MANGALORE UNIVERSITY

State Education Policy – 2024
[SEP-2024]

CURRICULUM STRUCTURE

FOR

BCA

BACHELOR OF COMPUTER APPLICATIONS

Semester II								
Sl. No	**Course Code**	**Title of the Cours*e***	**Category of Courses**	**Teaching Hours per Week**	**SEE**	**IA**	**Total Marks**	**Credits**
1		**Language-I**	**Lang**	**4**	**80**	**20**	**100**	**3**
2		**Language-II**	**Lang**	**4**	**80**	**20**	**100**	**3**
3	**BCA –2.1**	Data Structures	**Core**	**4**	**80**	**20**	**100**	**3**
4	**BCA –2.2**	Object Oriented Programming using Java	**Core**	**4**	**80**	**20**	**100**	**3**
5	**BCA –2.3**	Computational Mathematics	**Core**	**5**	**80**	**20**	**100**	**5**
6	**BCA –2.4**	Data Structures Lab	**practical**	**4**	**40**	**10**	**50**	**2**
7	**BCA –2.5**	Object Oriented Programming Lab	**practical**	**4**	**40**	**10**	**50**	**2**
8		**Constitution/Values**	**Compulsory**	**2**	**40**	**10**	**50**	**2**
Sub - Total				**31**	**520**	**130**	**650**	**23**

SEMESTER- II

Program Name	**BCA**	Semester	**II**
Course Title	**Data Structures(Theory)**		
Course Code:	**BCA-2.1**	No.of Credits	**03**
Contact hours	**4 Hours per week**	Duration of SEA/Exam	**3 Hours**
Formative Assessment Marks	**20**	Summative Assessment Marks	**80**

Course Outcomes (COs):

After the successful completion of the course, the 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, 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

Unit	Description	Hours
1	**Introduction to data structures:** Introduction, Basic terminology; Elementary Data Organization, Data Structures, Data Structure Operations **Introduction to Algorithms, Preliminaries:** Introduction, Algorithmic notations, Control structure. **Recursion:** Definition; Recursion Technique Examples – Factorial, Fibonacci sequence, Towers of Hanoi.	**13**

	Arrays: Basic Concepts – Definition, Declaration, Initialisation, Operations on arrays, Types of arrays, Representation of Linear Arrays in memory, Traversing linear arrays, Inserting and deleting elements, Multidimensional arrays- Two Dimensional Arrays Representation of two-dimensional arrays, Sparse matrices. **Sorting:** Selection sort, Bubble sort, Quick sort, Insertion sort, Merge sort	
2	**Searching:** Definition, Sequential Search, Binary search **Dynamic memory management:** Memory allocation and de-allocation functions - malloc, calloc, realloc and free. **Linked list:** Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly liked list, Header linked list, Circular linked list, Representation of Linked list in Memory; Operations on Singly linked lists– Traversing, Searching, Insertion, Deletion, Memory allocation, Garbage collection	**13**
3	**Stacks:** Basic Concepts –Definition and Representation of stacks- Array representation of stacks, Linked 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- Array representation of Queues, Linked representation of Queues, Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues, Operations on queues	**13**
4	**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,; Array representation of binary tree, Traversal of binary tree- preorder, inorder and postorder traversal **Graphs:** Terminologies, Matrix representation of graphs; Traversal: Breadth First Search and Depth first search.	**13**

Text Books:

1. Seymour Lipschutz, Data Structures with C, Schaum's Outlines Series, Tata McGraw Hill, 2011
2. R. Venkatesan and S. Lovelyn Rose, Data Structures, First Edition: 2015, Wiley India Pvt. Ltd. Publications

Reference Books:

1. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures, Computer Science Press, 1982.
2. Aaron M. Tenenbaum , Data structures using C, First Edition, Pearson Education
3. Kamathane, Introduction to Data structures, Pearson Education , 2004
4. Y. Kanitkar, Data Structures Using C, Third Edition, BPB
5. Padma Reddy: Data Structure Using C, Revised Edition 2003, Sai Ram Publications.
6. Sudipa Mukherjee, Data Structures using C – 1000 Problems and Solutions, McGraw Hill Education, 2007

Pedagogy: Lecture/ PPT/ Videos/ Animations/ Role Plays/ Think-Pair-Share/ Predict-Observe- Explain/ Demonstration/ Concept mapping/ Case Studies examples/ Tutorial/ Activity/ Flipped Classroom/ Jigsaw/ Field based Learning/ Project Based Learning/ Mini Projects/ Hobby Projects/ Forum Theatre/ Dance/ Problem Based Learning/ Game Based Learning/ Group Discussion/ Collaborative Learning/ Experiential Learning / Self Directed Learning etc.

