Punyashlok Ahilyadevi Holkar Solapur University, Solapur

NAAC Accredited-2022
'B^{++}'Grade (CGPA2.96)

Name of the Faculty: Science &Technology

(As per New Education Policy 2020)

Syllabus: Bachelor of Computer Application

Name of the Course: B. C. A. I (Sem. I & II)

(Syllabus to be implemented from June 2024)

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

BCA (Bachelor of Computer Application)

Preamble: Bachelor of Computer Applications (BCA) is a four year under Graduate Programme. The Course is designed to bridge the gap between IT industries and Academic institutes by incorporating the latest development, into the Curriculum and to give students a complete understanding within a structured framework. The Course helps the students to build-up a successful Career in Computer Science and for pursuing higher studies in Computer Science.

Objectives of the Programme:

1. Demonstrate the ability to adapt to technological changes and innovations in the discipline.
2. Develop computer programs using functional programming and object-oriented programming paradigms.
3. To train students in professional skills related to Software Industry.
4. To prepare the necessary knowledge base for research and development in Computer Science.
5. To help students build-up a successful career in Computer Science and to produce entrepreneurs who can innovate and develop software products.
6. An ability to apply knowledge of mathematics, statistics and computer science in practice.
7. An ability to enhance not only a comprehensive understanding of the theory but its application too in diverse fields.
8. The program prepares the young professional for a range of computer applications, computer organization, and techniques of Computer Networking, Software Engineering, Web Development, Database management and Advance Java.
9. An ability to design a computing system to meet desired needs within realistic constraints such as safety, security and applicability in multidisciplinary teams with a positive attitude.
10. In order to enhance the programming skills of the young IT professionals, the program has introduced the concept of project development in each language/technology learned during the curriculum.

Eligibility for BCA (Bachelor of Computer Application):

The candidate passing the Higher Secondary Examination Conducted by the Maharashtra State Board of Higher Secondary Education with Arts/Commerce/Science stream or its equivalent or any Diploma of not less than two years.

Programme Outcomes (PO):

These outcomes describe what students are expected to know and can do by the time of graduation. They relate to the skills, knowledge, and behavior's that students acquire in their graduation through the program

Programme Outcomes for BCA (Bachelor of Computer Application):

The BCA (Bachelor of Computer Application) programme enables students to attain, by the time of graduation:

PO1: Design and develop software-based solutions for real life problems, serving effectively to the requirements of computer field and Society.

PO2: Attain sufficient knowledge related to computer domains, possesses technical, soft and hard skills and apply them effectively in team work.

PO3: Ability to link knowledge of Computer Science with other two chosen auxiliary disciplines of study.

PO4: Display ethical code of conduct in the usage of Internet and Cyber systems.

PO5: Ability to pursue higher studies of specialization and to take up technical employment.

PO6: Identify, formulate and analyze complex real-life problems in order to arrive at computationally viable conclusions using fundamentals of mathematics, computer sciences, management and relevant domain disciplines.

PO7: Ability to operate, manage, deploy, configure computer network, hardware, and software operation of an organization.

PO8: Apply standard Software Engineering practices and strategies in real-time software project development.

PO9: Design and develop computer programs/computer -based systems in the areas related to algorithms, networking, web design, cloud computing, IoT and data analytics.

PO10: Acquaint with the contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems

PO11: The ability to apply the knowledge and understanding noted above to the analysis of a given information handling problem.

PO12: The ability to work independently on a substantial software project and as an effective team member.

BCA - I

Subject/ Core Course		Name and Type of the Paper			Hrs./week			Total Marks Per-Paper	UA	CA	Credits
		Type	Code	Name	L	T	P				
BCA Sem-I											
Major (SELECT ANY ONE GROUP)	GROUP -I	DSC1-1	G10-0101	Programming using C-I	2	-	-	50	30	20	2
		Practical	G10-0101-P	Practical based on DSC1-1	-	-	4	50	30	20	2
		DSC2-1	G10-0102	Python-I	2	-	-	50	30	20	2
		Practical	G10-0102-P	Practical based on DSC2-1	-	-	4	50	30	20	2
		DSC3-1	G10-0103	Basics of Mathematics	2	-	-	50	30	20	2
		Practical	G10-0103-P	Practical based on DSC3-1	-	-	4	50	30	20	2
	GROUP -II	DSC1-1	G10-0101	Programming using C-I	2	-	-	50	30	20	2
		Practical	G10-0101-P	Practical based on DSC1-1	-	-	4	50	30	20	2
		DSC2-1	G10-0102	Python-I	2	-	-	50	30	20	2
		Practical	G10-0102-P	Practical based on DSC2-1	-	-	4	50	30	20	2
		DSC4-1	G10-0104	Descriptive Statistics	2	-	-	50	30	20	2
		Practical	G10-0104-P	Practical based on DSC3-1	-	-	4	50	30	20	2
Generic/ Open Elective Courses		GE1/ OE1	G10-GE-OE-101	Office Automation Tools	2	-	-	50	30	20	2
SEC/ VSC		SEC-1	G10-SEC-101	Basics Web Designing	-	-	2	50	30	20	2
AES, IKS, VEC		L1-1	ENG-101	English	2	-	-	50	30	20	2
		IKS	G10-IKS-101	To be selected from the Basket of IKS	2	-	-	50	30	20	2
		VEC-1	ICD-101	Constitution of India	2	-	-	50	30	20	2
Total					14		14	550	330	220	22
BCA Sem-II											
Major (SELECT ANY ONE GROUP)	GROUP -I	DSC1-2	G10-0201	Programming using C-II	2	-	-	50	30	20	2
		Practical	G10-0201-P	Practical based on DSC1-2	-	-	4	50	30	20	2
		DSC2-2	G10-0202	Python-II	2	-	-	50	30	20	2
		Practical	G10-0202-P	Practical based on DSC2-2	-	-	4	50	30	20	2
		DSC3-2	G10-0203	Numerical Mathematics	2	-	-	50	30	20	2
		Practical	G10-0203-P	Practical based on DSC3-2	-	-	4	50	30	20	2
	GROUP -II	DSC1-2	G10-0201	Programming using C-II	2	-	-	50	30	20	2
		Practical	G10-0201-P	Practical based on DSC1-2	-	-	4	50	30	20	2

		DSC2-2	G10-0202	Python-II	2	-	-	50	30	20	2
		Practical	G10-0202-P	Practical based on DSC2-2	-	-	4	50	30	20	2
		DSC4-2	G10-0204	Probability Theory	2	-	-	50	30	20	2
		Practical	G10-0204-P	Practical based on DSC3-2	-	-	4	50	30	20	2
Generic/ Open Elective Courses		GE2/ OE2	G10-GE-OE-201	Basics of Software Engineering	2	-	-	50	30	20	2
SEC/ VSC		SEC2	G10-SEC-201	Advanced Web Designing	-	-	2	50	30	20	2
AES, IKS, VEC		L1-2	ENG-201	English	2	-	-	50	30	20	2
		VEC-2	ENS24	Environmental studies	2	-	-	50	30	20	2
CC1		CC1	CES-201	Community Engagement & Services	2	-	-	50	30	20	2
Total					14	-	14	550	330	220	22
Grand Total					28	-	28	1100	660	440	44

Abbreviations:				
L: Lectures	**T**: Tutorials	**P**: Practical	**UA** : University Assessment	**CA** : College Assessment
Generic/ Open Electives: **GE/OE**		Skill Enhancement Courses: **SEC**		
Indian Knowledge System: **IKS**		Ability Enhancement Courses: **AES**		
Value Education Courses: **VEC**		Vocational Skill and Skill Enhancement Courses: **VSEC**		
Co-curricular Courses: **CC**				

Student contact hours per week: 24 Hours (Min.)
Total Credits for BCA (Science)-I (Semester I and II): 44
Medium of instruction: English
I. Practical Examination is the Semester wise after theory Examination. II. Duration of Practical Examination as per respective BOS guidelines. III. Separate passing is mandatory for Theory, Internal and Practical Examination.
Exit Option at Level 4.5 : Students can exit after Level 4.5 with under certificate course inComputer Programming if he/she complete the courses equivalent to a minimum of 44 credits and anadditional. 4credits core NSQF course/Internship.
Course Structure: Lectures and Practical's should be conducted as per the scheme of lectures and practical'sindicated in the course structure.

