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

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Faculty of Science & Technology

Choice Based Credit System (CBCS)(w.e.f.2023-24)

Revised Structure for BCA-II

Subject/ Core Course	Name and Type of the Paper			Paper No.	No. of papers/ Practical (Hrs./week)			Total Marks Per Paper	UA	CA	Credits
	Type		Name		L	T	P				
Class :			**BCA - II Semester -III**								
****Core Courses DSC 1C and DSC 2C are compulsory while student can select either # DSC 3C/GE-3C or AIC-1A as a third subject.	DSC 1C	AIC-1A	Object Oriented Programming using Java-I	Paper-V	3	--	--	50	40	10	4.0
			Data Structure Using 'C'-I	Paper-VI	3	--	--	50	40	10	
	DSC 2C		Database Management System-I	Paper-V	3	--	--	50	40	10	4.0
			Software Engineering	Paper-VI	3	--	--	50	40	10	
	# DSC 3C/GE-3C		FA Using Tally	Paper-V	3	--	--	50	40	10	4.0
			Operating System-II	Paper-VI	3	--	--	50	40	10	
Total Sem-III					**18**	--	--	**300**	**240**	**60**	**12**
	$SEC-1		Web Technology Using PHP		4	--		100	80	20	4
Class :			**BCA - II Semester – IV**								
****Core Courses DSC 1D and DSC 2D are compulsory while student can select either # DSC 3D/GE-3D or AIC-1B as a third subject.	DSC 1D	AIC-1B	Object Oriented Programming using Java-II	Paper-VII	3	--	--	50	40	10	4.0
			Data Structure Using 'C'-II	Paper-VIII	3	--	--	50	40	10	
	DSC 2D		Database Management System-II	Paper-VII	3	--	--	50	40	10	4.0
			Software Testing & QA	Paper-VIII	3	--	--	50	40	10	
	# DSC 3D /GE-3D		Digital Marketing	Paper-VII	3	--	--	50	40	10	4.0
			Python-II	Paper-VIII	3	--	--	50	40	10	
			Environmental Studies		3	--	--	50	40	10	NC
Total Sem-IV					**18**			**300**	**240**	**60**	**12**
Total (Theory)					**36**	--	--	**600**	**480**	**120**	**24**
Core Practical	DSC 1C & 1D			Pr. II &III	--	--	8	200	160	40	4.0
	DSC 2C & 2D			Pr. II & III	--	--	8	200	160	40	4.0
	DSC 3C & 3D /GE-3C &GE-3D		AIC 1A & 1B	Pr. II & III	--	--	8	200	160	40	4.0
Total (Practical's)							**24**	**600**	**480**	**120**	**24**
Grand Total					**36**		**24**	**1800**	**960**	**240**	**48**
	$SEC				**4**			**100**	**80**	**20**	**4**

******Core Courses: DSC 1C, DSC 2C, DSC 1D and DSC 2D (Core computer science courses)**

 # Generic Electives: DSC 3C/GE-3C and DSC 3D /GE-3D: Commerce/Management

Additional Interdisciplinary Courses – Cyber Law/Bioinformatics /Optimization technics/Data Analytics /NCC.

$The students can choose MOOCs/ NPTEL/SWAYAM/Pathshala/Add-on / Skill based courses of university/college initiated courses of same credits.

$ These courses are not compulsory, but after completion of these courses students get additional credits on their mark lists.

$ SEC courses run by colleges should be communicated to university for information & necessary action.

BCA-II Semester- III

Course Code: DSC1C (Paper-V) **Course Title: Object Oriented Programming using Java-I**

