BANGALORE UNIVERSITY
ಬೆಂಗಳೂರು ವಿಶ್ವವಿದ್ಯಾಲಯ
Jnanabharathi, Bengaluru, Karnataka – 560 056

SCHEME AND SYLLABUS

For the Course

BACHELOR OF COMPUTER APPLICATIONS
(Data Science)

State Education Policy 2024 SCHEME

Academic Year 2024- 2025 and onwards

Members of the BoS in Computer Science

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

BANGALORE UNIVERSITY

Department of Computer Science and Applications
BCA Course Structure
BCA (with specialization in Data Science)
AS PER STATE EDUCATION POLICY

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
1	24BCA11	Discrete Structures	03	80	20	100	3
	24BCA12	Problem-Solving Technique	03	80	20	100	3
	24BCA13	Computer Architecture	03	80	20	100	3
	24BCA12P	Problem-Solving Technique Lab	04	40	10	50	2
	24BCA13P	Computer Architecture Lab	04	40	10	50	2
	24BCA1P	Office Automation Tools	4	40	10	50	2
	24BCAL11	Language L1	04	80	20	100	3
	24BCAL12	Language L2	04	80	20	100	3
	24BCACC1	The Constitution of India	02	40	10	50	2
2	24BCA21	Data Structure	03	80	20	100	3
	24BCA22	Object-Oriented Programming Using JAVA	03	80	20	100	3
	24BCA23	Operating Systems	03	80	20	100	3
	24BCA21P	Data Structure Lab	04	40	10	50	2
	24BCA22P	Object Oriented Programming Lab	04	40	10	50	2
	24BCA21P	LINUX and Shell Programming Lab	04	40	10	50	2
	24BCAL21	Language L1	04	80	20	100	3
	24BCAL22	Language L2	04	80	20	100	3
	24BCACC2	Environmental Studies	02	40	10	50	2
3	24BCA31	Database Management System	03	80	20	100	3
	24BCA32	Probability and Statistics	04	80	20	100	4
	24BCA33	Artificial Intelligence	04	80	20	100	4
	24BCA31P	Database Management System Lab	04	40	10	50	2

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
	24BCA32P	Artificial Intelligence Lab using Python	04	40	10	50	2
	24BCAE1	Elective: I Basics of Data Analytics using Spreadsheet	02	40	10	50	2
	24BCAL31	Language L1	04	80	20	100	3
	24BCAL32	Language L2	04	80	20	100	3
4	24BCA41	Computer Networks	03	80	20	100	3
	24BCA42	Design and Analysis of Algorithms	04	80	20	100	4
	24BCA43	Software Engineering	04	80	20	100	4
	24BCA41P	Computer Networks Lab	04	40	10	50	2
	24BCA42P	Design and Analysis of Algorithms Lab	04	40	10	50	2
	24BCAE2	Elective: II Data Visualization	02	40	10	50	2
	24BCAL41	Language L1	04	80	20	100	3
	24BCAL42	Language L2	04	80	20	100	3
	24BCASEC1	Office Management Tools	02	40	10	50	2
Semester – V							
5	24BCA51	Introduction to Data Science	03	80	20	100	3
	24BCA52	Time Series Analysis	03	80	20	100	3
	24BCA53	Machine Learning	03	80	20	100	3
	24BCA51P	Data Science Lab	04	40	10	50	2
	24BCA52P	Time Series Analysis Lab	04	40	10	50	2
	24BCA53P	Machine Learning Lab	04	40	10	50	2
	24BCASEC2	Quantitative Techniques	02	40	10	50	2
Semester – VI							
6	24BCA61	Big Data Analytics	05	80	20	100	5
	24BCA62	Exploratory Data Analysis	05	80	20	100	5
	24BCA63	Project Work	10	80	20	100	5
	24BCASEC3	Soft Skills	02	40	10	50	2

Department of Computer Science and Applications

BANGALORE UNIVERSITY, BANGALORE

Program Outcome

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

DISCRETE STRUCTURES

Course Outcome

CO1 Apply Set theory, Relations, Functions and Mathematical Induction to solve problems

CO2 Getting familiar with counting techniques and Mathematical Logic, and apply the same to address the real-life problems

CO3 Learn different operations of a Matrix and apply them to solve real-life problems.