Teaching and Practical Scheme

I. Contact session for teaching 60 minutes each.

II. One Practical Batch should be of 20 students.

Assessment

I. The final practical examination will be conducted by the University appointed examiners internal as well as external at the end of the semester for each lab course and marks will be submitted to the university by the panel.

II. The practical examination will be conducted semester-wise to maintain the relevance of the respective theory course with the laboratory course.

III. The final examinations shall be conducted at the end of the semester.

Practical Examination:

I. Each paper carries 30 Marks.

II. **Duration of Practical Examination:** 2 Hrs.

III. **Nature of Question Paper:** There will be four questions of 10 Marks each. Students willattempt any two out of four questions.

IV. Certified Journal carries 5 Marks and Viva voce carries 5 Marks.

Standard of Passing:

I. Minimum 12 marks in each subject. There shall be separate passing for theory (semesterend exam and Internal) and practical also.

II. Admission to BCA (Science) Part-III is allowed only if a student has passed on all the subjects of BCA (Science) Part-I.

Board of Paper Setters /Examiners:

For each semester-end examination, there will be a board of Paper setters and examiners for everycourse. While appointing paper setters/examiners, care should be taken to see that there is at least one person specialized in each unit of the course.

Credit system implementation: As per the University norm's.

Fees Structure: As approved by the PAHS University fee fixation committee.

Intake Capacity: 60

Award of Class:

Grading: PAHS University has introduced a ten-point grading system as follows:

Sr. No.	Grade Abbreviation	From (Marks)	To (Marks)	Status	Grade Point	Description
1	O	80	100	Pass	10	Excellent / Outstanding
2	A+	70	79.99	Pass	9	Very Good
3	A	60	69.99	Pass	8	Good
4	B+	55	59.99	Pass	7	Fair
5	B	50	54.99	Pass	6	Above Average
6	C+	45	49.99	Pass	5	Average
7	C	40	44.99	Pass	4	Below Average
8	F	0	39.99	Fail	0	Fail

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	I					
Paper Category:	DSC1-1 **(Major)**					
Paper Name:	Programming using C-I		**Subject Code: G10-0101**			
Credit:	02		**Theory:**	2 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives:

1. To understand the various steps in Program development.
2. To understand the basic concepts in C Programming Language.
3. To learn how to write modular and readable C Programs
4. To learn to write programs (using structured programming approach) in 'C' to solve problems
5. To build efficient programs in "C" language essential for future programming.

Course Outcomes:

1. Able to understand the basic concepts of C programming language.
2. Enhance skill on problem solving by constructing algorithms
3. Students will be able to comprehend the general structure of C program, concepts of variable, datatype, and operator and be able to create a C program to demonstrate these concepts.
4. Able to design and develop various programming problems using C programming concepts.
5. Understand and use various constructs of the programming language such as conditionals, and iteration.
6. Demonstrate the use of strings and string-handling functions
7. Apply the skill of identifying appropriate programming constructs for problem solving.

Unit-I: Programming Methodology and Introduction to C:	**No. of Lectures:15**	**Weightage: 8-12 Marks**

Programming Methodology: Definition of Problem , Problem solving steps , Programming planning tools, Definition of Logic, Types of logic- Sequence, Selection, Iteration, Algorithm- Definition, Characteristics, Examples of algorithm., Flowchart- Definition, Characteristics , flowchart symbols, Examples converts algorithms to flowchart, Pseudo Code- Definition, characteristics, Examples

Introduction to 'C': History of 'C' language, Features of 'C' language, Application of C language, Structure of 'C' program, Compilation and execution of program. (Object File and Executable File), 'C' Fundamentals: 'C' tokens- Keywords , Identifier, Special symbols ('C' character sets), Variables, Constants, Data types- Primitive, Derived, User defined, Operators- Arithmetic, Logical,

Assignment, Relational, Bitwise, Conditional, Increment, Decrement, sizeof, comma operator, Conditional (ternary) operator, etc., Type casting or type conversion, typedef and enum., Data input and output operations: Introduction to input and output operations, Introduction to different header files and its use, stdio.h header file functions - printf( ), scanf( ), getchar( ), putchar( ), Different format specifier with their use, Different back slash (escape sequence) character constants with their use

Unit-II: Control Statements and Array:	**No. of Lectures:15**	**Weightage: 18-22 Marks**

Control Statements: Introduction to control statement, Types of control statements- Selective or Decision making –Two-way decision making- if statement with its different forms , Multiway decision making-switch statement , Iterative or looping statement -While loop , do-while loop , for loop, Unconditional branching (jump) Statement - break, continue, goto.

Arrays: Introduction to array, Types of array -One dimensional, Two dimensional, Multidimensional array, Declaration and initialization of an array, Memory allocation of an array.

Character array (string): Declaration, Operation on string, Inbuilt String functions.

Reference Books

1. Programming in ANSI-C – E. Balgurusamy, 8th edition, MCGraw Hill Publication
2. The C programming Language - Ritchie and Kernighan, 2nd Edition, Pearson
3. Programming through C Language, Dr. Tulashiram B. Pisal and Mr. Balasaheb J. Kshirsagar, InSc Publishing House(IPH)
4. Let Us C - Y.C. Kanetkar, 15th edition, BPB Publication

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	I					
Paper Category:	Practical **(Major)**					
Paper Name:	Practical based on DSC1-1		**Subject Code: G10-0101-P**			
Credit:	02		**Practical:**	4 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

1. Write a program to check a given number is prime or not
2. Write a program to check a given number is palindrome or not
3. Write a program to check a given number is perfect or not
4. Write a program to check a given number is Armstrong or not
5. Write a program to check a given number is strong or not
6. Write a program to display prime numbers between 1 to 1000
7. Write a program to display the first 100 prime numbers.
8. Write a program to find the smallest and largest element from an array
9. Write a program to search for an element in an array
10. Write a program to display diagonal elements from a given matrix.
11. Write a program to display the transpose of a given matrix.
12. Write a program to display the addition and multiplication of two matrices.
13. Write a program to count vowels from a given string
14. Write a program to find the length of a given string without using the library function.

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	I					
Paper Category:	DSC2-1 **(Major)**					
Paper Name:	Python-I		**Subject Code: G10-0102**			
Credit:	02		**Theory:**	2 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives:

1. To learn the fundamentals of python Programming
2. To learn different data structures used in Python
3. To learn different control statements used in logic development.
4. To learn the various operations on the array, list, tuple, string, set, and dictionary.

Course Outcomes:

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

1. Understand the basic concepts and applications of Python.
2. Design, create, build, and debug Python applications.
3. Explore the Integrated Development Environment (IDE).
4. Write and apply decision structures for different operations.
5. Write loop structures to perform iterative tasks.

Unit-I: Introduction to Python and Control Statements in Python:	**No. of Lectures:15**	**Weightage: 10-15 Marks**

Introduction: Features of Python, Python Virtual Machine, Memory management, Garbage Collection, Installation of Python, setting the path to operating system environment, writing the first Python program, executing a Python program.

Datatypes in Python: Datatypes-Numeric, Sequence Type-String, List, Tuple, Boolean, Set, Dictionary, Binary Types, Type conversion- implicit and explicit, Python comments, literals, constants, Identifiers, naming conventions, operators, operator precedence and associativity, input and output statements, command-line arguments.