Total Contact Hours: Hrs. **Total Marks: 50(40 Lectures)**

Teaching Scheme: Theory 3 Lect./Week **Total Credits: 02**

Unit No	Content	No. of Lectures
1	**Introduction to java programming :** Features of Java, JVM, parts of java, steps to java programming, API Document, starting java programming, importing classes, Formatting the output, naming conventions , Data types in java, Operators in java, control statements in java, switch statement, break statement, continue statement, and return statement. Input/output in java-accepting input from the keyboard and display output with formatting.	12
2	**Arrays:** Array definition, types of array- single- dimensional ,multi-dimensional and jagged array ,array length property, array by command line argument. **String**:Creating String, String class methods, String comparison, immutability of string, Difference between mutable and immutable object String buffer and Builder: creating string buffer objects, String buffer class methods, String builder class, String builder class methods, String vs StringBuffer, StringBuffer vs StringBuilder. **OOPs Concept In java:** Introduction to OOP's, features of OOP's, access specifiers in java, constructor Class and object: object creation, initializing instance variables, Methods in java- instance method ,static method, 'this ' keywords, passing primitive data types to method, passing objects to method, passing arrays to method, recursion in java.	12
3	**Inheritance and polymorphism:** Inheritance -Inheritance introduction, use of inheritance , types of inheritance, 'super ' keywords in java, use of protected access specifier, Polymorphism- Polymorphism introduction, static and dynamic Polymorphism. Method overriding, method overload vs method overriding., use of 'final' keyword.Abstract method and class, Interface-interface introduction, multiple inheritance using interfaces, Abstract class vs. interfaces Packages: introduction to package ,types of packages-Built in and user defined package, creating and importing package, relating sub package in package, interfaces in package, access specifier in package, use math package. **Java I/O and stream:** java streams, OutputStream vs InputStream, OutputStream class, InputStream class, Hierarchy of OutputStream and InputStream class, Java FileWriter class, Java FileReader class, file class methods, creating file, reading file ,file copy, serialization and de- serialization in file. **Exception handling in java:** Exception- Exception Handling, Exception Handling classes Java Exceptions , Try-catch block, Multiple Catch Block, Nested try, Finally Block, Throw Keyword Throws Keyword, Throw vs Throws ,Final vs Finally, Custom Exceptions	14

Books Recommended:

1) Core Java by Dr.R. Nageshwar Rao
2) "Programming with Java" by E Balaguruswamy.
3) Horstmann, "Computing Concepts with Java 2 Essentials", John Wiley.
4) Decker & Hirshfield, "Programming Java", Vikas Publication
5) "Java-2 the complete Reference" by Patrick Naughton and HerbertzSchidt.

BCA-II Semester- III

Course Code: DSC1C (Paper-VI) **Course Title: Data structures Using 'C'- I**
Total Contact Hours: Hrs. **Total Marks: 50(40 Lectures)**
Teaching Scheme: Theory 3 Lect./Week **Total Credits: 02**

Unit No	Content	No. of Lectures
1	**An Introduction to Data Structures:** Introduction, Definition and types of Data structure. Abstract Data Type (ADT)- ADT for array, ADT for stack, ADT for queue. Algorithm: Definition, characteristics of algorithm. Complexity of algorithm- Space complexity, time complexity, Big-O Notation Design strategies of Algorithm- Divide and Conquer, Greedy Algorithm, branch & bound, backtracking, and dynamic programming.	**8**
2	**Array:** Introduction to Array, types of array- one dimensional, two dimensional and multidimensional, Operations of array- insert, delete, traverse, count, display, reverse	**4**
3	**Stack:** Introduction to Stack, Operations of stack- Create, isempty, isfull, push, pop, display, Implementation of stack using array (Static Implementation) Applications of Stack-Conversion of infix expression to postfix expression, Conversion of infix expression to prefix expression, Matching parenthesis in an expression (Checking expression is valid or invalid), Evaluation of postfix expression, Stack in recursion, Implementation of applications of stack.	**8**
4	**Queue:** Introduction to Queue, Operations of queue- Create, isempty, isfull, insert, remove, display , Types of Queue- Linear Queue, Circular Queue, Deque (Double Ended Queue), Priority queue. Implementation of all types of queue using array (Static Implementation), Difference between stack and queue, Applications of Queue	**8**
5	**Linked Lists:** Introduction to Linked Lists, Difference between Array and linked list. Types of linked list- 1) Linear linked list- Singly (Single) linear linked list and Doubly (Double) linear linked list. 2) Circular linked list- Singly (Single) circular linked list and Doubly (Double) circular linked list, Operations of linked list- Creation, Insertion, Deletion, Traversing, Searching, Display, count, reverse, Implementation of all types of linked list, Implementation of stack using linked list ,Implementation of queue using linked list	**12**

Books Recommended:

1. Tanenbaum: Data structures using C and C++
2. Data Structures Through C in Depth- S.K.Srivastava, D.Srivastava
3. Fundamentals of *Data Structures in C by Sahni*

