JSS COLLEGE OF ARTS, COMMERCE AND SCIENCE

(Autonomous)

Ooty Road, Mysuru-570025

Model Curriculum Structure for
Bachelor of Computer Applications (BCA)Programme

Syllabus for I and II Semesters

As per

Modified NEP - 2020

2024-25

{BoS Meeting Held on 13 - 08 – 2024 Annexure I (b)}

DEPARTMENT OF COMPUTER SCIENCE

The objectives of the BCA Program

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.

Program Structures for the Under-Graduate Program
Bachelor of Computer Applications (BCA)

Semester	Coerces	Type	Credits	L: T: P	Marks	
I	Major 1to 3	Theory	9	3: 0: 0	**300**	
	Major 1 to 3	Practical	6	0: 0: 2	**150**	
	Language 1	Theory	3	3: 0: 0	**100**	
	Language 2	Theory	3	3: 0: 0	**100**	
	Constitutional Values	Theory	2	2: 0: 0	**50**	
II	Major 1to 3	Theory	9	3: 0: 0	**300**	
	Major 1 to 3	Practical	6	0: 0: 2	**150**	
	Language 1	Theory	3	3: 0: 0	**100**	
	Language 2	Theory	3	3: 0: 0	**100**	
	Constitutional Values	Theory	2	2: 0: 0	**50**	

Modified NEP - 2020 Syllabus - BCA. for 2024-25 onwards

Year	Sem	Course Code	Title	Hours / Week			Credits			Maximum Marks						Exam Duration	Total Marks
										Th. IA		Pr. IA		Exam			
				L	T	P	L	T	P	C1	C2	C1	C2	Th.	Pr.		
I	I	**GCA 101** (Theory)	Digital Computer Organization	**3**	**0**	**0**	**3**	**0**	**0**	**10**	**10**	**-**	**-**	**80**	**-**	**3 Hours**	**100**
		GCA 102 (Practical)	Office Automation and HTML	**0**	**0**	**4**	**0**	**0**	**2**	**-**	**-**	**05**	**05**	**-**	**40**	**3 Hours**	**50**
		GCA 103 (Theory)	Problem Solving using C++	**3**	**0**	**0**	**3**	**0**	**0**	**10**	**10**	**-**	**-**	**80**	**-**	**3 Hours**	**100**
		GCA 104 (Practical)	C++ Programming	**0**	**0**	**4**	**0**	**0**	**2**	**-**	**-**	**05**	**05**	**-**	**40**	**3 Hours**	**50**
		GCA 105 (Theory)	Mathematical and Statistical Computing	**3**	**0**	**0**	**3**	**0**	**0**	**10**	**10**	**-**	**-**	**80**	**-**	**3 Hours**	**100**
		GCA 106 (Practical)	Mathematical and Statistical Computing using R	**0**	**0**	**4**	**0**	**0**	**2**	**-**	**-**	**05**	**05**	**-**	**40**	**3 Hours**	**50**
	II	**GCA 201** (Theory)	Data Structures	**3**	**0**	**0**	**3**	**0**	**0**	**10**	**10**	**-**	**-**	**80**	**-**	**3 Hours**	**100**
		GCA 202 (Practical)	Data Structures using C++	**0**	**0**	**4**	**0**	**0**	**2**	**-**	**-**	**05**	**05**	**-**	**40**	**3 Hours**	**50**
		GCA 203 (Theory)	Object Oriented Programming with Java	**3**	**0**	**0**	**3**	**0**	**0**	**10**	**10**	**-**	**-**	**80**	**-**	**3 Hours**	**100**
		GCA 204 (Practical)	Programming with Java	**0**	**0**	**4**	**0**	**0**	**2**	**-**	**-**	**05**	**05**	**-**	**40**	**3 Hours**	**50**
		GCA 205 (Theory)	Operating Systems	**3**	**0**	**0**	**3**	**0**	**0**	**10**	**10**	**-**	**-**	**80**	**-**	**3 Hours**	**100**
		GCA 206 (Practical)	Shell Programming	**0**	**0**	**4**	**0**	**0**	**2**	**-**	**-**	**05**	**05**	**-**	**40**	**3 Hours**	**50**

Modified NEP - 2020 Course Content for BCA, Semesters I and II

Semester: I

Course Code: GCA101	**Course Title:** Digital Computer Organization
Course Credits: 03 (3-0-0)	**Hours/Week:** 03
Total Contact Hours: 44	**Formative Assessment Marks:** 20
Exam Marks: 80	**Exam Duration:** 03