Control Statements: if statement, if..else statement, if..elif..else statement, while loop, for loop, else suite, infinite loop, nested loops, word indentation, break statement, continue statement, pass statement, assert statement, return statement.

Unit-II: Sequence in Python, Array and Functions:	No. of Lectures:15	Weightage: 15-20 Marks

String, List, Tuple, Set and Dictionary: Creating string, manipulating different operations on string, creating list, manipulating different operations on list, list comprehensions, creating tuple, manipulating different operations on tuple, creating set, manipulating different operations on set, creating dictionary, manipulating different operations on dictionary.

Arrays in Python: Introduction, advantages of array, creating an array, types of arrays, importing array module, indexing and slicing on arrays, methods of array module.

Functions: Difference between function and method, defining a function, calling function, returning result from a function, returning multiple values from a function, functions are objects, formal and actual arguments, types of arguments, local, nonlocal and global variables, global keyword, recursive functions, anonymous functions or lambdas, using lambdas with filter(), map() and reduce() functions

Reference Books

1. Python: The Complete Reference by Martin C. Brown.
2. Core Python Programming, Dreamtech publications, by R. Nageswara Rao.
3. Python Programming, A modular approach, First Edition, Pearson, by Taneja Sheetal
4. Learning with Python, Dreamtech publications, by Allen Downey
5. Python Programming for the Absolute Beginner by Michael Dawson-Cengage Learning.

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	I					
Paper Category:	Practical **(Major)**					
Paper Name:	Practical based on DSC2-1			**Subject Code: G10-0102-P**		
Credit:	02		**Practical:**	4 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

1. Write a Python program to find the sum of a list of numbers using a for loop.
2. Write a Python program to display stars in right-angled triangular form using nested for loops.
3. Write a Python program to display a multiplication table from 1 to 10 using nested for loops.
4. Write a Python program to display numbers from 10 to 6 and break the loop when the number about to display 5.
5. Write a Python program to display numbers from 1 to 5 using the continue statement.
6. Write a Python program to find the first occurrence of substring in a given main string.
7. Write a Python program to display elements in a list in reverse order.
8. Write a Python program to accept elements in the form of a tuple and display their sum and average.
9. Write a Python program to create a dictionary with employee details and retrieve the values upon giving keys.

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	I					
Paper Category:	DSC3-1 **(Major) (Group-I)**					
Paper Name:	Basics of Mathematics		**Subject Code: G10-0103**			
Credit:	02		**Theory:**	2 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives:

1. The purpose of the course is to familiarize the prospective learners with mathematical structure that are fundamentally discrete.
2. This course introduces set theory, relation and different counting principles.
3. These concepts are useful to study or describe object or problems in computer

Course Outcomes:

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

1. To provide overview of theory of discrete objects, starting with relations and partially ordered sets.
2. To describe the fundamental counting principle and to determine the number of possible combinations for a given situation using the fundamental counting principle
3. Understand the basic principles of sets and operations in sets.
4. Prove basic set equalities.
5. Demonstrate an understanding of relations able to determine their properties.

Unit I: Basics of Matrices and Elementary Logic	**No. of Lectures:15**	**Weightage: 12-18 Marks**

Basics of Matrices: -Definition of matrix, order, types of matrices: square matrix, rectangular matrix, diagonal matrix, scalar matrix, upper triangular matrix, lower triangular matrix, symmetric matrix, skew symmetric matrix, identity matrix, row matrix, column matrix, transpose of a matrix, inverse of a matrix. Algebra of matrices: addition, subtraction, scalar multiplication, matrix multiplication.
Elementary Logic: -Prepositional Calculus: Proposition Simple statement, Compound statement, Logical connectives, Disjunction, Conjunction, Negation, Implication, Double implication, Converse, inverse and contra positive of conditional statement, truth tables, tautology, Contradiction & neither, commutative laws, associative laws, distributive laws, DeMorgan's laws, logical equivalence.

Unit II: Sets, Relations and Counting Principles	No. of Lectures:15	Weightage: 12-18 Marks

Set:- Definition: Set, Subset, power set, disjoint sets , Operations on sets : Union, Intersection ,Complement , Difference , Symmetric difference, Algebraic properties of set operations: Commutative laws , Distributive laws, Associative laws , DeMorgan's laws , Cardinality of set.

Relation: Definition of Cartesian product, Types of relation: void, universal, identity, reflexive, symmetric, transitive, equivalence, anti-symmetric, partial ordering, asymmetric, Matrix representation of relation, Graphical representation (digraph) of relation, Indegree and out degree of a vertex, Transitive closure: Warshall's algorithm

Counting principles: Cardinality of a set, Pigeonhole principle, Addition principle, Multiplication principle, Inclusive exclusive principles for two sets & three sets, Problems

Books Recommended:

1. Discrete mathematics & its applications-K. Rosen
2. Computer Oriented Numerical Methods. –Rajaraman
3. Elements of Discrete Mathematics-C.L.Liu
4. Discrete Mathematical structure for Computer Science-Alan Doerr and K.Levessuer
5. Introductory Methods of Numerical Analysis-S.S. Sastry (Prentice Hall)
6. Matrices by Shantinarayan, S. Chand & Co. NewDelhi

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	I					
Paper Category:	Practical **(Major) (Group-I)**					
Paper Name:	Practical based on DSC3-1			**Subject Code: G10-0103-P**		
Credit:	02			**Practical:**	4 Hrs./Week	
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Choose one of the following languages to solve the assignment: Python, R, or MATLAB.

1. Write a Python program to input a matrix and print its order.
2. Create and print examples of different types of matrices: square, rectangular, diagonal, scalar, upper triangular, lower triangular, symmetric, skew symmetric, identity, row, column, and zero matrices.
3. Write a Python program to find the transpose of a matrix.
4. Write a Python program to find the inverse of a matrix using NumPy.
5. Implement functions for matrix addition, subtraction, and scalar multiplication.
6. Write a function to determine the order (dimensions) of a matrix.
7. Implement functions to check if a matrix is square, rectangular, or diagonal.
8. Develop functions to compute the determinant of a square matrix.
9. Write a function to find the inverse of a square matrix (if it exists).
10. Implement matrix division (if possible, using the inverse).
11. Solve a system of linear equations using matrices.
12. Implement algorithms like Gaussian elimination or LU decomposition using matrices.
13. Implement functions or classes for logical connectives: AND, OR, NOT, IMPLICATION, etc.
14. Write functions to compute the truth value of compound statements based on truth tables.
15. Implement functions to verify logical equivalences (e.g., DeMorgan's laws, distributive laws) using truth tables or direct computation.
16. Write functions to test if a statement is a tautology, contradiction, or neither based on truth tables.
17. Create functions to find the converse, inverse, and contrapositive of conditional statements.
18. Implement functions to check logical implications and double implications.
19. Use logical operators to solve puzzles or problems that involve propositional logic (e.g., solving logic puzzles, verifying the validity of arguments).
20. Write functions to perform Row operations (e.g., swapping rows, scaling rows) Column operations (e.g., swapping columns, scaling columns)

21. Write Python functions for set operations-Union, Intersection, Difference, Symmetric Difference, Complement, Power Set generation
22. Implement functions to verify algebraic properties of set operations-Commutative laws, Associative laws, Distributive laws, DeMorgan's laws
23. Develop Python functions to calculate- Cardinality of a set, Use of the Pigeonhole principle in set theory, Solve problems involving counting principles (Addition, Multiplication, Inclusion-Exclusion)
24. Implement Python functions to define relations and perform operations- Cartesian product of two sets
25. Checking for types of relations (void, universal, identity, reflexive, symmetric, transitive, equivalence, antiequivalence, antisymmetric, partial ordering)
26. Create Python scripts to solve complex counting problems involving- Permutations and combinations, Binomial coefficients and multinomial coefficients, Applications in probability and statistics (e.g., calculating probabilities using counting techniques)

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	I					
Paper Category:	DSC4-1 **(Major) (Group-II)**					
Paper Name:	Descriptive Statistics			**Subject Code: G10-0104**		
Credit:	02		**Theory:**	2 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives:

The main objective of this course is to acquaint students with some basic concepts of Statistics. They will be introduced to some elementary statistical methods of analysis.

