output with C: Formatted I/O functions - printf and scanf, control stings and escape sequences, output specifications with printf functions; Unformatted I/O functions - getchar, putchar, gets and puts functions.	
Unit 2	**Operators & Expressions**	**Hours 15**
	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; Looping Statements - Entry controlled and exit controlled statements, while, do-while, for loops, Nested loops.	

Unit 3	**Derived data and Strings**	**Hours 12**
	Derived data types in C: Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays - Declaration, Initialization and Memory representation. Strings: Declaring & Initializing string variables; String handling functions - strlen, strcmp, strrev, strcpy and strcat; Character handling functions - toascii, toupper, tolower, isalpha, isnumeric etc.	
Unit 4	**Functions and Structures**	**Hours 13**
	User Defined Functions: Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type. User defined data types: Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure member's initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.	
Unit 5	**Pointers**	**Hours 10**
	Pointers in C: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Ponter and function; Advantages and disadvantages of using pointers.	

TEXT BOOKS:

1. E. Balaguruswamy: Programming in ANSI C (TMH)
2. C: The Complete Reference, By Herbert Schildt.
3. C Programming Language, By Brain W. Kernighan
4. Kernighan & Ritchie: The C Programming Language (PHI)

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)

First Semester			
Subject Name: Programming in C Lab			
Subject Code	BCAP 106	**CIE Marks**	10
No of Hours/Week:	4	**SEE Marks**	40
Total Hours:	60	**Credits**	2

LIST OF PRACTICAL PROGRAMS

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

1. Basic Computer Proficiency
   a. Familiarization of Computer Hardware Parts
   b. Basic Computer Operations and Maintenance.
   c. Do's and Don'ts, Safety Guidelines in Computer Lab

2. Familiarization of Basic Software – Operating System, Basic DOS Commands, Word Processors, Spread Sheets, PowerPoint, Internet Browsers, Integrated Development Environment (IDE) with Examples.

3. Type Program Code preferably by using DevC++ or Codeblock or VisualCode, Debug and Compile basic programs covering C Programming fundamentals discussed during theory classes.

PART A:

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

PART B:

1. Write a C Program to find the length of a string without using built in function
2. Write a C Program to demonstrate string functions.

3. Write a C Program to demonstrate pointers in C
4. Write a C Program to check a number for prime by defining *isprime( )* function
5. Write a C Program to read, display and to find the trace of a square matrix
6. Write a C Program to read, display and add two m x n matrices using functions
7. Write a C Program to read, display and multiply two m x n matrices using functions
8. Write a C Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
9. Write a C Program to Reverse a String using Pointer
10. Write a C Program to Swap Two Numbers using Pointers
11. Write a C Program to demonstrate student structure to read & display records of n students.
12. Write a C Program to demonstrate the difference between structure & union.

Second Semester			
Subject Name: Data Structure Using C			
Subject Code	BCAT 201	CIE Marks	20
No of Hours/Week:	4	SEE Marks	80
Total Hours:	60	Credits	4

COURSE OBJECTIVES:

1. To explore and understand the concepts of Data Structures and its significance in programming.
2. To teach efficient storage mechanisms of data for an easy access
3. Provide and holistic approach to design, use and implement abstract data types.
4. Understand the commonly used data structures and various forms of its implementation for different applications using C
5. To improve the logical ability.

COURSE OUTCOMES:

1. Learn about Data structures, its types and significance in computing.
2. Student will be able to handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
3. Explore about Abstract Data types and its implementation
4. Ability to program various applications using different data structure in C

	Introduction to Data Structure	**Hours 10**
Unit 1	**Data structure** – Definition, Abstract data type, Classification of Data Structures, Operations on Data Structures-primitive and non-primitive. **String Processing**: Definition, Storing Stings, String as ADT, String operations – string length, copy, compare, pattern matching using user defined function and pointer. **Memory allocation**: Definition, types-static and dynamic, difference between static and dynamic. Memory management functions- malloc, calloc, realloc and free.	
	Arrays	**Hours 10**
Unit 2	**Arrays:** Definition, Linear arrays, memory representation of Linear Array, Accessing the Elements of an Array, Calculating the Address of Array Elements. Array operations - Inserting, deleting and merging. **Sorting**: Bubble sort, Insertion sort, Selection sort. **Searching**: Linear Search and Binary search. **Multidimensional array**- Declaration and memory representation, Sparse matrices.	
	Linked Lists	**Hours 15**
Unit 3	**Linked list:** Definition, Types. Memory representation of Singly linked list, operations- create, display, insert and delete. Doubly liked list- definition and memory representation. Circular linked list-definition and memory representation. Advantages and disadvantages of SLL, DLL and CLL. Garbage collection.	