Course Outcomes (COs):

On successful completion of this course, students will be able to:

1. Understand the digital computer system including classification of computers, anatomy of computer, input/output devices ad memory organization of computer.
2. Illustrate the types of Software, Computer languages and Translator programs.
3. Apply Boolean algebra to simplify logical expressions and solve problems using Karnaugh maps and other minimization techniques.
4. Design and analyze combinational and sequential logic circuits, including adders, subtractors, flip-flops, encoders, decoders, multiplexers, and counters.
5. Perform conversions between decimal, binary, octal, and hexadecimal number systems and carry out arithmetic operations in binary.

Unit – 1 **Course Contents**	**Hours**
Fundamentals of Computers: Introduction to Computers - Computer Definition, Characteristics Of Computers, Evolution and History Of Computer, Types Of Computer, Basic Organization Of A Digital Computer. **Input / Output Organization:** Peripheral Devices, Input–Output Interface. **Memory Organization:** Computers Memory System Overview- Characteristics and Types of Memory System. **Types Of Software**: System Software, Application Software and Utility Software; **Computer Languages:** Machine Level, Assembly Level & High-Level Languages, **Language Translators:** Assembler, Interpreter and Compiler.	**11**
Unit-2	
Number Systems: Introduction, Decimal, Binary, Octal and Hexadecimal. Inter-Conversions, Addition, Subtraction, Multiplication and Division In Binary Number System. 1’s and 2’s Complement Method in Binary Number System. Subtraction Using 1’s and 2’s Compliment, Weighted Number System, Binary Coded Decimal (BCD), Addition of BCD Numbers. Non-Weighted Number System, Applications, Excess-3, Gray Code Conversions, Gray and Binary Codes.	**11**

Unit-3	
Boolean Algebra: Basic Laws, Demorgan's Theorem, Duality Theorem, Sum of Product Method, and Products of Sum Method. Karnaugh Map (Upto 4 Variables, Don't Care Condition). **Fundamentals Of Gates:** Basic Gates, Derived Gates, and Universal Gates (Design).	**11**
Unit-4	
Combinational And Sequential Logic Circuits: Half Adder, Full Adder, Half Subtractor and Full-Subtractor. **Flip-Flops**: SR, JK, Master-Slave JK, T Flip-Flops, Decoders - 3 To 8 Lines, Encoders Octal to Binary. **Multiplexer:** 4 To 1 Line, Counters-3 Bits Binary Ripple Counter, 3 Shift Registers-Serial-In-Parallel-Out, Parallel-In-Serial-Out.	**11**

Reference:

1. Computer Fundamentals, V Rajaraman.
2. Computer System Architecture (3rd edition) Morris Mano PHI.
3. Computer Organization – by V. Carl Hamacher, Z.G.Vranesic, and S.G.Zaky, 3rd Edition. McGraw Hill,
4. Computer Organization & Design, (3rd Edition) by – D.A. Patterson & J.L. Hennessy – Morgan Kaufmann Publishers (Elsevier's)

Course Code: GCA102	**Course Title:** Office Automation and HTML
Course Credits: 02 (0-0-2)	**Hours/Week:** 04
Total Contact Hours: 60	**Formative Assessment Marks:** 10
Exam Marks: 40	**Exam Duration:** 03

Course Outcomes (COs):

On successful completion of this course, students will be able to:

1. Confidently work on Office Automation software such as word processor, spreadsheet, and power point.
2. Understand the Web Programming basics and create simple web pages using HTML.

Laboratory Program List

Part A:

1. Using a Word processor with suitable examples, write the steps and execute the following to table handling
   i. eating a table (At least 4 Columns and 6 Rows).
   ii. Entering appropriate data into the table.
   iii. Sort the table.
   iv. Apply the formulas on table numeric values.
2. Using a Word processor write the steps and execute for creating a “Mail Merge” document for “FORMLETTERS”.
3. Using a spreadsheet, with a suitable example, write steps and create a worksheet called “Employee” and calculate the following using formulas Enter Employee Code, Name and Basic Salary.
   i. Calculate DA (20% of Basic Salary).
   ii. Calculate HRA (10% of Basic Salary).
   iii. Calculate CCA (8.5% of Basic Salary).
   iv. Calculate Total Salary (Basic Salary + DA + HRA + CCA)
   v. Calculate Deductions (10% of Total Salary).
   vi. Calculate Net Salary (Total Salary – Deductions).