Course Outcomes:

1. On completion of this course, the students will be able to:
2. To prepare frequency distribution and represent it by graphically with the help of tables.
3. To compute various measures of central tendency, and dispersion and to interpret them.
4. To compute the ncorrelation coefficient and interpret its value.
5. To estimate or predict through linear regression method.

Unit I: - Population and Sample:	**No. of Lectures:18**	**Weightage: 12-18 Marks**

Introduction: Concept of Statistical Population and Sample with illustration, Methods of sampling SRSWR, SRSWOR, Stratified.

Data condensation and Graphical method: Raw data, Attribute, Variables, Discrete and Continuous Variable, General principles of classification of raw data, Construction of frequency distribution, Cumulative frequency distribution, Graphical representation of frequency distribution, Histogram, Ogives, Numerical problems.

Measures of Central Tendency: Concepts, Objects of Central Tendency, Criteria for good Measures of Central Tendency, Arithmetic Mean (A.M.)- definition, formula for computation for ungrouped and grouped data, combined A.M., effect of change of origin and scale, merits and demerits, examples.

Median: definition, formula for computation for ungrouped and grouped data, graphical method, merits and demerits, examples.

Mode: definition, formula for computation for ungrouped and grouped data, graphical method, merits and demerits, Empirical relation between mean, mode and median, Numerical Problems.

Measures of Dispersion: Concept of dispersion, Absolute and Relative measures of dispersion, Range- definition, formula for computation for ungrouped and grouped data, coefficient of range, merits and demerits, examples.

Variance and Standard Deviation (S.D.): definition, formula for, computation for ungrouped and grouped data, Coefficient of Variation (C.V.), effect of change of origin, and scale, merits and demerits, Numerical problems.

Unit-II: - Correlation and Regression	**No. of Lectures:12**	**Weightage: 12-18 Marks**

Correlation: Bivariate data, scattered diagram, Concept of correlation, types of correlation, cause and effect relation. Karl Pearson's coefficient of correlation (r), limits of r ($-1 \leq r \leq 1$), Interpretation of r, basic assumptions on which r is based, Numerical problems.

Regression for ungrouped data: Concept of regression, Lines of regression, Derivation of lines of regression by least square principle. Properties of regression coefficients, Numerical problems.

Books Recommended:

1. Fundamentals of Mathematical Statistics-Kapoor and Gupta.
2. Modern elementary Statistics –J.E.Freund
3. Statistical Methods –J.Medhi.
4. Fundamentals of Statistics-S.C.Gupta.

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	I					
Paper Category:	Practical **(Major) (Group-II)**					
Paper Name:	Practical based on DSC4-1			**Subject Code: G10-0104-P**		
Credit:	02		**Practical:**	4 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Choose one of the following languages to solve the assignment: Python / R / MATLAB

1. Write Python scripts to illustrate the concepts of statistical population and sample. Generate a random sample from a given population and calculate sample statistic (mean, median, mode, variance, etc.), then compare them with population parameters.
2. Implement Python functions to simulate different sampling methods and analyze their effectiveness using statistical metrics.
3. Develop Python functions to process and condense data. Convert raw data into variables (discrete and continuous), create frequency distributions (ungrouped and grouped), and calculate cumulative frequency distributions.
4. Develop Python scripts to process raw data into frequency distributions and plot histograms and ogives. Analyze graphical representations to interpret data trends.
5. Solve numerical problems related to data condensation and graphical methods using Python. Compute frequencies, and cumulative frequencies, and plot histograms for given datasets. Analyze and interpret graphical representations to draw conclusions about data distributions.
6. Write Python functions to compute measures of central tendency:
    I. Arithmetic Mean (AM) for ungrouped and grouped data
    II. Median for ungrouped and grouped data using formulas and graphical methods
    III. Mode for ungrouped and grouped data, including handling multimodal distributions
7. Implement Python functions to explore empirical relations between mean, median, and mode. Calculate and compare these measures for various datasets to observe trends and relationships.
8. Develop Python functions to compute measures of dispersion:
    I. Range for ungrouped and grouped data
    II. Variance and Standard Deviation (SD) for ungrouped and grouped data
    III. Coefficient of Variation (CV) to compare variability across different datasets
9. Write Python scripts to compute relative measures of dispersion:
    I. Coefficient of Range and Coefficient of Variation for datasets with different scales and origins
    II. Discuss the impact of scale and origin changes on these measures

10. Solve numerical problems involving measures of dispersion using Python. Compute and compare ranges, variances, standard deviations, and coefficients of variation for given datasets. Interpret results to conclude data variability and distribution
11. Implement Python functions to calculate and classify types of correlation: positive, negative, and zero correlation.
12. Develop Python functions to compute Karl Pearson's coefficient of correlation. Calculate the correlation coefficient (r) for given datasets and validate calculations against theoretical limits ($-1 \leq r \leq 1$).
13. Write Python scripts to verify basic assumptions underlying the correlation coefficient (linearity, normality, etc.). Solve numerical problems involving correlation coefficient calculations and interpretation.
14. Implement Python functions to introduce regression analysis. Explain the concept of regression and its applications. Calculate regression lines using the method of least squares.
15. Develop Python scripts to perform regression analysis for given datasets. Interpret regression results.

BCA (Science)-I, Level - 4.5 UG Certificate Level	
Sem:	I
Paper Category:	GE1/OE1
Paper Name:	Office Automation Tools **Subject Code: G10-GE-OE-101**

Credit:	02		**Theory:**	2 Hrs./Week	
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives:

1. To provide an in-depth training in use of office automation, internet and internet tools. The course also helps the candidates to get acquainted with IT.
2. To help the students to understand how to format, edit, and print text documents and prepare for desktop publishing.
3. To create various documents newsletters, brochures, making document using photographs, charts, presentation, documents, drawings and other graphic images.
4. To work with the worksheet and presentation software..

Course Outcomes:

1. At the end of this course, the student should be able to
2. Integrate both graphs and tables created in Microsoft Excel into a laboratory report in Microsoft Word.
3. Generate equations, sample calculations, and basic diagrams in Microsoft Word.
4. Input experimental data into Microsoft Excel.
5. Perform calculations in Microsoft Excel using both manually inputting formulas and built-in Functions.
6. Generate simple and effective tables and graphs to describe experimental data in Microsoft Excel.
7. Properly format and organize a formal laboratory report in Microsoft Word.