BCA-II Semester- III

Course Code: DSC2C (Paper-V) **Course Title: Database Management System-I**

Total Contact Hours: Hrs. **Total Marks: 50(40 Lectures)**

Teaching Scheme: Theory 3 Lect./Week **Total Credits: 02**

Unit No	Content	No. of Lectures
Unit-1	**Introduction to Database Management System:** Definition, Limitations of traditional file system, Advantages of DBMS, Components of DBMS, Database Users, Database Structure Database Architecture- 2-tier and 3 level (schema) tier architecture, Instances and Schemas, Database languages, Data Independence, Types of data models(Hierarchical, Network, Relational, hybrid) Conceptual Design: E-R model-entities, attributes and its types, Relationship, Relationship sets, Generalization, Specialization, Aggregation Relational Database Concepts: Difference between DBMS and RDBMS, Introduction and features of RDBMS, 12 Codd's Rules.	**10**
Unit-2	**SQL Commands:** DDL Commands: Create, Alter, Rename, truncate, drop. DML-Insert, update, delete. DQL-SELECT Statements using WHERE clause. DCL- Grant, Revoke, TCL-Rollback, Commit and Savepoint, **User Creation:** Creating users Granting & Revoking permissions on Database Objects **Data types and Operators:** comparison, conditional, arithmetic, logical, set and Special Operators – IN (NOT IN), BETWEEN (NOT BETWEEN), LIKE (NOT LIKE), IS NULL (IS NOT NULL) **Built-in Functions:** Arithmetic, string, Date and Time, Conversion, Aggregate, OLAP and General. Common clause: Order by, Group by, having. **Integrity Constraints:** Importance of Data Integrity, NOT NULL, UNIQUE, PRIMARY, FOREIGN KEY constraint with ON DELETE, ON DELETE CASCADE, CHECK, DEFAULT constraints. **Relational Algebra operations:** Select, Project, Cartesian Product, Union, Set Difference and Join. **SQL Joins:** Equi Join/Inner Join/Simple Join, Cartesian, Non-Equi, Outer Joins, Self-Join and lossless join.	**20**
Unit-3	**Purpose and usage of a View:** Types of VIEWS, Relational Views, Object Views, Using VIEWS for DML Operations, In-Line View, Forced Views, Putting CHECK Constraint upon VIEWS, Creation of READ ONLY VIEWS, Materialized Views **Working with Sub Queries and Nested Sub Queries:** purpose and usage of a Sub Query, Type of Sub Queries- Single Row, Multiple Row, Multiple Column, Applying Group Functions in Sub Queries, IN, ANY, SOME, ALL Operators in Sub Queries. Correlated Sub Queries: Handling Data Retrieval with EXISTS and NOT EXISTS Operators **Working with Sequences-** Creating, retrieving data, modifying, dropping sequences, Synonyms, **Index-** What is index, advantages Types of indexes, creating index, Retrieving data using index, Pseudo Columns: Types of Pseudo Columns, CURRVAL and NEXTVAL, LEVEL, ROWID, ROWNUM.	**20**

Books Recommended:

1) Database System Concepts by Korth Silberschetz
2) Fundamentals of Database Systems by Elmsari, Navathe
3) SQL and PL/SQL Programming by Ivan Bayross
4) SQL and PL/SQL Programming by Oracle Press

BCA-II Semester- III

Course Code: DSC2C (Paper-VI) **Course Title: Software Engineering**
Total Contact Hours: Hrs. **Total Marks: 50(40 Lectures)**
Teaching Scheme: Theory 3 Lect./Week **Total Credits: 02**

Unit No	Content	No. of Lectures
Unit-1	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.	**10**
Unit-2	Software requirements: Types of Requirements: System, Functional, Non-functional, User. Fact finding techniques: Interviews, Questionnaire, Record reviews, Observation Analysis and Design Tools: Flow chart, Decision tables and Trees, Structured English, HIPO System Design: Data Flow Diagram (Physical, Logical), Entity Relation Diagram ERD Case studies: Inventory, Library, Payroll, Loan, Online Booking, Data Dictionary, structured chart, Input and output design, Database Design: Types of Dependencies, Normalization(1NF, 2NF, 3NF, BCNF, 4NF, 5NF),	**15**
Unit-3	Coding: Coding standards, Size Estimation, Effort Estimation, and Cost Estimation, Testing fundamentals Software Implementation and Maintenance: Traditional and incremental approaches, conversion methods, Overview of maintenance process, types of maintenance. Software Quality Assurance: SQA Tasks, Goals and Metrics, Software Reliability. Software risk management: definition, types of risk, risk identification-risk monitoring and management.	**15**