4. Using spread sheet draw X-Y Line Chart and Bar Charts based on the following worksheet data and write the steps

ITEM	MONTHLY SALES (in Thousands)
Cotton	2,750
Wool	3,100
Yarn	2,975
Jute	2,100
Fiber	3,010

5. Using a spreadsheet write the steps and execute the following: Roll No, Stud Name, Marks 1, Marks 2, Mark 3, Total Percentage, Result
   i. Create appropriate records
   ii. Calculate the total and marks using the formula.
   iii. Update result column using IF function. (Result: Distinction, First Class, Second Class, Pass, Fail).
6. Using Power Point with suitable examples write steps and execute the following:
   i. Create presentation slides with Titles, Sub Titles and Charts choosing different slide layouts.
   ii. Use Design templates for background.
   iii. Format the slide design.
7. Using PowerPoint, Create the presentation for
   i. "Components of PC" using organization chart.
   ii. Use different views such as slide view, slide sorter view and slide show view.
8. Using PowerPoint, Create the presentation to demonstrate
   i. Insert Images, Shapes.
   ii. Charts for tabulated data
9. Using PowerPoint, Create the presentation to demonstrate Transitions, animations and Slide show.

Part B:

1. Design a page with suitable background and text colour with the title "My First Web Page" using all the Font tag attributes.
2. Write an HTML code to design a page containing some text in a paragraph by giving a suitable heading style
3. Write an HTML program for the demonstration of Lists.
   a) Unordered List
   b) Ordered List
4. Write an HTML program for demonstrating Hyperlinks.
   a. avigation from one page to another.
   b. Navigation within the page.
5. Write an HTML program for a time-table using tables.
6. Write an HTML program to develop a static Registration Form.
7. Write an HTML program to develop a static Login Page.
8. Write an HTML code to create a web page with a pink background and display a moving message in red.
9. Write an HTML program to develop a static Web Page for a Shopping Cart.
10. Write an HTML program to develop a simple calculator.

Course Code: GCA103	**Course Title:** Problem Solving using C++
Course Credits: 03 (3-0-0)	**Hours/Week:** 03
Total Contact Hours: 44	**Formative Assessment Marks:** 20
Exam Marks: 80	**Exam Duration:** 03

Course Outcomes (COs):

On successful completion of this course, students will be able to:

1. Understand the fundamental concepts and benefits of Object-Oriented Programming (OOP) and how it differs from Procedure-Oriented Programming paradigms.
2. Interpret and apply C++ syntax and structure, including input-output statements, keywords, identifiers, constants, variables, data types, operators, expressions and file handling to create basic programs and solve problems.
3. Describe the control structures, functions, and different parameter passing methods and write programs to solve problems.
4. Demonstrate the concepts of classes and objects, access specifiers, constructors, destructors, and OOP features like polymorphism, and inheritance with the help of programs.

Unit – 1 **Course Contents**	**Hours**
Introduction to Programming: Program development life cycle, Introduction to Procedure Oriented Programming and Object-Oriented Programming (OOP) paradigms, basic concepts of OOP, benefits and applications of OOP. **Introduction to C++:** Overview of C++, Structure of C++ Program, Input-Output statements, Keywords, Identifiers, Constants, Variables, Data types, Operators, Types of Operators, Expressions, Precedence of Operators, Type Conversion, Storage classes.	**11**
Unit-2	
Control statements: Selection And Iteration Statements, Loop Control Statements. **Modular Programming:** Functions and Its Types, Recursion, Functions with Default Arguments, Inline Functions, Function Overloading, Call by Value and Reference, Math Library Functions.	**11**

Unit-3	
Derived Data Types: Arrays, Array Types, Strings, String Manipulation Functions, Pointers, Pointer Arithmetic. **Managing Console, I/O Operations:** C++ Stream, C++ Stream Classes, Unformatted I/O Operations, Formatted Console I/O Operations, Managing Output with Manipulators. **User Defined Data Type:** Class Definition, Instance Variables, Member Methods, Accessing Members, Access specifiers, this pointer, Friend Function, Constructors, Types of Constructors, Destructor.	**11**
Unit-4	
Polymorphism: Operator Overloading, Rules for Operator Overloading, Overloading Unary and Binary Operators. **Inheritance:** Inheritance, Types of Inheritance, Virtual Functions and Abstract Classes. **File Handling:** Introduction To Files and File Handling, File Opening Modes, Classes for File Stream Operations, and File I/O Operations (Opening, Reading, Writing, Append and Closing).	**11**

Reference Books:

1. Object-Oriented Programming With C++, By M. T. Somashekara, D. S. Guru, H. S. Nagendraswamy, K. S. Manjunatha, PHI Learning Pvt. Ltd.
2. Object-Oriented Programming with C++, By E Bala Guruswamy, Tata McGraw-Hill Publication Company Ltd.
3. The C++ Programming Language, By Stroustroup, Bjarne, Addison Wesley.
4. How To Solve It by Computer - R G Dromey, Prentice-Hall International.