Program Name	**BCA**	Semester	**II**
Course Title	**Object Oriented Programming using Java(Theory)**		
Course Code:	**BCA-2.2**	No.of Credits	**03**
Contact hours	**4 Hours per week**	Duration of SEA/Exam	**3 Hours**
Formative Assessment Marks	**20**	Summative Assessment Marks	**80**

Course Outcomes (COs):

After the successful completion of the course, the student will be able to:

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

Unit	Description	Hours
1	**Fundamentals of Object Oriented Programming**: Introduction, Object Oriented Paradigm, Basic Concepts of OOP, Benefits and Applications of OOP. **Introduction to Java:** Java Features, Java Environment, Simple Java Program, Java Program Structure, Java Tokens, Java Statements, Java Virtual Machine. **Java Programming Basics**: Constants, Variables, Data Types, Declaration of variables, Giving values to the variable, Scope of variables, Symbolic constants, Type casting. **Operators and Expressions**: Arithmetic Operators,	**13**

	Relational Operators, Logical Operators, Assignment Operator, Increment and Decrement Operators, Conditional Operator, Special Operators, Mathematical functions. **Using I/O**: Byte streams and character streams, predefined streams, reading console input, reading characters, strings, writing console output. **Decision Making & Branching**: Simple if statement, if..else statement, nesting of if..else statement, the else..if ladder, the Switch statement..	
2	**Decision making & Looping** -The while statement, the do statement, the for statement . Jumps in loops, Labelled loops. **Class & Objects** - Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The 'this' keyword, Overloading Methods, Using Objects as Parameters, Returning Objects, Recursion, Understanding 'static', Introducing 'final ', Using Command-Line Arguments, Varargs : Variable-Length Arguments **Arrays and Strings**: One dimensional arrays, Creating an arrays, Two dimensional arrays , Strings, Vectors, Wrapper classes.	**13**
3	**Inheritance** - Inheritance Basics, Using 'super', Creating Multilevel hierarchy, Method Overriding, Using Abstract Classes, Using final with Inheritance. **Packages & Interfaces** - Packages, Access protection in packages, Importing Packages, Interfaces. **Exception Handling** - Exception Handling Fundamentals – Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch clauses, finally, Java's builtin Exceptions	**13**
4	**Multithreaded Programming**- Introduction, Creating threads, Extending the thread class, stopping & blocking thread, Life cycle of a thread, Using thread methods, Implementing the runnable interface. **Event and GUI programming**: The Applet Class, Types of Applets, Applet Basics, Applet Architecture, An Applet Skeleton, Simple Applet Display Methods, Requesting Repaint, The HTML APPLET tag. Event Handling - The delegation event model, Event Classes ActionEvent, KeyEvent & MouseEvent Classes, Event Listener Interfaces – ActionListener, KeyListener & MouseListener interfaces.	**13**

	Using the Delegation Event Model. Window Fundamentals, Working with Frame Windows, Creating a Frame Window in an Applet. Creating a Windowed Program, Displaying information within a window. **Introducing swing** – two key swing features, components and containers, the swing packages, a simple swing application, event handling. Exploring Swing- Jlabel, JTextField, JButton, Checkboxes , 13 Radio buttons , Jlist , JComboBox.	

Text Books:

1. E Balagurusamy, Programming with Java – A Primer, Fourth Edition, Tata McGraw Hill Education Private Limited.
2. Herbert Schildt, Java : The Complete Reference, Seventh Edition, McGraw Hill Publication.

Reference Books:

1. Herbert Schildt, Java 2-The Complete Reference,Fifth Edition, McGraw Hill publication.
2. CayS. Horstmann, Core Java VolumeI–Fundamentals, Prentice Hall.
3. Somashekara, M.T., Guru, D.S., Manjunatha, K.S, Object Oriented Programming with Java, EEE Edition, PHI.

Pedagogy: Lecture/ PPT/ Videos/ Animations/ Role Plays/ Think-Pair-Share/ Predict-Observe- Explain/ Demonstration/ Concept mapping/ Case Studies examples/ Tutorial/ Activity/ Flipped Classroom/ Jigsaw/ Field based Learning/ Project Based Learning/ Mini Projects/ Hobby Projects/ Forum Theatre/ Dance/ Problem Based Learning/ Game Based Learning/ Group Discussion/ Collaborative Learning/ Experiential Learning / Self Directed Learning etc.