Books Recommended:

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-II Semester- III

Course Code: DSC 3C/GE-3C (Paper-V) **Course Title: Financial Accounting with Tally**

Total Contact Hours: Hrs. **Total Marks: 50 (40 Lectures)**

Teaching Scheme: Theory 3 Lect./Week **Total Credits: 02**

Unit No	Content	No. of Lectures
Unit-1	**Introduction to Book-keeping and Accountancy-** Definition and Objectives, Importance of Book-keeping, Difference between Book-keeping and Accountancy, Definition of Accountancy, Basis of Accounting System, characteristics of accounting information, Basic Accounting Terminologies, Accounting Concepts, Conventions and Principles, Accounting Standards (AS) and IFRS **Fundamentals of Double Entry Book-keeping-** Introduction of Double entry Book-keeping System, Methods of Recording Accounting Information (Indian, Single, Double), Advantages of Double entry Book-keeping system, Classification of Accounts, Golden Rules of Debit and Credit (Traditional Approach), Modern Approach of Rules of Accounts, Accounting Equations **Journal-** Importance and Utility of Accounting Documents, Definition, Importance and Utility of Journal, Specimen of Journal, Recording of Journal entries with GST. **Ledger-** Definition and Importance of Ledger, Specimen of Ledger, Posting of entries from Journal/Subsidiary Books to Ledger, Balancing of Ledger Accounts, Preparation of Trial Balance	**12**
Unit-2	**Subsidiary-Books-**Introduction and need for maintaining Subsidiary Books, Cash Book with Cash Column, Cash Book with Cash and Bank Columns, Simple and Analytical Petty Cash Book under Imprest System, Purchase Book, Purchase Return Book, Sales Book, Sales Return Book, Journal Proper. **Bank Reconciliation Statement-** Introduction and Utilities of Accounting Documents, Need and Importance, Introduction of Bank Reconciliation Statement, Reasons for difference between Cash Book balance and Pass Book balance, Specimen of Bank Reconciliation Statement. **Depreciation-** Introduction and Importance of Depreciation, Factors of Depreciation, Methods of Depreciation, Accounting Treatment for Depreciation. **Rectification of Errors-**Introduction and Effects of errors, Types of Errors, Detection & Rectification of errors, Preparation of Suspense Accounts	**14**
Unit-3	**Final Accounts of a Proprietary concern-** Introduction, Objectives and Importance of Final Accounts, Preparation of Trading Account. Preparation of Profit and Loss Account, Preparation of Balance Sheet. Effects of following adjustments. Closing stock, Outstanding Expenses, Prepaid Expenses, Depreciation on assets, Bad debts and R.D.D., Discount on Debtors and Creditors, Income received in advance , Accrued Income, Goods distributed as free sample, Goods withdrawn by proprietor for Personal use, Interest on capital, Interest on Drawings **Introduction to Tax Deducted at Source (TDS)-**TDS in Tally, TDS Masters, Vouchers / Transactions, Advance to a Party, TDS Reports, TDS Return, TDS E-Return, TDS Outstanding, GST Basics. **Implementation through Tally** Create, Alter & Display Stock Groups and Stock Items, All inventory	**14**

	voucher types and transactions Inventory details in accounting vouchers. Reports like Stock summary, Inventory books like Stock item, Group summary, Stock transfers, Physical stock register, Movement analysis, Stock group & item analysis, stock category analysis Ageing analysis, Salesorder & Purchase order book, Statement of inventory related to Godowns, categories, stock query, Reorder status, Purchase & Sales order summary, Purchase & Sales bill pending, Exception reports like negative stock& ledger, overdue receivables & payables, memorandum vouchers, optional vouchers, post-dated vouchers, reversing journal	

Books Recommended:

1) Elements of double entry book keeping – Batliboi
2) Advanced Accounts – M.C.Shukla, T.S.Grewal and S.C.Gupta
3) An Introduction to Accountancy – S.N.Maheshwari.
4) Accounting for Management – S.K.Bhattach
aryya& John Dea