Course Code: GCA104	**Course Title:** C++ Programming
Course Credits: 02 (0-0-2)	**Hours/Week:** 04
Total Contact Hours: 60	**Formative Assessment Marks:** 10
Exam Marks: 40	**Exam Duration:** 03

Course Outcomes (COs):

On successful completion of this course, students will be able to:

1. Demonstrate fundamental C++ programming concepts by writing programs for simple problems.
2. Utilize features of C++, such as recursion, function overloading, and friend functions, to enhance the functionality and efficiency of programs.
3. Design and construct classes and objects in C++ to model real-world entities, and demonstrate inheritance, operator overloading, constructors, and file-handling operations.

Laboratory Program List

Part -A

1. Program to swap 2 numbers with and without using temporary variables.
2. Program to convert the Fahrenheit to Celsius and vice-versa.

1. Program to compute to add and multiply two complex numbers.
2. Program to demonstrate functions of a simple calculator.
3. Program to display multiplication table of a given number.
4. Program to check whether a number is a palindrome or not.
5. Program to generate Fibonacci series.
6. Program to compute sum of principle diagonal, lower diagonal and upper diagonal elements of a matrix.
7. Program to reverse a given string without using built-in function.
8. Program to demonstrate the usage of any five Math.h library functions.

Part-B

9. Program to demonstrate call by value and call by reference.
10. Program to generate factorial of a given number using recursion.
11. Program to create a Class for representing student details with appropriate member functions to accept and display the details.
12. Program to demonstrate function overloading.
13. Program to demonstrate friend function.
14. Program for single inheritance.
15. Program to demonstrate multilevel inheritance.
16. Program to demonstrate operator overloading.
17. Program to demonstrate the usage of default and parameterized constructors.
18. Program to read and display the contents of a text file.

Course Code: GCA105	**Course Title:** Mathematical and Statistical Computing
Course Credits: 03 (3-0-0)	**Hours/Week:** 03
Total Contact Hours: 44	**Formative Assessment Marks:** 20
Exam Marks: 80	**Exam Duration:** 03

Course Outcomes (COs):

On successful completion of this course, students will be able to:

1. Construct, evaluate, and apply logical statements and truth tables, understand the principles of set theory, perform various set operations, and effectively use Venn diagrams for solving complex problems.
2. Understand Cartesian products, relations, and their properties, including equivalence relations and partitions. They will also gain skills in function composition, inverse functions, and representing relations through matrices and directed graphs.
3. Organize and interpret data using statistical methods, calculate measures of central tendency and dispersion, analyze correlation between variables, and perform linear regression analysis.

Unit – 1 **Course Contents**	**Hours**
Mathematical Logic Introduction: Statements Connectives - Negation, Conjunction, Disjunction- Statement Formulas and Truth Tables- Conditional and Hours Bi Conditional Statements- Tautology, Contradiction. **Set Theory:** Sets And Subsets, Set Operations and The Laws of Set Theory, Counting and Venn Diagrams.	**11**
Unit-2	
Cartesian Products and Relations, Properties of Relations. **Computer Recognition**: Relation Matrices and Directed Graphs, Equivalence Relations and Partitions. **Functions**: One-to-One, Onto Functions, Function Composition, and Inverse Functions.	**11**
Unit-3	
Statistical methods: Introduction, Definitions, Classifications, Frequency Distribution, Mean - Arithmetic Mean for Grouped and Ungrouped Data. **Median**: Meaning, Calculations of Median for Ungrouped. **Mode:** Meaning, Calculations of Mode for Discrete Series and Continuous Series.	**11**

Unit-4	
Standard Deviation: Meaning, Standard Deviation for Actual Mean Method, Assumed Mean Method, and Step Deviation Method Using Discrete Series and Continuous Series. **Coefficient of Variation:** Meaning and Problems. **Correlation:** Meaning, Types, Rank Correlations and Problems. **Simple Linear Regression**: Meaning, Properties of Regression Coefficients.	**11**