Unit-I: Introduction to Computer & MS-Word :	**No. of Lectures:15**	**Weightage: 8-12 Marks**

Introduction to Computer: Applications of Computer, Advantages of Computer, Characteristics of Computer, Hardware and Software, Block diagram of computer

MS Word: Working with Documents -Opening and Saving files, Editing text documents, Inserting, Deleting, Cut, Copy, Paste, Undo, Redo, Find, Search, Replace, Formatting page and setting Margins, Converting files to different formats, Importing and Exporting documents, Sending files to others, Using Tool bars, Ruler, Using Icons, using help., **Formatting Documents:** Setting Font styles, Font selection- style, size, color etc, Type face - Bold, Italic, Underline, Case settings, Highlighting, Special symbols, Setting Paragraph style, Alignments, Indents, Line Space, Margins, Bullets and Numbering.. **Setting Page style:** Formatting Page, Page tab, Margins, Layout settings, Paper tray, Border and Shading, Columns, Header and footer, Setting Footnotes and end notes – Shortcut Keys;

Inserting manual page break, Column break and line break, Creating sections and frames, Anchoring and Wrapping, Setting Document styles, Table of Contents, Index, Page Numbering, date and Time, Author etc., Creating Master Documents, Web page. Creating Tables: Table settings, Borders, Alignments, Insertion, deletion, Merging, Splitting, Sorting, and formula. , Drawing: Inserting Clip Arts, Pictures/Files etc., **Tools:** Word Completion, Spell Checks, Mail merge, Templates, Creating contents for books, Creating Letter/Faxes, Creating Web pages, Using Wizards, Tracking Changes, Security, Digital Signature. Printing Documents – Shortcut keys.

Unit-II: MS Excel and MS PowerPoint:	**No. of Lectures:15**	**Weightage: 18-22 Marks**

MS Excel: Spread Sheet and its Applications, Opening Spreadsheet, Menus - main menu, Formula, Editing, Formatting, Toolbars, Using Icons, Using help, Shortcuts, Spreadsheet types. Working with Spreadsheets- opening, Saving files, setting Margins, Converting files to different formats (importing, exporting, sending files to others), Spreadsheet addressing - Rows, Columns and Cells, Referring Cells and Selecting Cells – Shortcut Keys., **Entering and Deleting Data:** Entering data, Cut, Copy, Paste, Undo, Redo, Filling Continuous rows, columns, Highlighting values, Find, Search and replace, Inserting Data, Insert Cells, Column, rows and sheets, Symbols, Data from external files, Frames, Clipart, Pictures, Files etc, Inserting Functions, Manual breaks., **Setting Formula:** Finding the total in a column or row, Mathematical operations (Addition, Subtraction, Multiplication, Division, Exponentiation), using other Formulae., **Formatting Spreadsheets:** Labeling columns and rows, Formatting- Cell, row, column and Sheet, Category- Alignment, Font, Border and Shading, Hiding/ Locking Cells, Anchoring objects, Formatting layout for Graphics, Clipart etc., Worksheet Row and Column Headers, Sheet Name, Row height and Column width, Visibility - Row, Column, Sheet, Security, Sheet Formatting and style, Sheet background, Colour etc, Borders and Shading ,Shortcut keys., **Working with sheets:** Sorting, Filtering, Validation, Consolidation, and Subtotal., **Creating Charts:** Drawing. Printing. Using Tools – Error checking, Formula Auditing, Creating and Using Templates, Pivot Tables, Tracking Changes, Security, Customization.

MS Powerpoint: Presentation – Opening new presentation, Different presentation templates, setting backgrounds, selecting presentation layouts., **Creating a presentation:** Setting the Presentation style and adding text to the Presentation. Formatting a Presentation: Adding style, Colour, gradient fills, Arranging objects, Adding Header and Footer, Slide Background, and Slide layout. Adding Graphics to the Presentation- Inserting pictures, movies, tables etc into the presentation, Drawing Pictures using draw., **Adding Effects to the Presentation:** Setting Animation and transition effect. Printing Handouts and generating Standalone Presentation viewer.

Reference Books:

1. Information Technology in Business: Principles, Practices, and Opportunities by James A Senn, Prentice Hall.
2. Technology and Procedures for Administrative Professionals by Patsy Fulton-Calkins, Thomson Learning.
3. Computer Fundamental MS Office-Including Internet and Web Technology: Anupama Jain, Avneet Mehra
4. The Complete Reference: Virginia Andersen, McGraw Hill
5. MS Office 2007 in a Nutshell: S. Saxena, Vikas Publications
6. MS-Office 2007 Training Guide: S. Jain, BPB Publications
7. Learning Computer Fundamentals, MS Office and Internet and Web Technology: D. Mai dasani. Reading, Vols. 1 and 2. Macmillan, 1975, Bhasker, W. W. S. and Prabhu, N. S.

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	I					
Paper Category:	SEC1					
Paper Name:	Basics Web Designing		**Subject Code: G10-SEC-101**			
Credit:	02		**Practical:**	4 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objective:

1. Give the distinguishing characteristics of scripting language.
2. Discuss the reasons for and effects of nonstandard client-side scripting language characteristics, such as limited data types, dynamic variable types and properties, and extensive use of automatic type conversion.
3. Develop event-driven programs that use HTML intrinsic event attributes, DOM events, listeners, and DOM-generated events.
4. Use the DOM to modify a document's attributes and style properties as well as to modify its parse-tree representation.

Course Outcomes:

1. Explain the history of the internet and related internet concepts that are vital in understanding web development.
2. Discuss the insights of internet programming and implement complete applicationsover the web.
3. Demonstrate the important HTML tags for designing static pages and separate designfrom content using the Cascading Style sheet.
4. Utilize the concepts of JavaScript.

List of Assignment:

1.	Write Html code to display “Welcome in Web Technology” message.
2.	Write Html code to display different heading levels.
3.	Write Html code to subscript and superscript.
4.	Write Html code to implement all physical tags.
5.	Write Html code to implement all logical tags.
6.	Write Html code to implement <pre> tag.
7.	Write Html code to implement <font> tags.
8.	Write Html code to implement <address> tags.
9.	Write Html code to implement <meta> tags.

10.	Write Html code to implement lists.
11.	Write Html code to implement hyperlink tag with target attribute.
12.	Write Html code to implement link, alink and vlink attributes.
13.	Write Html code to implement image as hyperlink.
14.	Write Html code to implement image map.
15.	Write Html code for display University examination time table.
16.	Write Html code for display Railway time table.
17.	Write Html code to collect student information.
18.	Write CSS code implement ISS.
19.	Write CSS code implement ESS.
20.	Write CSS code implement lists.
21.	Write CSS code implement class and ids.
22.	Write CSS code implement links.
23.	Write CSS code implement padding.
24.	Write CSS code implement background and border.
25.	Write CSS code implement 2D.
26.	Write CSS code implement 3D.
27.	Write CSS code implement animation.
28.	Write JavaScript code for addition of any two numbers.
29.	Write JavaScript code for subtraction of any two numbers.
30.	Write JavaScript code for multiplication of any two numbers.
31.	Write JavaScript code for division of any two numbers.
32.	Write JavaScript code to calculate simple interest.
33.	Write JavaScript code to calculate area and perimeter of circle.
34.	Write JavaScript code to calculate area and perimeter of rectangle.
35.	Write JavaScript code to calculate area and perimeter of triangle.
36.	Write JavaScript code to calculate area and perimeter of square.
37.	Write JavaScript code to exchange value of any two numbers.
38.	Write JavaScript code to exchange value of any two numbers (without using third variable).
39.	Write JavaScript code to implement alert.
40.	Write JavaScript code to implement confirm.
41.	Write JavaScript code to implement eval.
42.	Write JavaScript code to find out given number is even or odd.

43.	Write JavaScript code to find out given number is positive or negative.
44.	Write JavaScript code to find out maximum number between any two numbers.
45.	Write JavaScript code to calculate factorial of any number.
46.	Write JavaScript code to calculate digit sum of any number.
47.	Write JavaScript code to calculate face value of any number.
48.	Write JavaScript code to find out given number is prime or not.
49.	Write JavaScript code to find out given number is Armstrong or not.
50.	Write JavaScript code to find out given number is palindrome or not.
51.	Write JavaScript code to find out given number is perfect or not.
52.	Write JavaScript code to find out given number is strong or not.
53.	Write JavaScript code to find out given number is perfect or not.
54.	Write JavaScript code to find out given number is strong or not.
55.	Write JavaScript code for function without argument and without return value.
56.	Write JavaScript code for function without argument and with return value.
57.	Write JavaScript code for function with argument and without return value.
58.	Write JavaScript code for function with argument and return value.
59.	Write JavaScript code calculate string length.
60.	Write JavaScript code to working form.
61.	Write JavaScript code for validation handling.
62.	Write JavaScript code for working with array.