CO4 Apply graph theory concepts to solve the problems of real world.

UNIT – I 11 Hours

Set Theory: Fundamentals of Set theory, Set Operations, Laws of Set Theory, Counting and Venn Diagrams, Cartesian Product, Relations, Types of Relations, Functions, Types of Functions, Function Composition, Inverse Functions. Mathematical Induction.

UNIT – II 11 Hours

Logic and Counting: Fundamentals of Logic, Propositional Logic, Logical Connectives and Truth Tables, Logic Equivalence, Tautology and Contradiction. Basics of counting, Counting Principles, Pigeonhole Principle, Permutation, Combinations.

UNIT - III 11 Hours

Matrices: Basics of Matrix, Types of Matrices, Operations on Matrices, Inverse of a matrix, Solution for system of linear equations, Determinant, Properties of Determinant, Cramer's Rule, Introduction to Eigen Values and Eigen Vectors.

UNIT - IV 12 Hours

Graph Theory: Graphs: Introduction, Representing Graphs, Operations on graphs, Directed Graphs Graph Isomorphism, Paths, Cycles, Euler Graph, Hamilton Graph, Planar Graphs.
Trees: Introduction, Applications of Trees, Spanning Trees, Minimum Spanning Trees, Prim's and Kruskul's Algorithms.

Text Book

1 Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, 5th Edition, Pearson Education, 2006.

2 Richard Bronson, Schaum's Outline of Matrix Operations, McGraw-Hill publications, 2nd Edition,

Reference Books

1 Gregory Hartman, Fundamentals of Matrix Algebra, Third Edition,2011.

2 Gary Haggard, John Schlipf and Sue Whitesides: Discrete Mathematics for Computer Science, (with Student Solutions Manual CD-ROM), Cengage Learning; 1st edition, 2005.

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

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

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

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

PROBLEM-SOLVING TECHNIQUE

Course Outcomes

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

CO1 To understand algorithmic strategies for enhancing problem-solving proficiency

CO2 Demonstrate problem-solving tools and techniques using C.

CO3 To analyze the given problems and use appropriate algorithms.

CO4 To implement sorting and searching techniques to develop programs.

UNIT –1 12 Hours

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

UNIT-II 11 Hours

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

UNIT - III 11 Hours

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

UNIT - IV 11 Hours

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

Text Book

1 R. G. Dromey, "How to Solve it by Computer", Person Education India, 2008.

2 Brain M. Kernighan and Dennis M. Ritchie, " The C Programming Language", 2nd edition, Princeton Hall Software Series, 2012.

3 Thomas H Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, The MIT Press Cambridge, Massachusetts London, England, 2008.

Reference Books

1 E. Balaguruswamy, "Programming In ANSI C", 4th edition, TMH Publications, 2007

2 Greg Perry and Dean Miller, "C programming Absolute Beginner's Guide", 3rd edition, Pearson Education, Inc, 2014.

3 Donald E. Knuth, The Art of Computer Programming", Volume 2: Seminumerical Algorithms, 3rd Edition, Addison Wesley Longman, 1998.

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

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

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

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

COMPUTER ARCHITECTURE

Course Outcome

CO1 Understand various arithmetic and logical operations on different types of numbers to design an arithmetic and logic unit.

CO2 Demonstrate Design and implement sequential logic circuits using ICs

CO3 Analyze the basics of computer organization and its design and the basic processing unit

CO4 Implement the instruction sets and to develop assembly language programming skills.