Unit 4	**Stacks and Queues**	**Hours 15**
	Stacks: Definition, Array representation of stacks, Linked representation of stacks, Operations on stack. Expressions: Polish Notation, Application of Stacks. Algorithm – Convert infix to postfix expression and evaluate postfix expression. **Queues** – Definition, Array representation of queue, Linked list representation of queues. Types of queues: Simple queue, Circular queue, Double ended queue, Priority queue, Operations on Queues, Applications of queues.	
Unit 5	**Graphs and Trees**	**Hours 10**
	Graphs: Graph theory terminology: pendent vertex, isolated vertex, degree of vertex, path, walk, self-loop, cycle, Sequential representation of Graphs: Adjacency matrix. **Tree** – Definitions, Types of trees, memory representation of binary tree, tree traversing techniques, Binary Search Trees- definition and construction of binary search tree. AVL Tree- definition. Threaded Binary tree – Definition.	

TEXT BOOK:

1. Data Structure using C by Reema Thereja, Oxford University Press.

REFERENCE BOOKS:

1. Data Structure by S. Lipschutz - (Schaum Series)
2. Introduction to Data Structure in C by: A.N.Kamthane; Pearson Education
3. Alfred V Aho, John E Hopcroft and Jeffery D Ullman, “Data Structures and Algorithms”, Pearson Education.

Second Semester			
Subject Name: Data Structure Using C Lab			
Subject Code	BCAP 202	CIE Marks	10
No of Hours/Week:	4	SEE Marks	40
Total Hours:	60	Credits	2

LIST OF PRACTICAL PROGRAMS

PART A:

1. Develop a Program in C for the operations on Strings like finding the string of length, copying two strings, comparing of two string and pattern matching & replacing. Support the program with functions for each of the above operations. Don't use Built-in functions
2. Write a C program to read name and roll number of n number of students from user and store them in a file.
3. Write a C Program to implement dynamic array, find smallest and largest element of the array.
4. Write a C Program read, display and to find the trace of a square matrix
5. Write a C Program to read, display and add two m x n matrices using functions
6. Write a C Program to read, display and multiply two m x n matrices using functions
7. Write a C Program to read the names of cities and arrange them alphabetically.
8. Write a C Program to search an element using linear search technique.
9. Write a C Program to sort the given list using selection sort technique.
10. Write a program to implement merge sort.

PART B:

1. Program to implement linear linked list to perform insert and delete operations on it.
2. Write a C Program to implement Stack and its different operations.
3. Write a C Program to convert an infix expression to postfix.
4. Write a C Program to evaluate a postfix infix expression.
5. Write a C Program to implement simple queue and its different operations.
6. Write a program to implement circular queue using array.
7. Program to create and display different traversal of a binary tree.
8. Write a program to implement BFS.
9. Write a program to implement DFS.
10. Write a program to implement AVL Tree

Second Semester			
Subject Name: Java Programming			
Subject Code	BCAT 203	CIE Marks	20
No of Hours/Week:	4	SEE Marks	80
Total Hours:	60	Credits	4

COURSE OBJECTIVES:

1. To introduce the object-oriented programming concepts.
2. To understand object-oriented programming concepts, and apply them in solving problems.
3. To introduce the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes
4. To introduce the implementation of packages and interfaces
5. To introduce the concepts of exception handling and multithreading.
6. To introduce the design of Graphical User Interface using applets and swing controls.

COURSE OUTCOMES:

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

1. Understand the features of Java and the architecture of JVM
2. Write, compile, and execute Java programs that may include basic data types and control flow constructs and how type casting is done
3. Identify classes, objects, members of a class and relationships among them needed for a specific problem and demonstrate the concepts of polymorphism and inheritance
4. The students will be able to demonstrate programs based on interfaces and threads and explain the benefits of JAVA's Exceptional handling mechanism compared to other Programming Language
5. Write, compile, execute Java programs that include GUIs and event driven programming and also programs based on files

Unit 1	**Introduction to Java**	**Hours 10**
	History of java, Features of java, JDK Environment & tools like (java, javac, appletviewer, javadoc, jdb) OOPs Concepts Class, Abstraction, Encapsulation, Inheritance, Polymorphism, Difference between C++ and JAVA, Structure of java program, Data types, Variables, Operators, Keywords, Naming Convention	
Unit 2	**Objects and Classes**	**Hours 10**
	Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character, String Buffer, File, this reference, garbage collection, Function Prototyping, Function Overloading	
Unit 3	**Inheritance and Polymorphism**	**Hours 10**
	Inheritance in java, Super and sub class, Overriding, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java. Exception handling: Exception handling in Java.	