Reference Books:

1. Ralph P. Grimaldi, "Discrete and Combinatorial Mathematics", 5th Edition, Pearson Education, 2004.
2. Kenneth H. Rosen, "Discrete Mathematics and its Applications", 6th Edition, McGraw Hill, 2007.
3. Jayant Ganguly, "A Treatise on Discrete Mathematical Structures", Sanguine Pearson, 2010.
4. D.S. Malik and M.K. Sen, "Discrete Mathematical Structures: Theory and Applications", Thomson, 2004.
5. Thomas Koshy, "Discrete Mathematics with Applications", Elsevier, 2005, Reprint 2008.
6. Fundamentals of Mathematical Statistics by Gupta and Kapoor (Sultan Chand).
7. Mathematical Statistics by John Freund (Prentice Hall India Pvt. Ltd.)

Course Code: GCA106	**Course Title:** Mathematical and Statistical Computing Using R
Course Credits: 02 (0-0-2)	**Hours/Week:** 04
Total Contact Hours: 60	**Formative Assessment Marks:** 10
Exam Marks: 40	**Exam Duration:** 03

Course Outcomes (COs):
On successful completion of this course, students will be able to:

1. Develop practical skills in implementing set operations and function operations using R programming.
2. Implement logic gates using R and perform comprehensive statistical analysis including calculations of central tendency and conduct linear regression analysis.
3. Compute Cartesian products and analyze relations for properties such as reflexivity, symmetry, and transitivity through R scripts.

Laboratory Program List

Part A

1. Write an R Program to implement operations of Set (Union, Intersection, Difference, Subset).
2. Write an R Program to implement logic gates (NOT, AND, OR, XOR).
3. Write an R Program to implement a Cartesian Product of Two sets.
4. Write an R Program to check whether the given relation is Reflexive.
5. Write an R Program to check whether the given relation is Symmetric.
6. Write an R Program to check whether the given relation is Transitive.
7. Write an R program to implement one to one function.
8. Write an R Program to implement inverse function.

Part B

1. Write an R Program to Calculate central tendency (mean, median, mode).
2. Write an R Program to Calculate standard deviation and variance for discrete & continuous series.
3. Write an R Program to Calculate the coefficient of variance for discrete & continuous series.
4. Write an R Program to Calculate simple Linear Algebra Operations.
5. Write an R Program to Calculate the arithmetic mean for grouped and ungrouped data.
6. Write an R Program to Calculate cumulative sums, and products, minima, maxima.
7. Write an R Program to Calculate frequency distribution for discrete & continuous series.
8. Write an R Program to Calculate Simple Linear Regression.

Semester II

Course Code: GCA201	**Course Title:** Data Structures
Course Credits: 03 (3-0-0)	**Hours/Week:** 03
Total Contact Hours: 44	**Formative Assessment Marks:** 20
Exam Marks: 80	**Exam Duration:** 03

Course Outcomes (COs):

On successful completion of this course, students will be able to:

1. Understand the basics of Data Structures.
2. Identify the appropriate data structures and algorithms for solving real-world problems.
3. Understand the practical applications of Tree and Graphs.
4. Understand the fundamentals of sorting and searching algorithms.

Unit – 1 **Course Contents**	**Hours**
Introduction: Data Structure Definition, Basic Terminology and Concepts, Importance of Data Structures in Programming. Classification of Data Structures. Primitive Data Structures, Non-Primitive Data Structures. **Stack:** Definition, Memory Representation, Algorithms for Stack Operations (Push, Pop), Applications of Stack.	**11**
Unit-2	
Queue: Definition, Memory Representation, Linear Queue, Circular Queue, Enqueue, Dequeue. Applications of Queue. **Linked Lists**: Definition, Types. **Singly Linked List:** Implementation, Insertion [At the Beginning], Deletion [At the End]. **Doubly Linked List:** Memory Representation of Singly Linked List and Doubly Linked Lists. Applications of Linked List.	**11**
Unit-3	
Tree: Definition, Memory Representation Using Array and Linked List. **Binary Tree:** Definition, Traversal Algorithms [Pre-Order, In-Order, Post-Order], Construction of Tree from In-Order and Pre-Order, In-Order and Post-Order. **Binary Search Trees:** Insertion of a Node, Deletion of A Node. **Advanced Tree Structures AVL And B-Trees:** Definition and Applications.	**11**