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	II					
Paper Category:	DSC1-2 **(Major)**					
Paper Name:	Programming using C-II			**Subject Code: G10-0201**		
Credit:	02		**Theory:**	2 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objective:

1. To develop logic which will help them to create programs, and applications in the 'C' language.
2. To learn the basic programming constructs they can easily switch over to any other language in the future.
3. To learn problem-solving techniques using C.
4. To train the student in the basic concepts of the programming language C.
5. To improve the programming skills using C.

Course Outcomes:

Upon successful completion of this course, students will be able to-

1. To implement advanced C programming concepts like function, pointer, structure and union etc.
2. To understand the dynamics of memory by the use of pointers.
3. To understand file handling using C Programming language.
4. To understand the concept of macros and preprocessors.

Unit-I: Functions and Pointers:	**No. of Lectures:20**	**Weightage: 12-18 Marks**

Functions: Introduction to function, Need or use of function, Types of Functions, Inbuilt/ Predefined/ Library functions, User defined function, Steps to add or include user defined function in program- Function declaration (Prototyping) ,Function calling , Function definition (Function Implementation), **Types of Function:** Function with argument without return value, Function with argument with return value, Function without argument with return value, Function without argument without return value, local and global variable, Recursion and its Advantages and disadvantages.

Pointers: Definition and declaration of pointer, Pointer initialization, Pointer and function , Pointer and array, Chain of pointer, Call by value, Call by reference , Dynamic memory allocation, malloc(), calloc(), realloc() and free(), **Storage Classes:** Introduction to storage Classes and use, storage classes- auto, extern, static, register

Unit-II: Structures and Union and File Handling:	No. of Lectures:10	Weightage: 12-18 Marks

Structures and Union: Definition and declaration of structure and union, difference between structure and union, Array of structures, Passing structure to function, Pointer to structure, Nested structure, self-referential structure., **File Handling:** Introduction, Standard input- get char(), getch(), getche(), Standard output- putchar(), putch(), putche(), Formatted input- scanf(), sscanf(), fsclose(), File modes, Text and binary mode., **Macros and Preprocessing:** Features of C pre-processor, Macro – Declaration, Expansion, File Inclusion

Reference Books

1. Programming in ANSI C, E. Balagurusamy, McGraw Hill Education India Private Limited.
2. Programming through C Language, Dr. Tulashiram B. Pisal and Mr. Balasaheb J. Kshirsagar, InScPublishing House (IPH), Edition-1st, 2021.
3. The ‘C’ programming language, Brian Kernighan, Dennis Ritchie, Pearson.

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	II					
Paper Category:	Practical **(Major)**					
Paper Name:	Practical based on DSC1-2 **Subject Code: G10-0201-P**					
Credit:	02		**Practical:**	4 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

1. WAP to find out the factorial of any number.
2. Write a program that swaps two numbers using pointers.
3. Write a program in which a function is passed the address of two variables and then alters its contents.
4. Write a program that takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle, and displays the value of the area and circumference from the main() function.
5. Write a program that counts the total number of vowels present in the string.
6. Write a program to find the sum of n elements entered by the user.
7. Write a program to allocate memory dynamically using malloc() and calloc().
8. Write a program to illustrate the difference between structure and union.
9. WAP to pass an array of structure to function.
10. Write a program to copy the content of one file into another file.
11. Write a program to display the content of binary files.
12. Write a program to accept integer numbers in file, find even and odd numbers between them store even numbers into even file and odd number into odd file and display the content of files.

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	II					
Paper Category:	DSC2-2 **(Major)**					
Paper Name:	Python-II			**Subject Code: G10-0202**		
Credit:	02		**Theory:**	2 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objective:

1. To learn the use of functions in programming.
2. To understand the use of modules and packages in the application hierarchy.
3. To understand Python programming using object-oriented programming principles.
4. To learn handling of various exceptions during the application development.
5. To understand the working with different file operations.

Course Outcomes:

1. Upon successful completion of this course, students will be able to
2. Write and implement a functional and modular approach to application development.
3. Design an application using an object-oriented paradigm.
4. Create error-free applications by applying the exception-handling concept.
5. Design an application that contains the use of different files for data processing.

Unit-I: Object Oriented programming:	**No. of Lectures:15**	**Weightage: 10-14 Marks**

Modules and packages: Introduction to modules and packages, import statement, from…import statement, creating our own modules, working with built-in modules- Math module, time module and random module.

Python Object Oriented: Difference between procedure-oriented and object-oriented programming. Features of object-oriented programming- classes and objects, inheritance, polymorphism, encapsulation, abstraction. Creating class, self-variable, constructor, types of variables, namespaces, types of methods, passing member of one class to another class, inner classes. Types of inheritance, super() method, method overloading, method overriding, abstract classes, and interfaces.

Unit-II: Threading, Exception Handling and File:	No. of Lectures:15	Weightage: 16-20 Marks

Threading: Understanding threads, Class and threads, Creating Threads, Thread Synchronization, Treads Life cycle, Multi-threading.

Exception Handling: Error in Python program, exceptions, steps in exception handling using try, except, else and finally blocks, types of exceptions- built-in and user-defined exceptions, assert statement.

File Input Output: Types of files in Python, opening a file- the file opening modes, closing a file, working with text files containing strings, working with binary files, with statement, pickling and unpickling, seek() and tell() methods, random accessing of binary files, zipping and unzipping files, working with directories.

Reference Books

1. Python: The Complete Reference by Martin C. Brown.
2. Core Python Programming, Dreamtech publications, by R. Nageswara Rao.
3. Python Programming, A modular approach, First Edition, Pearson, by Taneja Sheetal

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	II					
Paper Category:	Practical **(Major)**					
Paper Name:	Practical based on DSC2-2		**Subject Code: G10-0202-P**			
Credit:	02		**Practical:**	4 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

1. Write a function to return the addition and subtraction of two numbers using a function return two values.
2. Write a python program to demonstrate the different methods of array module.
3. Write a python program to demonstrate the types of array.
4. Write a python program to understand the positional arguments of a function
5. Write a python program to understand the keyword arguments of a function
6. Write a python program to understand the default arguments in a function
7. Write a python program to understand Variable length arguments in a function.
8. Write a python program to understand Anonymous (lambda) Function.
9. Write a python program to understand local, non-local and global variables.
10. Write a python program to create a module and import it.
11. Write a python program to create a package and import it.
12. Write a python program to demonstrate the instance method, class method and static method.
13. Write a python program to demonstrate inner classes.
14. Write a python program to demonstrate Constructors in Inheritance.
15. Write a python program to demonstrate method overloading.
16. Write a python program to demonstrate method overriding.
17. Write a python program to read all the strings from the text file and display them.
18. Write a python program to append data to an existing file and display them.
19. Write a python program to count a number of lines, words and characters in a file.
20. Write a python program to copy an image file into another file.
21. Write a python program to apply different manipulation operations of directories.
22. Write a python program to handle the ZeroDivisionError exception.
23. Write a python program to handle syntax errors given by eval( ) function.
24. Write a python program to handle IOError produced by open( ) function.
25. Write a python program to illustrate the use of raising an exc

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	II					
Paper Category:	DSC3-2 **(Major) (Group-I)**					
Paper Name:	Numerical Mathematics			**Subject Code: G10-0203**		
Credit:	02		**Theory:**	2 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives:

1. The course is designed to have a grasp of important concepts of Numerical Methods in a scientific way.
2. The learner is expected to solve as many examples as possible to get complete clarity and understanding of the topics covered.