UNIT-1 [12Hours]

Number Systems: Decimal, Binary, Hexadecimal, Octal Number System Conversions, Binary Arithmetic, Complements- r's complement, (r-1)'s complement, Addition and subtraction of BCD, Octal Arithmetic, Hexadecimal Arithmetic, Binary Codes, Decimal Codes, Error detecting and correcting codes, ASCII, EBCDIC, UNICODE, Digital Logic Circuits: Digital Computers, Logic Gates, Universal Gates, Boolean algebra, Map Simplification.

UNIT-2 [11Hours]

Combinational Circuits- Half Adder and Full Adder, Flip-Flops- SR Flip- Flop, D Flip-Flop, J-K Flip-Flop, T Flip-Flop, Sequential Circuits- Flip-Flop input equations, State Table, State Diagram and problems. Digital Components: Integrated Circuits, Decoders-3-to-8-line decoder, NAND gate Decoder, Octal to Binary Encoder, Multiplexers- 4-to-1 line Multiplexer, Registers- 4 bit register with parallel load, Shift Registers- Bidirectional shift register with parallel load, Binary Counters-4-bit synchronous binary counter.

UNIT-3 [11Hours]

Basic Computer Organization and Design: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input-Output Interrupt, Complete Computer Description, Design of Basic Computer, Design of Accumulator logic. Central Processing Unit: Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer (RISC), CISC Vs RISC.

UNIT-4 [11Hours]

Introduction to 8085 Assembly language programming: Architecture of 8085, Pin Configuration, The 8085-programming model, Instruction classification, Instruction, data formats, and storage. How to write assemble and execute a simple program, overview of 8085 instruction set. Introduction to 8085 Instructions: Instruction classification of 8085 based on word length and functions, Data Transfer operations, Arithmetic operations, Logic Operations, Brach operations, Writing Assembly language programs, Addressing modes of 8085.

Text Book

1 M. Morris Mano- "Computer System Architecture", 3rd Edition Pearson India, 2019.

2 Ramesh Gaonkar- "Microprocessor Architecture, Programming and Applications with the 8085", 5th Edition, Penram International Publishing (India) Private Limited,2007.

3 Andrew S. Tanenbaum, Todd Austin –"Structured Computer Organization", PHI Pearson 6th, Edition,2013.

Reference Books

1 William Stallings- "Computer Organization and Architecture", Pearson/PHI, 6th Edition,2007.

2 Andrew S. Tanenbaum-" Structured Computer Organization", PHI /Pearson 4th Edition,1998.

3 M.V. Subramanyam, "Switching Theory and Logic Design", Laxmi Publications Ltd, 2011.

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

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

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

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

PROBLEM-SOLVING TECHNIQUE LAB

Write, and execute C Program for the following:

1. To read the radius of the circle and to find area and circumference.
2. To read the numbers and find the biggest of three.
3. To check whether the number is prime or not.
4. To find the root of quadratic equation.
5. To read a number, find the sum of the digits, reverse the number and check it for palindrome.
6. To read the numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers.
7. To read percentage of marks and to display appropriate message. If a percentage is 70 and above- Distinction, 60-69 – First Class, 50-59 – Second Class, 40-49 Pass, below 40 – Fail.(Demonstrate of if-else ladder)
8. To simulate a simple calculator with addition, subtraction, multiplication, division and it should display the error message for division of zero using switch case.
9. To read marks scored by n students and find the average of mark (Demonstration of single dimensional array)
10. To remove duplicate elements in a single dimensional array.
11. To find the factorial of a number.
12. To generate Fibonacci series.
13. To demonstrate string functions. (String Length, String Copy, String Concatenate, String Comparison)
14. To find the length of the string without using built-in function.
15. To read, display and add two n x m matrices using function.