Unit-4	
Graph: Definition, Memory Representation of Graph. Adjacency Matrix, Adjacency List. Graph Traversal Algorithms: Breadth-First Search (BFS), Depth-First Search (DFS). **Sorting Techniques:** Bubble Sort, Selection Sort [Algorithm, Time & Space Complexity]. **Searching Techniques:** Linear And Binary Search [Algorithm, Time & Space Complexity]. **Heap:** Heap Operations and Applications.	**11**

Reference Books:

1. Data Structures Through C++ (4th Edition) Yashvant Kanetkar.
2. Data Structures and Algorithm Analysis in C++" by Mark Allen Weiss.
3. Data structure and Algorithms using C++ by Sachi Nandan Mohanty, Pabitra Kumar Tripathy.
4. Data Structures and Algorithms in C++, Second Edition by Adam Drozdek.

Course Code: GCA202	**Course Title:** Data Structures using C++
Course Credits: 02 (0-0-2)	**Hours/Week:** 04
Total Contact Hours: 60	**Formative Assessment Marks:** 10
Exam Marks: 40	**Exam Duration:** 03

Course Outcomes (COs):

On successful completion of this course, students will be able to:

1. Implement data structures using C++.
2. Demonstrate searching and sorting techniques using C++.
3. Demonstrate advanced programming skills through C++ programming language.

Laboratory Program List

Part A:

1. Program to find the GCD of two numbers.
2. Program to implement Tower of Hanoi.
3. Program to print Fibonacci series.
4. Program to find largest and smallest element in an array.
5. Program to perform stack operations.
6. Program to perform Linear queue operations
7. Program to insert a node at the beginning of a singly linked list.
8. Program to delete a node at the end of a singly linked list.

Part B:

1. Program to construct a binary search tree
2. Program for Binary Tree traversal.
3. Program to implement DFS
4. Program to implement BFS
5. Program to Sort an Array (Selection Sort)
6. Program to Sort an Array (Bubble Sort)
7. Program to perform Linear Search of an Element in an Array.
8. Program to perform Binary Search of an Element in an Array.

Course Code: GCA203	**Course Title:** Object-Oriented Programming with Java
Course Credits: 03 (3-0-0)	**Hours/Week:** 03
Total Contact Hours: 44	**Formative Assessment Marks:** 20
Exam Marks: 80	**Exam Duration:** 03

Course Outcomes (COs):

On successful completion of this course, students will be able to:

1. Understand the Java programming fundamentals.
2. Describe with examples of basic Java OOP concepts.
3. Understand the Java Interfaces and Packages.
4. Deliberate the Details of Multithreading, Exception Handling & File Handling
5. Design GUI applications using tools like AWT.

Course Contents:

Unit – 1 **Course Contents**	**Hours**
Fundamentals of Object-oriented Programming: Object-oriented Paradigm, Basic Principles of Object-oriented Programming, Advantages of Object-Oriented Programming, Applications of Object-Oriented Programming. **Introduction to Java Language:** Java History, Features, Overview, Difference between C, C+ + and Java, Java Environment- JDK, JVM, JRE and API, Java Program Structure, Java Tokens, Implementing a Java Program, Command Line Arguments. **Java Programming Fundamentals:** Data types, Variables & Constants, Keywords & Naming Conventions, Type Casting, Operators and Expressions, Control Structures, Jumping Statements.	**11**
Unit-2	
Classes & Objects: Basics of Objects and Classes, Constructors, Access Modifiers, Method Overloading, Overloading Constructors, Static members, this keyword. **Arrays**: One-dimensional Arrays, Two dimensional Arrays, Array of Objects. **Strings**: String Handling functions.	**11**
Unit-3	
Multithreading in Java: Concepts of Thread, Thread Life Cycle, Creating Threads & Implementing Runnable Interface, Thread Synchronization & Thread Priority. **Exception Handling:** Concepts of Exception, Different Types of Exceptions, Creating User-Defined Exceptions Using Try-Catch-Finally-Throw Blocks, Nested Try, Catch, Throw, and Throws Blocks.	**11**

Unit-4	
File Handling: I/O Handling, I/O Streams, Types of Files, Byte Stream, Binary I/O Classes & Its Hierarchy, FileInputStream & FileOutputStream Classes, Object I/O Classes. **Event Handling & GUI programming:** Event Handling, Event Types, Event Handling Mechanism, Keyboard & Mouse Handling, Introduction to AWT & GUI basics, AWT hierarchy of classes, AWT controls – Frames, Panels, Layout managers & other controls of AWT.	**11**