Course Outcomes:

At the end of this course, the student should be able to

1. Ability to appreciate real-world applications which use these concepts.
2. Skill to formulate a problem through Mathematical Modeling and programming.

Unit I: - Polynomial Interpolation Approximation and Errors	**No. of Lectures:18**	**Weightage: 12-18 Marks**

Interpolation and Extrapolation: Argument and entries, equally spaced data and not equally spaced data, Finite difference operators: forward difference operator, backward difference operator, divided difference operator, Relation between these operators, Interpolation and Extrapolation, Newton's forward difference interpolation formula, Newton's backward difference interpolation formula, Lagrange's interpolation formula, problems, Newton's divided difference interpolation formula (only formula without proof), problems.

Errors: Absolute error, relative error and percentage error

Normalized Floating point representation of real numbers, arithmetic operations on the numbers in normalized floating-point notation: addition, subtraction, multiplication and division.

Unit II:- Numerical Integration and Ordinary Differential Equation:	No. of Lectures:12	Weightage: 12-18 Marks

Numerical Integration General quadrature formula for equidistant ordinates (without proof), Trapezoidal rule, Simpson's 1/3 rd rule and Simpson's 3/8 th rule, derivation of rules and examples Ordinary Differential Equation: Degree and order of a differential equation, Definition of an ordinary differential equation, Picard's Methods, Taylor Series Method, Euler's method, Runge-Kutta second order method, Runge-Kutta fourth order method, examples.

Reference Books:

1. Introduction to Numerical Analysis by S. S. Sastri, Tata McGraw Hill
2. Computers and Numerical Methods by Balguruswamy, (TMH).
3. Numerical Methods for Scientific and Engineering Computation M. K. Jain, S. R. K. Iyengar and R. K. Jain, New Age International Publisher
4. Applied Numerical Methods with MATLAB for Engineers and Scientists Steven C Chapra, Tata McGraw Hill, 2/e (2010).

B.C.A. -I, Level - 4.5 UG Certificate Level						
Sem:	II					
Paper Category:	Practical **(Major) (Group - I)**					
Paper Name:	Practical based on DSC3-2		**Subject Code: G10-0203-P**			
Credit:	02		**Practical:**	4 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Choose one of the following languages to solve the assignment: Python/R/MATLAB.

1. Write a Python program that implements Newton's forward and backward difference interpolation formula to interpolate a value at a given point using equally spaced data points.
2. Write a Python program that implements Newton's forward and backward difference interpolation formula to interpolate a value at a given point using not equally spaced data points.
3. Create a Python function that performs Lagrange interpolation to find the value of the function at a given point using distinct data points.
4. Implement a Python function to compute the forward and backward difference operator for a given set of function values and step size.
5. Implement Newton's divided difference interpolation formula in Python, focusing on calculating the divided differences for a set of given data points.
6. Create a Python function that calculates the absolute error, relative error and percentage error between a true value and an approximate value.
7. Write a Python program that performs addition, subtraction, multiplication, and division of two floating-point numbers in normalized form, handling overflow and underflow conditions.
8. Implement a Python function to compute the integral using the general quadrature formula for equidistant ordinates. Handle both open and closed formulas based on user input.
9. Write a Python program to numerically integrate a function using the Trapezoidal rule. Include examples demonstrating the accuracy improvement with finer partitions.
10. Develop a Python function to apply Simpson’s 1/3 rule for numerical integration of a function. Test the function with various examples to illustrate its accuracy.
11. Implement Simpson’s 3/8 rule in Python to compute the integral of a function. Compare its performance with Simpson’s 1/3 rule for different functions and step sizes.
12. Create a Python function that determines the degree and order of a given ordinary differential equation (ODE) represented symbolically.
13. Write a Python program to solve a first-order ODE using Picard’s method. Implement it for a specific example and compare with analytical solutions if available.

14. Develop a Python function that applies the Taylor series method to solve a first-order ODE numerically. Include examples to demonstrate convergence and accuracy.
15. Implement Euler's method in Python for solving a first-order ODE numerically. Show how the error evolves with step size by comparing with analytical solutions.
16. Write a Python program to solve a first-order ODE using the Runge-Kutta second-order method. Compare its accuracy and stability with Euler's method.
17. Implement the Runge-Kutta fourth-order method in Python to solve a first-order ODE. Evaluate its performance in terms of accuracy and computational efficiency.

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	II					
Paper Category:	DSC4-2 **(Major) (Group-II)**					
Paper Name:	Probability Theory			**Subject Code: G10-0204**		
Credit:	02		**Theory:**	2 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives:

The main objective of this course is to acquaint students with some basic concepts of Probability and probability distributions.

Course Outcomes:

At the end of this course, the student should be able to

1. To distinguish between random and nonrandom experiments.
2. To find the probabilities of the events.
3. To apply discrete and continuous probability distributions studied in this course in different situations.

Unit-I: - Permutations, Combinations, and Probability:	**No. of Lectures:15**	**Weightage: 12-18 Marks**

Permutations and Combinations: Principles of counting, Permutations of n dissimilar objects taken r objects at a time (with & without repetitions), Permutations of n objects not all similar - Combinations of n objects taken r objects at a time, Combinations with restriction on selection (excluding or including a particular object in the group), Numerical problems.

Probability: Deterministic and non-deterministic (Random) experiment- Sample space (finite, infinite, countable),

Events-Types of events, Probability –Classical definition, axioms of probability (Axiomatic Definition of Probability), probability of an event, Theorems of probability (with proof)-

i) $0 \leq P(A) \leq 1$,

ii) $P(A) + P(A') = 1$,

iii) $P(\Phi) = 0$

iv) $P(A) \leq P(B)$ when A is subset of B

v) Addition law of probability (Statement only).

Concept & definition of conditional probability, multiplication law of probability (Statement only), Concept & definition of independence of two events, Numerical problems.

Unit-II: - Random Variables and Probability Distributions:	No. of Lectures:15	Weightage: 12-18 Marks

Discrete Random Variable: -Definition of r.v., discrete r.v., Definition of probability mass function (p.m.f.), cumulative distribution function (c.d.f.) & properties of c.d.f., Definition of mathematical expectation & variance, theorems on expectation, Numerical problems.

Standard Discrete Distribution: -Binomial distribution-Definition, mean, variance (statement only), illustration of real-life situations, additive property (statement only). Poisson distribution-Definition-mean, variance (statement only), illustration of real-life situations, additive property (Statement only), Numerical Problems.

Continuous Random Variable: -Definition-continuous r.v., probability density function(p.d.f.), cumulative distribution function (c.d.f.), statement of properties of c.d.f., Definition of mean &variance, Numerical problems.

Standard Continuous Distribution: -Uniform Distributions-Definition-mean, variance (Statement only), Numerical Problems Normal Distribution-Definition, identification of parameters, nature of probability curve, standard normal variate (s.n.v.), properties of normal distribution, distribution of aX+b, aX+bY+c when X & Y are independent, Numerical Problems.

Books Recommended:

1. Fundamentals of Mathematical Statistics-Kapoor& Gupta.
2. Modern elementary Statistics –J.E.Freund
3. Statistical Methods –J.Medhi.
4. Fundamentals of Statistics-S.C.Gupta.
5. Fundamentals of applied Statistics-Gupta &Kapoor.
6. Business Statistics –S. Shah

B.C.A. -I, Level - 4.5 UG Certificate Level						
Sem:	II					
Paper Category:	Practical **(Major) (Group - II)**					
Paper Name:	Practical based on DSC4-2		**Subject Code: G10-0204-P**			
Credit:	02		**Practical:**	4 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Choose one of the following languages to solve the assignment: Python, R, or MATLAB.