BCA-II Semester- III

Course Code: DSC 3C/GE-3C (Paper-VI) **Course Title: Operating System-II**
Total Contact Hours: Hrs. **Total Marks: 50 (40 Lectures)**
Teaching Scheme: Theory 3 Lect./Week **Total Credits: 02**

Unit No	Content	No. of Lectures
Unit-1	**Deadlocks**: Definition, Characteristics, Resource Allocation Graph, Methods of Handling Dead Locks- Deadlock Prevention, Deadlock Avoidance, Deadlock detection and Recovery.	**10**
Unit-2	**Memory Management:** Basic, Address Binding, Logical &Physical address Space, Dynamic Loading, Overlays, Memory partitioning: Fixed and Variable, Contiguous Memory allocation, Allocation Strategies(First Fit, Best Fit, Worst Fit),Swapping, fragmentation, compaction, Paging and Segmentation. **Virtual Memory:** demand paging, Page fault, Page Replacement policies: Optimal(OPT), First in First Out(FIFO),Least Recently used(LRU), Thrashing.	**15**
Unit-3	**Storage Management:** File concept, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping). **Disk Management:** disk structure, disk scheduling (FCFS, SSTF, SCAN, CSCAN), disk reliability, disk Formatting, boot block, bad blocks.	**15**

Books Recommended:

1. Operating System Concepts by Siberchatz and Galvin.
2. Modern O.S. By Andrews Tanenbaum.

BCA-II Semester- III

Course Code: $ SEC-I **Course Title: Web Development using PHP**

Total Contact Hours: Hrs. **Total Marks: 100 (60 Lectures)**

Teaching Scheme: Theory 4 Lect./Week **Total Credits: 04**

Unit No	Content	No. of Lectures
1	Introduction to web applications, Client Side Vs Server Side Scripting WebServers: Local Servers and Remote Servers, Installing Web servers, Static website vs Dynamic website development. Introduction to PHP and PHP Framework, Basic PHP syntax, Data types in PHP, Variables, Constants, operators and Expressions, printing data on PHP page, Control statements–if, switch case, for, while, do while. Arrays: Initialization of an array, Iterating through an array, Sorting arrays, Array Functions, types of array Functions: Defining and Calling Functions, Passing by Value and passing by references, Inbuilt Functions.	**15**
2	String: Formatting String for Presentation and Storage, Joining and Splitting String, Comparing String, Matching and replace Substring, patterns, basic regular expressions. Working with file and Directories: Understanding file& directory, Opening and closing a file, Coping , renaming and deleting a file, Working with directories, Building a text editor, File Uploading & Downloading, Generating Images with PHP-Basics of computer Graphics, Creating Image, Manipulating Image, Using text in Image. Error And Exception Handling: Error Logging, Configuration Directives, PHP's Exception Class, Throw New Exception, Custom Exceptions, Date and Time Functions, HTTP Authentication, PHP Authentication, Authentication Methodologies	**15**
3	State Management: Cookies: Setting time in a cookie with PHP, Deleting a cookie, Creating session cookie, Working with the query string Session: Starting a session, Registering Session variables, working with session variables, destroying session, passing session Ids, encoding and decoding session variables, Auto-Login, Recently Viewed Document Index	**10**
4	Working With Forms: Forms controls properties, methods and events, Retrieving form data with $_POST, $_GET and $_REQUEST arrays, Validating retrieved data, Strategies for handling invalid input, Super global variables, Super global array, Importing user input, Accessing user input, Combine HTML and PHP code, Using hidden fields, Redirecting the user, File upload and scripts, Validation-Serverside validation, Client side validation. Database Connectivity with Php: Database Functions, Gmail Data Grid options SQL Injection, Uploading and downloading images in Database, Registration and Login forms with validations, Pagging, Sorting	**20**

Books Recommended:

1) PHP: The Complete Reference-Steven Holzner.
2) Programming PHP- Rasmusler dorf, Kevin Tatroe.