Reference Books:

1. D.S. Guru, M.T. Somashekara, & K.S. Manjunatha, Object Oriented Programming with Java, PHI Learning, 2017.
2. E Balagurusamy, Programming with JAVA, TMH, 2007
3. Herbert Schildt, Java 7, The Complete Reference, 8th Edition, 2009

Course Code: GCA204	**Course Title:** Programming with Java
Course Credits: 02 (0-0-2)	**Hours/Week:** 04
Total Contact Hours: 60	**Formative Assessment Marks:** 10
Exam Marks: 40	**Exam Duration:** 03

Course Outcomes (COs):

On successful completion of this course, students will be able to:

1. Implement simple programs using Java Fundamental concepts.
2. Identify classes, objects, members of a class, and the relationships among them needed for finding the solution to specific problems using Objected Oriented Programming concepts of Java.
3. Design & Develop simple GUI programs using the AWT GUI tool.

Laboratory Program List

Part A:

1. Program to find whether the given number is Positive, Negative, or Zero.
2. Program to list the factorial of the numbers 1 to 10.
3. Program to demonstrate classes & objects.
4. Program to demonstrate method overloading.
5. Program to demonstrate single inheritance (simple calculator – base class, Advanced Calculator – derived class).
6. Program to find Maximum & Minimum elements in a one-dimensional array of numbers.
7. Program to check whether the given string is palindrome or not.
8. Program to create a ‘Student’ class with Reg.no., name, and marks of 3 subjects. Calculate the total marks of 3 subjects and create an array of 3 student objects & display the results.

Part B:

1. Program to generate negative array size exception
2. Program to generate NullPointer Exception.
3. Program that reads two integer numbers for the variables a and b. The program should catch NumberFormatException & display the error message.
4. Program to create AWT window with 4 buttons M/A/E/Close. Display M for Good Morning, A for Afternoon, E for Evening, and Close button to exit the window.
5. Program to demonstrate the various mouse handling events.
6. Program to read and write Binary I/O files.
7. Program to create a window with three buttons father, mother, and close. Display the respective details of the father and mother as name, age and designation using AWT controls.
8. Program to create menu bar and pull-down menus.

Course Code: GCA205	**Course Title:** Operating Systems
Course Credits: 03 (3-0-0)	**Hours/Week:** 03
Total Contact Hours: 44	**Formative Assessment Marks:** 20
Exam Marks: 80	**Exam Duration:** 03

Course Outcomes (COs):

On successful completion of this course, students will be able to:

1. Understand the fundamentals of the operating system.
2. Describe the concepts of process, process management, CPU Scheduling, process synchronization, Deadlocks, memory management, and Virtual Memory management.
3. Illustrate the file system and structure.
4. Understand the UNIX OS, Shell Programming, Conditional Control Structures in Shell Programming.

Course Contents:

Unit – 1 **Course Contents**	**Hours**
Introduction: Definition, Computer System Components, User View, System View and System Goals, Batch Systems, Multi Programmed Systems, Time-Sharing Systems, Real-Time Systems, System Components, Operating System Services. **Process**: Process Concept, Process State Diagram Process Control Block, Process Scheduling- Scheduling Queues, Scheduler, Cooperating Process, Inter-process Communication. **CPU Scheduling**: Basic Concepts, Preemptive and Non-Preemeptive Scheduling, Scheduling Criteria, Scheduling Algorithms-FCFS, Shortest Job First Priority Scheduling, Round Robin Scheduling.	**11**
Unit-2	
Process Synchronization: The Critical Section Problem, Solution for Critical Section Problem, Bakery Algorithm, Semaphores-Meaning, Types of Semaphores, Synchronization Problems- Bounded Buffer Problem, Readers-Writers Problem. **Deadlocks:** Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.	**11**

Unit-3	
Memory Management: Introduction, Logical Versus Physical Address Space, Dynamic Loading, Dynamic Linking, Swapping, Contiguous Allocation, Partitioned Memory Allocation, Paging, Virtual Memory Management-Segmentation, Segmentation with Paging. **File System:** File Concepts, File Attributes, File Operations, File Types, File Structure, Access Methods, Directory Structure, File-System Structure, Allocation Methods- Contiguous Allocation, Linked Allocation and Indexed Allocation, Free Space Management.	**11**
Unit-4	
Introduction to Unix System: The Unix Operating System, The UNIX architecture. **Shell Programming:** Vi editor, shell types, shell command line processing, shell script features, executing a shell script, system and user-defined variables, expr command, shell screen interface, read and echo statement, command substitution, escape sequence characters, shell script arguments, positional parameters, test command, file test, string test, numeric test. **Conditional Control Structures:** if statement, case statement Looping Control Structure-while, until, for, statements. Jumping Control Structures – break, continue, exit. Shell Programs covering the above concepts.	**11**