1. Write Python functions to demonstrate principles of counting permutations and combinations using recursive and iterative methods.
2. Implement Python scripts to calculate permutations n dissimilar objects taken r at a time (with and without repetitions) also Handle cases where not all objects are distinct.
3. Develop Python functions to compute combinations of n objects taken r at a time and also implement combinations with restrictions (e.g., excluding or including specific objects).
4. Write Python scripts to define and manipulate sample spaces which Represent finite, infinite, and countable sample spaces and Classify types of events (simple, compound, mutually exclusive, exhaustive).
5. Implement Python functions to compute probabilities using different definitions- Classical definition for equally likely outcomes and Axiomatic definition with theorems ($0 \leq P(A) \leq 1$, $P(A) + P(A') = 1$, $P(\Phi) = 0$, $P(A) \leq P(B)$ when A is subset of B).
6. Develop Python functions to calculate conditional probabilities and Implement Bayes' theorem
7. Develop Python functions to Calculate probabilities of intersections and unions of events.
8. Create Python functions to solve complex permutation and combination problems. Test with scenarios involving constraints and repetitions, and discuss computational efficiency.
9. Implement Python functions to determine independence between events based on theoretical calculations and simulations. Discuss scenarios where events are independent or dependent.
10. Write Python programs to Define discrete random variables (r.v.) and their properties also implement probability mass function (p.m.f.) and cumulative distribution function (c.d.f.) and verify properties of c.d.f. such as non-decreasing behavior and limits at infinity.

11. Develop Python functions to Calculate mathematical expectation (mean) and variance of discrete random variables and Apply theorems related to expectation (e.g., linearity of expectation) to numerical problems.

12. Implement Python scripts to Define and simulate Binomial distributions and also compute mean and variance of Binomial distributions.

13. Write Python programs to define and simulate Poisson distributions also Calculate mean and variance of Poisson distributions.

14. Develop Python functions to define continuous random variables (r.v.) and their probability density function (p.d.f.) and implement cumulative distribution function (c.d.f.) and verify its properties.

15. Write Python scripts to Compute mean and variance of continuous random variables and Validate properties of c.d.f. including non-decreasing behavior and limits.

16. Implement Python functions to define Uniform distributions and calculate mean and variance and solve numerical problems using Uniform distributions.

17. Write Python scripts to model Normal distributions. Calculate probabilities using Z-scores, transformations of variables, and illustrate properties of Normal distributions with numerical examples.

18. Design a comprehensive python function that integrates concepts from discrete and continuous random variables, probability distributions, and applications. Use Python to model, simulate, and analyze scenarios involving complex distributions and their implications.

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	II					
Paper Category:	GE2/OE2					
Paper Name:	Basics of Software Engineering			**Subject Code: G10-GE-OE-201**		
Credit:	02		**Theory:**	2 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives –

The aim of this course is to prepare learners for a foundational understanding of computers, encompassing:

1. To understand the fundamental concepts and characteristics of systems.
2. To categorize different types of systems and understand system analysis.
3. To explore various SDLC models and their applications with different functional and non-functional requirements.
4. To learn various techniques for gathering system requirements.
5. To understand the process of designing and implementing a system.
6. To understand coding standards, size measures, complexity analysis, and verification.
7. To grasp the fundamentals of software testing.
8. To learn about software implementation and the maintenance process.

Course Outcomes-

1. Students will be able to define a system, identify its characteristics, and describe various types of systems.
2. Students will be able to describe and compare models such as the Waterfall model, V-shape model, Spiral model, Prototyping, Incremental, RAD, and Agile methodologies.
3. Students will be able to identify, document, and analyze user and system requirements.
4. Students will be proficient in conducting interviews, questionnaires, record reviews, and observations for requirement gathering.
5. Students will be able to design data flow diagrams, entity-relationship diagrams, structured charts, and create a data dictionary. They will also learn input and output design.

6. Students will learn to write clean, maintainable code, estimate effort and cost, and verify code correctness.
7. Students will be able to perform different types of testing and understand testing methodologies.
8. Students will understand the steps involved in implementing software and the different types of software maintenance.

Unit I: - **Basics to Software Engineering**	**No. of Lectures:15**	**Weightage: 12-18 Marks**

System concepts: Introduction system, characteristics, Elements of system, Types of system, System Analysis, Role of System Analyst, Software Engineering: Definition, Characteristics of software, Qualities of software. System Development life cycle: Waterfall model, V-shape model, Spiral model, Prototyping, incremental, RAD, Agile. Software requirements: Functional, Non-functional requirements, User requirements, System requirements, Fact-finding techniques: Interviews, Questionnaires, Record reviews, Observation Analysis and Design Tools: Flowcharting, Decision tables, Decision Trees, Structured English, Structure charting Techniques.

Unit II: - **Software Design, Implementation and Maintenance**	**No. of Lectures:15**	**Weightage: 12-18 Marks**

Data flow Diagram (Physical, Logical), Entity relation diagram, structured chart, Data Dictionary, Input and output design

Coding: Verification, size measures, complexity analysis, coding standards, Effort Estimation, Cost Estimation, Testing and its types, need

Construction of the system: traditional and incremental approaches, conversion methods, Software Implementation, Overview of the maintenance process, and types of maintenance.

Reference Books-

1. Analysis and Design of Information Systems By James Senn.
2. Practical guide to structure System Design By Miller/Page/jones.
3. Software Engineering By Pressman.

4. System Analysis and Design By Parthsarty

BCA (Science)-I, Level - 4.5 UG Certificate Level						
Sem:	II					
Paper Category:	SEC2					
Paper Name:	Advanced Web Designing			**Subject Code: G10-SEC-201**		
Credit:	02		**Practical:**	4 Hrs./Week		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objective:

1. To identify the capabilities of JavaScript and jQuery and their role in web design and thedocument object model.
2. To respond to user events using jQuery, creating interactivity.
3. To provide a collection of syntax for template designs.
4. To develop and apply appropriate website or web application information architectures.
5. To design effective user interfaces.

List of Assignments:

1.	Hide and Show Elements.
2.	Change CSS Dynamically
3.	Fade In and Fade Out
4.	Slide Up and Slide Down
5.	Animate an Element
6.	Form Validation
7.	Append and Remove Elements
8.	Handle Events
9.	AJAX Request
10.	Chaining Methods
11.	Basic Grid Layout
12.	Basic Grid Layout
13.	Cards
14.	Forms
15.	Buttons
16.	Modals
17.	Alerts

18.	Carousel
19.	Tooltips
20.	Progress Bar
21.	Tables
22.	List Group
23.	Create a Website with Bootstrap Grid

Equivalent Subject for Old Syllabus B.C.A. (Science) - I (Semester–I and II) (NEP-2020)

Semester-I		
Sr. No.	**Name of the Old Paper (w.e.f. 2022-2023)**	**Name of the New Paper (w.e.f. 2024-2025)**
1.	Fundamental of Computer	No Equivalence
2.	Office Automation	Office Automation Tools
3.	Programming and Problem Solving using 'C' – I	Programming using 'C'-I
4.	Web Programming-I	Basics Web Designing
5.	Basics of Mathematics	Basics of Mathematics
6.	Descriptive Statistics	Descriptive Statistics
7.	Fundamentals of Electronics	No Equivalence
8.	Linear Electronics	No Equivalence
Semester-II		
1.	Introduction to Python Programming	Python-I (Sem-I)
2.	Operating System	No Equivalence
3.	Programming and Problem Solving using 'C' – II	Programming using 'C'-II
4.	Web Programming-II	No Equivalence
5.	Graph Theory	No Equivalence
6.	Probability Theory	Probability Theory
7.	Digital Electronics	No Equivalence
8.	Introduction of Microprocessor and Interfacing	No Equivalence