Unit 4	**Packages and Interfaces**	**Hours 15**
	Packages, Access Protection, Importing Packages. Interfaces. Multi-Threaded Programming: What are threads? How to make the classes threadable; Extending threads; Implementing runnable; Synchronization; Changing state of the thread; Bounded buffer problems, producer consumer problems.	
Unit 5	**Applet & IO**	**Hours 15**
	Introduction, Applet Life cycle - Creating applet - Applet tag; Applet Classes - Color - Graphics - Font; AWT -Components and container used in AWT; Layout managers; Listeners and Adapter classes; Event Delegation model; Swing - Introduction to Swing Component and Container Classes. I/O programming: Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files.	

TEXT BOOKS:

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

REFERENCE BOOKS:

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

Second Semester			
Subject Name: Java Programming Lab			
Subject Code	BCAP 204	CIE Marks	10
No of Hours/Week:	4	SEE Marks	40
Total Hours:	60	Credits	2

LIST OF PRACTICAL PROGRAMS

PART A:

1. Write a java program to illustrate the creation of variables of basic types and effect of type Conversions
2. Write a java program that displays roots of a quadratic equation ax2+bx+c=0 , calculate the discriminate D and based on the value of D, Describe the nature of root(use nested if statement).
3. Write a Java program for sorting a given list of names in ascending order.
4. Write a Java program to multiply two given matrices.
5. Write a java program to demonstrate creation and accessing of objects and methods.
6. Write a java program to illustrate use of constructors overloading and method overloading.
7. Write a Java program to demonstrate the concepts of Single inheritance
8. Write a Java program to demonstrate the concepts of Multilevel inheritance
9. Write a Java Program to illustrate the implementation of concepts of multiple inheritance using interface.
10. Write a Java Program to illustrate use of most commonly used wrapper class methods

PART B:

1. Write a Java Program to demonstrate String Methods used for manipulating strings like accessing, inserting, modifying and appending.
2. Write a Java Program using Synchronized Threads which demonstrate Producer-Consumer Concept.
3. Write a Java Program for creation of Java Built-in Exceptions.
4. Write a Java Program for creation of User Defined Exceptions.
5. Program which creates and displays a message on the window
6. Program to draw several shapes in the created window
7. Create a simple applet which reveals the personal information of yours.
8. Create a frame which displays your personal details with respect to a button click
9. Develop an Applet that receives an integer in one text field & compute its factorial value & returns it in another text filed when the button “Compute” is clicked.
10. Program to create a window when we press “M or m” the window displays Good Morning, “A or a” the window displays Good After Noon “E or e” the window displays Good Evening, “N or n” the window displays Good Night

Second Semester			
Subject Name: Discrete Mathematics			
Subject Code	BCAT 205	CIE Marks	20
No of Hours/Week:	4	SEE Marks	80
Total Hours:	60	Credits	04

COURSE OBJECTIVES:

1. Provide a survey of Discrete Mathematics, the study of finite systems, needed in computer science.
2. Further develop the mathematical concepts and technique which should serve as a preparation for more advanced quantitative courses.

COURSE OUTCOMES:

On successful completion of this course; the student will be able to:

1. Verify the correctness of an argument using propositional and predicate logic and truth tables.
2. Solve problems on Sets, Relations and functions.
3. Construct the proofs by using proof by mathematical induction, principles of counting and application of permutation and combination
4. Demonstrate the ability to solve problems using operations of matrices and determinants.

Unit 1	**Mathematics Logic**	**Hours 15**
	Definition of proposition, proposition variables, negation of statements, logical connectives and truth tables, predicates and quantifiers, Conditional statement, contra-positive and converse, bi-conditional, tautology, contradiction, logical equivalence, properties of proposition operation-commutative, associative, distributive, idempotent negation. introduction to proofs. methods of proof.	
Unit 2	**Set Theory**	**Hours 10**
	Definition of a set, sub-set with examples, Venn diagrams, types of sets-equal sets, null set, disjoint sets, finite set, infinite set, power set, cardinality of set. Operations on sets. Definition of a relation with examples, types of relations. Definition of a function with examples, types of function.	
Unit 3	**Counting**	**Hours 10**
	Principle of mathematical induction, problems on principle of mathematical induction. Fundamental principle of counting (statement with examples only), permutations-definition and problems. Combinations - definition and problems. Pigeon hole principle- statement and proof, extended pigeonhole principle- statement and proof, applications of Pigeon hole principle	
Unit 4	**Matrices and Determinants**	**Hours 15**
	Definition of matrix and order of matrix, types of matrices, operations on matrices, inverse of a matrix, determinant of matrix, properties of determinant, applications of	