16. To read a string and to find the number of alphabets, digits, vowels, consonants, space and special characters.
17. To swap two numbers using pointers.
18. To demonstrate student structure to read & display records of n students.
19. To demonstrate the difference between structure and union for the following Student name (String), Student roll no(integer), Student mark(float)
20. To design the following pattern using nested for loop:

```
        *
      *   *
    *   *   *
  *   *   *   *
*   *   *   *   *
```

COMPUTER ARCHITECTURE LAB

1. Write an 8085 Program to swap two 8-bit numbers.
2. a. Write a Program to find the largest of two numbers
   b. Write an 8085 Program to find the smallest of two numbers
3. Write an 8085 Program to find whether an 8-bit number is positive, negative or zero. If positive display EE, if negative display FF, if zero display DD.
4. Write an 8085 Program to check whether 4^{th} bit of a number is zero or one. If 4^{th} bit is 1 display FF, if 4^{th} bit is 0 display DD.
5. Write an 8085 Program to calculate the sum of first ten natural numbers.
6. Write an assembly language program in 8085 microprocessors to find the sum of digits of an 8-bit number.
7. Write an 8085 Program to find the reverse of an 8-bit number
8. Write an 8085 Program to check whether 1-byte number is a palindrome or not. If it is a palindrome display FF otherwise display DD.
9. Write an 8085 Program to check whether a number is ODD or EVEN. If Even no. display DD, if odd no. display FF.
10. Write an 8085 program to count a number of ones in the given 8-bit number.
11. Write an 8085 program to find Addition & Subtraction of two 8 –bit HEX numbers.
12. Write an 8085 program to find Addition of two 16 –bit numbers.
13. Write an 8085 program to find Subtraction of two 16 –bit numbers.
14. Write an 8085 program for Swapping of two 16-bit numbers.
15. Write an 8085 program to implement 2 out of 5 codes
16. Write an 8085 program to generate Fibonacci series
17. Write an 8085 program to find the first ten terms of odd and even numbers.
18. Write an 8085 program to find 4-Digit BCD addition.
19. Write an 8085 program to find Multiplication of 2-digit BCD numbers.
20. Write an 8085 program to find division of two 8-bit umbers.

DATA STRUCTURES

Course Outcome

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

UNIT I 11 Hours

Introduction and Overview: Definition, Elementary data organization, Data Structures, data Structures operations, Abstract data types, algorithms complexity, time-space trade-off. Preliminaries: Mathematical notations and functions, Algorithmic notations, control structures, Complexity of algorithms, asymptotic notations for complexity of algorithms. Introduction to Strings, Storing String, Character Data Types, String Operations, word processing, Introduction to pattern matching algorithms.

UNIT II 11 Hours

Arrays: Definition, Linear arrays, arrays as ADT, Representation of Linear Arrays in Memory, Traversing Linear arrays, Inserting and deleting, multi-dimensional arrays, Matrices and Sparse matrices, searching and sorting techniques using array. Linked list: Definition, Representation of Singly Linked List in memory, Traversing a Singly linked list, Searching in a Singly linked list, Memory allocation, Garbage collection, Insertion into a singly linked list, Deletion from a singly linked list; Doubly linked list, Header linked list, Circular linked list.

UNIT III 11 Hours

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

UNIT IV 12 Hours

Binary Trees: Definitions, Tree Search, Traversal of Binary Tree, Tree Sort, Building a Binary Search Tree, Height Balance: AVL Trees, Contiguous Representation of Binary Trees: Heaps, Red Black Tree: Insertion and Deletion, External Searching: B-Trees, Applications of Trees. Graphs: Mathematical Background, Computer Representation, Graph Traversal. Hashing: Hash Table ADT, understanding Hashing, Components of Hashing, Hash Table, Hash Function, Hashing Techniques, collisions, collision resolution techniques.