BCA-II Semester- IV

Course Code:DSC1D Paper-VII **Course Title: Object Oriented Programming using Java-II**

Total Contact Hours: Hrs. **Total Marks: 50(40 Lectures)**

Teaching Scheme: Theory 3 Lect./Week **Total Credits: 02**

Unit No	Content	No. of Lectures
1	Collection Framework in java: Type casting:- type casting in java, types of type casting. Wrapper classes: use of Wrapper classes, Number classes(Long,Integer,Byte,Short,Float and double) and importance methods of Number class, Character class and importance methods of character class, auto boxing and unboxing, Collection Framework:-Use of Collection framework, Hierarchy of Collection Framework, Collection objets-Set,List, Map, Queue Collection classes-Stack,ArrayList,vector,Linked List,priority queue,HashSet,LinkedHashSet,Sotred Set,TreeSet,Hashtable and HashMap	**10**
2	**Multithreading:** Single tasking, Multi-tasking, use of thread, creating and running thread, terminating thread, thread class methods, multiple threading, Thread communication, thread priorities,Application of thread and thread life cycle. **Networking:** Introduction to Networking, TCP/IP protocol, UPD protocol, socket programming, InetAdress Class, URL Connection class, communication between client and server, two way communication between client and server.	**15**
3	**Swing:** Hierarchy of Swing classes JButton, JLabelJava ,JTextField, JTextArea, JPasswordField, JCheckBox, JRadioButton,JComboBox, , JList, JOptionPaneJava JScrollBar, JMenuItem & JMenuJava , Image Event hadling:- Java Event Handling, Java Event classes and Listener interfaces. LayoutManager- BorderLayout FlowLayout, GridLayout, CardLayout , BoxLayout JDBC: JDBC Introduction ,JDBC Driver ,DB Connectivity Steps ,Connectivity with Oracle or MySql DriverManager, ConnectionStatement, ResultSet, PreparedStatement, ResultSetMetaData, CallableStatement	**15**

Books Recommended:

1) Core Java by Dr.R. Nageshwar Rao
2) "Programming with Java" by E Balaguruswamy.
3) Horstmann, "Computing Concepts with Java 2 Essentials", John Wiley.
4) Decker & Hirshfield, "Programming Java", Vikas Publication
5) "Java-2 the complete Reference" by Patrick Naughton and HerbertzSchidt.

BCA-II Semester- IV

Course Code: DSC1D Paper-VIII **Course Title: Data structures using 'C'- II**

Total Contact Hours: Hrs. **Total Marks: 50(40 Lectures)**

Teaching Scheme: Theory 3 Lect./Week **Total Credits: 02**

Unit No	Content	No. of Lectures
1	**Trees:** Introduction to Tree, Introduction to Binary Trees, Types of Binary tree- Strictly Binary tree, Complete Binary tree, Extended (2-Tree) Binary tree, Binary expression tree, Binary Search tree, Heap Tree- Min heap tree, Max heap tree, Representation of Binary tree using- Array, Linked list Operations of Binary search tree-Creating and inserting node, Searching node, Counting total nodes, Counting and displaying leaf nodes, Tree Traversal methods- Preorder, Inorder, Postorder, Deletion of Nodes, Implementation of binary search tree, Height balanced tree/Balanced Binary Tree/AVL tree, Application of tree	**10**
2	**Graph:** Concept & terminologies used in graph, Graph Representation using- Array and linked list, Graph traversals – BFS & DFS, Dijakstra's shortest path algorithm, and application of graph.	**10**
3	**Sorting:** Introduction and definition of Sorting, Types of Sorting-Bubble sort, Quick sort, Shell sort, Selection sort, Insertion sort, Heap Sort, Merge sort, Radix Sort, Tree Sort techniques **Searching:** Introduction and definition of Searching, Types of searching-Linear (Sequential) Search, Binary Search, Indexed sequential search, Hashing and different Hash functions.	**20**

Books Recommended:

1. Tanenbaum: Data structures using C and C++
2. Data Structures Through C in Depth- S.K.Srivastava,D.Srivastava
3. Fundamentals of *Data Structures in C by Sahni*
4. Ulman: Data structures and Algorithms
5. Nikaulus Wirth: Algorithms, data structures, Programs.