	determinants and matrices. Finding rank of a matrix, normal form, echelon form, Cayley Hamilton theorem, Eigen values and Eigen vectors.	
Unit 5	**Graphs and Trees**	**Hours 10**
	Definitions and Basic Properties of Graphs, Euler paths and circuits, Hamiltonian paths and circuits, Planar graphs, coloring graphs, Isomorphism of Graphs. **Trees:** Trees, rooted trees and path length in rooted trees, Spanning tree and Minimal Spanning tree, Isomorphism of trees, Weighted trees and Prefix Codes.	

TEXT BOOKS:

1. Ralph P. Grimaldi, Discrete and Combinatorial Mathematics, 5thEdition, Pearson Education, 2004.
2. C. L. Liu and D. P. Mohapatra, Elements of Discrete Mathematics: A Computer Oriented Approach, McGraw Hill, Third Edition, 2012.
3. C. L. Liu, Elements of Discrete Mathematics, Tata McGraw-Hill, 2000.
4. B. S. Grewal, Higher Engineering Mathematics, Khanna Publishers.

REFERENCE BOOKS:

1. Kenneth H Rosen: Discrete Mathematics and its Applications, McGraw Hill publications, 7th edition, 2007.
2. J. P. Tremblay and R.P. Manohar: Discrete Mathematical Structures with applications to Computer Science, Mc Graw Hill Ed. Inc. 2002.
3. Krishnamurthy, "Combinatorics Theory & Application", East-West Press Pvt. Ltd., New Delhi.

Second Semester			
Subject Name: Soft Skill Enhancement Lab			
Subject Code	BCAP 206	CIE Marks	10
No of Hours/Week:	4	SEE Marks	40
Total Hours:	60	Credits	2

LIST OF PRACTICAL PROGRAMS

PART A:

1. Basics of Communication Skills and Listening comprehension.
2. Group Discussion – dynamics of group discussion, Lateral thinking, Brainstorming.
3. Personality Development: Decision-Making, Problem Solving, Goal Setting, Time Management & Positive Thinking.
4. Writing Skills: Letter writing, Essays for Competitive examinations.
5. Resume writing – structure and presentation, planning, defining the career objective.
6. Body Language-To reveals your inner self and personality.
7. Meetings- making meeting effective, chairing a meeting, decision-making, seeking opinions, interrupting and handling interruptions, clarifications, closure, negotiation skills.
8. Interview Skills – concept and process, pre-interview planning, opening strategies, answering strategies, interview through tele and video-conferencing.

PART B:

1. Mock interview.
2. Official Mail id Creation
3. Micro Presentation
4. NAD registration Step by Step
5. Use of word processing, presentation tools for communication and presentation skills.
6. Use of Google forms, drive for collaborative work.
7. Use of spreadsheet for data interpretation and data analysis.
8. Using netiquettes in online mode of communication using Zoom / Google Meet / MS-Teams etc.

TEXT BOOKS:

1. Personality Development and SOFT SKILLS, BARUN K. MITRA Oxford University Press.
2. M. Ashraf Rizvi, “Effective Technical Communication”, Tata McGraw-Hill Publishing Company Ltd. 2005.

REFERENCE BOOKS:

1. Andrea J. Rutherford, “Basic Communication Skills for Technology”, 2nd Edition, Pearson Education, 2007.
2. Meenakshi Raman & Sangeeta Sharma, “Technical Communication”, Oxford University Press, 2011. DELTA’s key to the Next Generation TOEFL Test: “Advanced Skill Practice,” New Age International (P) Ltd., Publishers, New Delhi.

Question Paper Pattern for Semester End Examination (SEE)
(Common for I and II Semester)
SUBJECT NAME

Time: 3 Hours **Max. Marks: 80**

Instruction to Candidate: Answer all the Sections

SECTION A

I. Answer any ten of the following questions (10X2 = 20)

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.

SECTION B

II. Answer any five of the following questions (5X5 = 25)

13.
14.
15.
16.
17.
18.
19.

SECTION C

III. Answer any five of the following questions (5X7 = 35)

20.
21.
22.
23.
24.
25.
26.