Text Book

1 Seymour Lipschutz, “Data Structures with C”, Schaum’s Outlines, Tata Mc Graw Hill, 2011.

2 Robert Kruse, C.L.Tondo, Bruce Leung, Shashi Mogalla,“Data Structures and Program Design using C”, Pearson Education, 2009

Reference Books

1 Mark Allen Weiss,“ Data Structures and Algorithm Analysis in C”, Second Edition, Pearson Education,2013

2 Forouzan,“A Structured Programming Approach using C”,2nd Edition, Cengage LearningIndia,2008.

Course Articulation Matrix: Mapping of Course Outcomes (COs) with Program Outcomes (POs12)

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

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

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

OBJECT ORIENTED PROGRAMMING USING JAVA

Course Outcome
CO1 - Understand object-oriented programming concepts
CO2 - Demonstrate the important feature of Object-oriented programming
CO3 - Examine event handling, String handling, and exception handling concepts
CO4 - Implement concepts to solve real-world problems

UNIT-1 [12 Hours]

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

UNIT-2 [11 Hours]

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

UNIT-3 [11 Hours]

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

UNIT-4 [11 Hours]

Exception handling: Types of Java exception – checked and unchecked exceptions; Usage of try-catch-finally blocks. Multithreading: comparison of multithreading and multitasking; Life cycle of a thread; two ways of creating thread – by extending the Thread class and by implementing the Runnable Interface, Thread synchronization. Advanced concepts: Collections in Java; Introduction to JavaBeans and Java security manager, Importance of generic programming in java with examples.

Text Book

1 E. Balagurusamy, Programming with JAVA, McGraw Hill, New Delhi, 2007

2 Herbert Schildt, Java A Beginner's Guide – Create, Compile, and Run Java Programs Today, Sixth Edition, Oracle Press, 2014

Reference Books

1 Ken Arnold, James Gosling, "The Java Programming Language, Fourth Edition, Addison Wisely, 2005

2 R Herbert Schildt, 'The Complete Reference Java, 7th Edition, McGraw Hill, 2007

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

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

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

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

OPERATING SYSTEMS

Course Outcomes

CO1 To analyze the memory management and its allocation policies

CO2 To understand synchronization and deadlock conditions and their possible solutions

CO3 To discuss the storage management policies with respect to different storage

CO4 To evaluate the concept of the operating system with respect to Linux architecture and commands

UNIT –1 [11 Hours]

Introduction: Computer System Organization, Architecture, Structure, Operations, Process Management, Memory Management, Storage Management. Operating System Structures: Services, System Calls, Types, Operating System Structure, System Boot. Processes: Process Concept, Scheduling, Operations, Inter-process Communication. Multithreaded Programming: Multithreading Models

UNIT-II [12 Hours]

Process Synchronization: The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization Examples. Process Scheduling: Criteria, Scheduling Algorithms, Multi-Processor Scheduling, Real-time CPU Scheduling. Deadlocks: System model, Characterization, Methods for handling deadlocks, Deadlock Prevention, Avoidance, Detection and Recovery from deadlock.

UNIT – III [11 Hours]

Memory Management Strategies: Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table. Virtual Memory Management: Demand Paging; Copy-on-Write, Page Replacement; Allocation of Frames, File System: File Concept, Access Methods, Directory and Disk Structure, Protection. File-system Implementation: Structure, File-System and Directory Implementation, Allocation Methods, Free Space Management. Mass-Storage Structure: Overview, Disk Scheduling, Disk Management.

UNIT – IV [11 Hours]

Introduction to Linux Programming: Linux system Architecture, Linux Command format, Linux Internal and External Commands, Directory Commands, File related commands, Disk related commands, General Utilities

Text Books

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne; “Operating Systems Concepts”, 9th Edition, 2016 India, Wiley.
2. William Stallings, "Operating Systems and Design Principles", Pearson, 5th Edition,2018

Reference Books

1. D M Dhamdhere : Operating Systems - A concept Based Approach, 3rd Edition, Tata McGraw - Hill, 2017.
2. Sumitabha Das: "UNIX Concepts and Applications", 4th Edition, Tata McGraw Hill, 2017.
3. M G Venkateshmurthy, Introduction to UNlX and Shell Programming", Pearson Edition, 2005.