BCA-II Semester- IV

Course Code: DSC2D Paper-VII **Course Title: Software Testing**
Total Contact Hours: Hrs. **Total Marks: 50(40 Lectures)**
Teaching Scheme: Theory 3 Lect./Week **Total Credits: 02**

Unit No	Content	No. of Lectures
Unit-1	Importance or need of software testing Introduction to White Box Testing: Advantages and Disadvantages of White box testing, Static Techniques: Informal Reviews, Walkthroughs, Technical Reviews, Inspection, Dynamic Techniques or Structural Techniques-Statement Coverage Testing, Branch Coverage Testing, Path Coverage Testing, Conditional Coverage Testing, Loop Coverage Testing Introduction to Black Box Testing: Advantages and Disadvantages of black box testing, Black Box Techniques: Boundary Value Analysis, Equivalence Class Partition, State Transition Technique, Cause Effective Graph, Decision Table, Use Case Testing Experienced Based Techniques: Error guessing, Exploratory testing	**12**
Unit-2	Levels of Testing Functional Testing: Integration Testing and types - Top Down , Bottom Up , Non Incremental, System Testing, Acceptance Testing- Alpha and Beta, Smoke Testing, Regression Testing- Unit , Regional, Full Non Functional Testing: Adhoc Testing, Performance Testing : Load Testing, Stress Testing, Volume Testing, Soak Testing, Recovery Testing Test cases design Techniques: Test Case and types, Test Case Template, write a test case and examples, Preparing Review Report	**12**
Unit-3	Software Test Life Cycle, Writing Test Plan, Preparing Traceability Matrix, Writing Test Execution Report and Summary Report. Difference between Bug, Defect, Failure, Error, Bug/Defect Life Cycle, Defect Tracking and Reporting, Types of Bugs, Identifying the Bugs, Reporting the Bugs Introduction to automated testing: Differences between Manual and Automation Testing, Install and configure selenium testing tool, Case study through selenium tool: Design test case for Email login page, Internet Banking Login, Online shopping.	**16**

Books Recommended:

1) The art of Software Testing–Glenford J. Myers
2) Lessons learned in Software Testing– Cem Kaner, James Bach, Bret Pettichord
3) A Practitioner's Guide to Software Test Design- Lee Copeland
4) Software Testing Techniques, 2ndedition- Boris Beizer

BCA-II Semester- IV

Course Code: DSC2D Paper-VIII **Course Title: Database Management System-II**
Total Contact Hours: Hrs. **Total Marks: 50(40 Lectures)**
Teaching Scheme: Theory 3 Lect./Week **Total Credits: 02**

Unit No	Content	No. of Lectures
Unit-1	Introduction of Transaction, ACID properties, transaction states, scheduling and its types, conflict and view serializability, Introduction of Concurrency Control, problems of concurrency control, lock based protocols, timestamp based protocol, deadlock, deadlock handling methods.	**12**
Unit-2	Introduction, recovery algorithms, log base recovery, shadow paging, recovery with concurrent transaction, checkpoints or syncpoints or savepoints. Query Optimization: Overview Query Processing and Optimization – Heuristics and Cost Estimates in Query Optimization.	**8**
Unit-3	Introduction to PL/SQL, Advantages, Architecture, Data types, Variable and Constants, Using Built-in Functions, Conditional, Looping and Iterations Statements, Selection Case, Simple Case, GOTO Label and EXIT, SQL within PL/SQL. Procedures in PL/SQL: STORED PROCEDURES, PROCEDURE with Parameters (IN, OUT and IN OUT), POSITIONAL Notation and NAMED Notation, Dropping a Procedure. Functions in PL/SQL: Difference between Procedures and Functions, types of functions and parameter modes, Packages in PL/SQL: importance, advantages Implementing packages, Private and Public Objects in PACKAGE Cursor in PL/SQL: Types of Cursors, Cursor Attributes, Cursor with Parameters, Cursors with LOOPs Nested Cursors, Cursors with Sub Queries and procedure. Exceptions in PL/SQL: Types of exceptions, RAISE_APPLICATION_ERROR, PRAGMA_AUTONOMOUS_TRANSACTION Database Triggers in PL/SQL: Types of Triggers, Row Level Triggers, Statement Level Triggers, Implementing triggers for various DML operations (insert, delete, update), DDL Triggers, Trigger Auditing.	**20**

Books Recommended:

1) Database System Concepts by Korth Silberschetz
2) Fundamentals of Database Systems by Elmsari, Navathe
3) SQL and PL/SQL Programming by Ivan Bayross
4) SQL and PL/SQL Programming by Oracle Press