Program Name	**BCA**	Semester	**II**
Course Title	**Computational Mathematics (Theory)**		
Course Code:	**BCA-2.3**	No.of Credits	**05**
Contact hours	**5 Hours per week**	Duration of SEA/Exam	**3 Hours**
Formative Assessment Marks	**20**	Summative Assessment Marks	**80**

Course Outcomes (COs):

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

- Obtain an intuitive and working understanding of numerical methods for the basic problems of numerical analysis.
- Gain experience in the implementation of numerical methods using a computer.
- Trace error in these methods and need to analyse and predict it.
- Provide knowledge of various significant and fundamental concepts to inculcate in the students an adequate understanding of the application of Statistical Methods.
- Demonstrate the concepts of numerical methods used for different applications

Unit	Description	Hours
1	**Computer Arithmetic:** Number System**,** Number representation, Floating point Arithmetic. **Errors in numerical computation** - Errors and their computation **Solution of Algebraic and Transcendental equations**: Introduction, the Bisection method, the method of False position, the Iterative method, Newton-Raphson method, Ramanujan's method. **Interpolation:** Introduction Finite differences- forward differences, backward differences, Central differences, Newton's formula for interpolation, Languages interpolation formula. **Divided differences**- Newton's general interpolation formula	**15**

2	**Least Squares** - Introduction, least squares curve fitting procedures - fitting a straight line, non-linear curve fitting, curve fitting by a sum of exponentials **Numerical differentiation and integration** - Numerical differentiation, Integration- Trapezoidal rule, Simpson's 1/3 rule and Simpson's 3/8 rule.	**15**
3	**Matrices and linear system of equations**: Basic definitions, matrix operations, transpose of a matrix, the inverse of a matrix, matrix norms. **Solution of linear system: Direct methods**- Matrix inversion method, Gaussian elimination method, Gauss-Jordan method, LU decomposition. **Solution of linear systems- Iterative methods**- Gauss-Seidal methods, Jacobi's method.	**15**
4	**Numerical solution of ordinary differential equations**: Solution by Taylor's series, Euler's method, Modified Euler's method, Runge-Kutta methods, Predictor-corrector methods - Adams-Moulton method, Milne's method, Boundary value problems- Finite difference method.	**15**

Text Book:

1. S.S. Sastry, Numerical Analysis, 3rd edition, PHI publication.

Reference Books:

1. M. K. Jain, S.R.K. Iyengar & R. K. Jain, Numerical methods for Scientific and Engineering computation, 5th edition, New Age International publishers.
2. V Rajaraman, Computer Oriented Numerical Methods, 3rd Edition, PHI, 2006

Pedagogy: Lecture/ PPT/ Videos/ Animations/ Role Plays/ Think-Pair-Share/ Predict-Observe- Explain/ Demonstration/ Concept mapping/ Case Studies examples/ Tutorial/ Activity/ Flipped Classroom/ Jigsaw/ Field based Learning/ Project Based Learning/ Mini Projects/ Hobby Projects/ Forum Theatre/ Dance/ Problem Based Learning/ Game Based Learning/ Group Discussion/ Collaborative Learning/ Experiential Learning / Self Directed Learning etc.

Program Name	**BCA**	Semester	**II**
Course Title	**Data Structures Lab**		
Course Code:	**BCA-2.4**	No.of Credits	**02**
Contact hours	**4 Hours per week**	Duration of SEA/Exam	**3 Hours**
Formative Assessment Marks	**10**	Summative Assessment Marks	**40**

PART-A

1. Program to sort the given list using selection sort technique.
2. Program to sort the given list using insertion sort technique
3. Program to solve Tower of Hanoi using Recursion
4. Program to reverse String using Stack
5. Program to search an element using recursive binary search technique.
6. Program to implement Stack operations using arrays.
7. Program to implement Queue operations using arrays.
8. Program to implement dynamic array. Find smallest and largest element.

PART-B

1. Program to sort the given list using merge sort technique.
2. Program to implement circular queue using array.
3. Program to sort the given list using quick sort technique.
4. Program to implement Stack operations using linked list.
5. Program to implement Queue operations using linked list.
6. Program to evaluate postfix expression.
7. Program to perform insert node at the end, delete a given node and display contents of single linked list.
8. Menu driven program for the following operations on Binary Search Tree(BST) of Integers
   (a) Create a BST of N Integers
   (b) Traverse the BST in Inorder, Preorder and Post Order.