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

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

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

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

DATA STRUCTURES LAB

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

1. Write a program to search for an element in an array using binary and linear search.
2. Write a program to sort list of n numbers using Bubble Sort algorithms.
3. Perform the Insertion and Selection Sort on the input {75,8,1,16,48,3,7,0} and display the output in descending order.
4. Write a program to insert the elements {61,16,8,27} into singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
5. Write a program to insert the elements {45, 34, 10, 63,3} into linear queue and delete three elements from the list. Display your list after each insertion and deletion.
6. Write a program to simulate the working of Circular queue using an array.
7. Write a program to insert the elements {61,16,8,27} into ordered singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
8. Write a program for Tower of Honoi problem using recursion.
9. Write recursive program to find GCD of 3 numbers.
10. Write a program to demonstrate working of stack using linked list.
11. Write a program to convert an infix expression x^y/(5*z)+2 to its postfix expression
12. Write a program to evaluate a postfix expression 5 3+8 2 - *.
13. Write a program to create a binary tree with the elements {18,15,40,50,30,17,41} after creation insert 45 and 19 into tree and delete 15,17 and 41 from tree. Display the tree on each insertion and deletion operation.
14. Write a program to create binary search tree with the elements {2,5,1,3,9,0,6} and perform inorder, preorder and post order traversal.
15. Write a program to Sort the following elements using heap sort {9.16,32,8,4,1,5,8,0}.

16. Given S1={“Flowers”} ; S2={“are beautiful”} I. Find the length of S1 II. Concatenate S1 and S2 III. Extract the substring “low” from S1 IV. Find “are” in S2 and replace it with “is” .
17. Write a program to implement adjacency matrix of a graph.
18. Write a program to insert/retrieve an entry into hash/ from a hash table with open addressing using linear probing.

OBJECT-ORIENTED PROGRAMMING LAB

1. Java program to display “Hello World” and display the size of all the data types.
2. Java program to implement the usage of static, local and global variables.
3. Java program to implement string operations string length, string concatenate, substring
4. Java program to find the maximum of three numbers
5. Java program to check whether the number is odd or even.
6. Java program to implement default and parameterized constructors.
7. Java program to implement an array of objects.
8. Java program to implement Single Inheritance
9. Java program to implement Multiple Inheritance using Interface
10. Java program to implement an applet
11. Java program to demonstrate a division by zero exception
12. Java program to add two integers and two float numbers. When no arguments are supplied give a default value to calculate the sum. Use method overloading.
13. Java program that demonstrates run-time polymorphism.
14. Java program to catch negative array size Exception. This exception is caused when the array is initialized to negative values.
15. Java program to handle null pointer exception and use the “finally” method to display a message to the user.
16. Java program to import user-defined packages
17. Java program to check whether a number is palindrome or not
18. Java program to find the factorial of a list of numbers reading input as command line argument.
19. Java program to display all prime numbers between two limits.
20. Java program to create a thread using Runnable Interface.

LINUX AND SHELL PROGRAMMING LAB

1. Write a shell script to print all the prime numbers between M to N (M<N).
2. Write a shell script to reverse a given number and check whether it is a palindrome.
3. Write a shell script to find the sum of digits of a given number using loops and without using loops.
4. Write a shell script to implement 10 Linux commands using case.

5. Write a Shell script that displays a list of all the files in the current directory to which the user
6. has read, write and execute permissions?
7. Write a shell script to copy a file within the current directory
8. Write a shell script to copy file between two directories

9. Write a Shell script to create two data files and compare them to display unique and common entries.
10. Write a shell script to count the number of vowels in a string.
11. Write a shell script to convert uppercase characters to lowercase and vice versa.
12. Write a shell script to accept a word and perform pattern matching in a given file.
13. Write a shell script to find the factorial of a number
14. Write a Menu-driven program to demonstrate the zombie process and orphan process.