Reference Books:

1. Operating System Concepts – 5th edition by Abraham Silberschartz and Peter Galvin, McGraw Hill,2000
2. Modern Operating Systems – Andrew S Tanenbaum, Prentice Hall
3. Operating Systems: Internals and Design Principles, William Stallings, Prentice Hall
4. Sumitabha Das: UNIX – Concepts and Applications, 4th Edition, Tata McGraw Hill, 2006.

Course Code: GCA206	**Course Title:** Shell Programming
Course Credits: 02 (0-0-2)	**Hours/Week:** 04
Total Contact Hours: 60	**Formative Assessment Marks:** 10
Exam Marks: 40	**Exam Duration:** 03

Course Outcomes (COs):

On successful completion of this course, students will be able to:

1. Develop skill in shell scripting to perform simple operations and problems.
2. Perform file manipulation using shell scripts.
3. Understand and implement shell scripts for system information.

Laboratory Program List

Part A:

1. Write a shell script to swap 2 values.
2. Write a shell script to check if the given number is even or odd.
3. Write a shell script to find the largest of 3 numbers.
4. Write a shell script to perform arithmetic operations.
5. Write a shell script to find the sum of the first 10 natural numbers.
6. Write a shell script to display the multiplication table of a given number.
7. Write a shell script to find the length of a given string.
8. Write a shell script to find the factorial of a given number.
9. Write a shell script that counts the numbers of lines and number of words present in a given file.
10. Write a shell script to display the Fibonacci series up to N number.

Part B:

1. Write a shell script to search for a particular element from an array of elements.
2. Write a shell script to calculate the TA, HRA, and DA of an employee.
3. Write a shell script that displays a list of all files in the current directory to which the user has read write and execute permissions.
4. Develop an interactive script that asks for a word and file name and then tells how many times that word occurred in the file.
5. Write a shell script to extract a sub-string from a given string.
6. Perform the following operations
   a) Concatenate 2 strings
   b) Rename a file
   c) Delete a file
   d) Copy the file

7. Write a shell script to display the
   a) Version of the shell
   b) The user information
   c) Login date and time
   d) List of processes running on the system
   e) User home directory
12. Write a C program to display the PID of the parent and the PID of a child process.
13. Write a shell script that takes two filenames as arguments. It should check whether the contents of two files are same or not, if they are same then second file should be deleted.
14. Assume a file with the given information
15. First Name Middle Name Age

    ------------- ----------------- ------

16. Write a shell script to

a. Sort the first name in alphabetical order

b. Sort the age in terms of ascending order

c. Sort the age in terms of descending order

d. Sort the middle name in alphabetical order

Practical Evaluation [50 Marks]

- **Internal assessment: C1= 05 Marks, C2 =05 Marks [Total = 10 Marks]**
- **Semester End Practical Examination C3=40 marks**
- **Writing:** One program from both Part A and Part B (10 Marks each): **10 + 10 = 20 Marks**
- **Debug and Execution:** (05 Marks each) **05 + 05 = 10 Marks**
- **Record: 05 Marks**
- **Viva:** 05 **Marks**

Theory Evaluation [100 Marks]

CIE, SEE and QP Pattern for Theory Courses:

- Total Lecture hours per paper: 44
- No. of Units 4 (11 Hours Each)
- Internal Assessment **C1 = 10 Marks, C2 = 10 Marks**
- Semester End Theory Exam **C3 = 80 Marks**

Question Paper Pattern

Instructions: Answer Part-A and Part-B:

Part-A

Answer any 10 out of 12 Questions (3 Questions drawn from each unit).
Each question carries 2 Marks. (10 X 2 =20) Q. No. 1 to Q. No. 12.

Part-B

Answer the following Questions.

13. a)

OR

b)

14. a)

OR

b)

15. a)

OR

b)

16. a)

OR

b)

Each main Question with internal choice and carries 15 Marks. (4 X 15 =60)
(Each main question may split into sub-questions like (10 + 5, 8 +7, 5+5+5, …) and with a maximum of 3 sub-questions)