Evaluation Scheme for Lab Examination:

Assessment Criteria		
Program-1	**PART-A** **Writing:7 Marks Execution: 8Marks**	**15 Marks**
Program-2	**PART-B** **Writing:10 Marks Execution:10Marks**	**20 Marks**
Practical Record		**05 Marks**
Total		**40Marks**

Program Name	**BCA**	Semester	**II**
Course Title	**Object Oriented Programming Lab**		
Course Code:	**BCA-2.5**	No.of Credits	**02**
Contact hours	**4 Hours per week**	Duration of SEA/Exam	**3 Hours**
Formative Assessment Marks	**10**	Summative Assessment Marks	**40**

PART-A

1. Program to accept student name and marks in three subjects. Find the total marks, average and grade (depending on the average marks).
2. Program, which reads two numbers having same number of digits. The program outputs the sum of product of corresponding digits.(Hint Input 327 and 539 output 3x5+2x3+7x9=84)
3. Program to input Start and End limits and print all Fibonacci numbers between the ranges.(Use for loop)
4. Define a class named Pay with data members String name, double salary, double da, double hra, double pf, double grossSal, double netSal and methods: Pay(String n, double s) - Parameterized constructor to initialize the data members, void calculate() - to calculate the following salary components, and void display() - to display the employee name, salary and all salary components.

   Dearness Allowance = 15% of salary
   House Rent Allowance = 10% of salary
   Provident Fund = 12% of salary
   Gross Salary = Salary + Dearness Allowance + House Rent Allowance
   Net Salary = Gross Salary - Provident Fund

   Write a main method to create object of the class and call the methods to compute and display the salary details. [class basics]

5. Program to create a class DISTANCE with the data members feet and inches. Use a constructor to read the data and a member function Sum ( ) to add two distances by

using objects as method arguments and show the result. (Input and output of inches should be less than 12.).

6. Program to create a class “Matrix” that would contain integer values having varied numbers of columns for each row. Print row-wise sum.
7. Program to extract portion of character string and print extracted string. Assume that ‘n’ characters extracted starting from mth character position.
8. Program to add, remove and display elements of a Vector.

PART-B

1. Create a class named 'Member' having data members: Name, Age, PhoneNumber, Place and Salary. It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherit the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same. [inheritance]

2. Program to implement the following class hierarchy: Student: id, name
   StudentExam (derived from Student): Marks of 3subjects, total marks
   StudentResult (derived from StudentExam) : percentage, grade
   Define appropriate methods to accept and calculate grade based on existing criteria and display details of N students

3. Write a Program to calculate marks of a student using multiple inheritance implemented through interface. Class Student with data members rollNo, name, Stringcls and methods to set and put data.

   Create another class test extended by class Student with data members mark1, mark2, mark3 and methods to set and put data.
   Create interface sports with members sportsWt = 5 and putWt().
   Now let the class results extends class test and implements interface sports. Write a Java program to read required data and display details in a neat format.

4. Write a Program to create an abstract class named shape that contains two integers and an empty method named print Area().
   Provide three classes named Rectangle, Triangle and Ellipse such that each one of the classes extends the class shape. Each one of the class contains only the method print Area() that print the area of the given shape.[Abstract class].

5. Create a package to convert temperature in centigrade into Fahrenheit, and one more package to calculate the simple Interest. Implement both package in the Main () by accepting the required inputs for each application.

6. Write a Program that implements a multi-threaded program has three threads. First thread generates a random integer every second, and if the value is even, second thread computes the square of the number and prints. If the value is odd the third thread will print the value of cube of the number.[Multithreading]

7. Program that creates a user interface to perform basic integer operations.

   The user enters two numbers in the TextFields - Num1 and Num2. The result of operations must be displayed in the Result TextField when the "=" button is clicked. Appropriate Exception handling message to be displayed in the Result TextFieldwhen Num1 or Num2 is not an integer or Num2 is Zero when division operation is applied.

8. Using the swing components, design the frame for shopping a book that accepts book code, book name, and Price. Calculate the discount on code as follows.

Code	Discount rate
101	15%
102	20%
103	25%
Any other	5%

   Find the discount amount and Net bill amount. Display the bill.

Evaluation Scheme for Lab Examination:

Assessment Criteria		
Program-1	**PART-A** **Writing:7 Marks Execution: 8Marks**	**15 Marks**
Program-2	**PART-B** **Writing:10 Marks Execution:10Marks**	**20 Marks**
Practical Record		**05 Marks**
Total		**40 Marks**