BCA (Science)-II Semester- III

Course Code: # DSC 3D /GE-3D **Course Title: Digital Marketing**
Total Contact Hours: Hrs. **Total Marks: 50 (40 Lectures)**
Teaching Scheme: Theory 6 Lect./Week **Total Credits: 02**

Unit No	Content	No. of Lectures
Unit-1	**Introduction to Digital Marketing:** Meaning of Digital Marketing, Differences from Traditional Marketing, Return of Investments on Digital Marketing vs. Traditional Marketing, E Commerce, Tools used for successful marketing, SWOT Analysis of Business for Digital Marketing, Media and promotion plan, Blogs, Websites, Portal and Their Differences, Visibility, Visitor Engagement, Conversion Process, Retention, Performance Evaluation, Online Reputation Management	**8**
Unit-2	**Search Engine Optimization (SEO):** Optimization, Definition, its importance, Strategies and techniques used to optimize any article/page/website/blog for traffic generation and revenue, different On page Optimization Techniques, different Off Page Optimization Techniques, Preparing Reports, Creating Search Campaigns, and Creating Display Campaigns. **Social Media Optimization (SMO):** Introduction to Social media, Types of Social Media platforms (Facebook, Twitter, Instagram, YouTube), Roles of Social Media in Marketing, Goals and Strategies, Facebook Marketing, Email Marketing, Google plus marketing , Word press Blog Creation, Twitter Marketing, Linkedln Marketing, Pinterest, Instagram Marketing, eCommerce Marketing, Affiliate Marketing, SMS Marketing, mage Optimization, social media Analytical Tools.	**16**
Unit-3	**Search Engine Marketing:** Introduction and Use of Search Engine Marketing, Introduction to Online Advertising and Ad words, Tools used — Pay Per Click, Google Adwords, Display Advertising Techniques, Advertisement Designing, Adwords Account And Campaign Basics, Adwords Targeting And Placement, Adwords Bidding And Budgeting, Adwords Tools, Opportunities, Optimizing Performance, Ads Type, Bidding Strategies, Search Network, Display Network, Shopping Ads, Video Ads, Universal App Ads, Tracking Script, Remarketing, Performance Monitoring, Report Generation. **Website Traffic Analysis:** Web Analytics Tools, Google Analytics, Navigating Google Analytics, Traffic Sources, Acquisition, Behavior, Content, Visitors, Live Data, Demographics	**16**

Books Recommended:

1) Ryan, D. (2014). Understanding Digital Marketing: Marketing Strategies for Engaging the Digital Generation, Kogan Page Limited.
2) The Beginner's Guide to Digital Marketing (2015). Digital Marketer. Pulizzi,J.(2014) Epic Content Marketing, Mcgraw Hill Education.
3) Lorrie Thomas. The McGraw-Hill 36-Hour Course: Online Marketing, McGraw Hill.

BCA (Science)-II Semester- III

Course Code: # DSC 3D /GE-3D **Course Title: Python-II**

Total Contact Hours: Hrs. **Total Marks: 50 (40 Lectures)**

Teaching Scheme: Theory 6 Lect./Week **Total Credits: 02**

Unit No	Content	No. of Lectures
Unit-1	**Functions:** Difference between a Function and a Method, Defining a function, Calling a function, Advantages of functions, Types of functions, Function parameters:-Formal parameters, Actual parameters, Anonymous functions, Global and Local variables, **Modules:** Importing module, Creating & exploring modules, Math module, Random module, Time module, Numpy, Scipy **Object Oriented Programming:** Features, Concept of Class & Objects, Constructor, Types of Variables, Namespaces, Types of Methods, Inner Classes, Constructors in Inheritance, Overriding Super Class Constructors and Methods, Types of Inheritance, Abstract Classes and Interfaces, The Super() Method, Operator Overloading, Method Overloading, Method Overriding.	**16**
Unit-2	**Threads:** Introduction, uses, types, creating threads, thread class methods and synchronization **Exception Handling:** Errors in a Program, Exceptions, Exception handling, Types of Exceptions, User-defined Exceptions **Python File Operation:** Types of File, Opening and Closing a File, Reading and writing to files, Manipulating directories	**12**
Unit-3	**Graphical user interface-** root window, fonts and colors, working with containers, canvas, frame, widgets and its types. **Database connectivity**-Installing MySQLdb module, working with MySQL, Retrieving, inserting, Deleting and Updating rows into table, creating database tables	**12**

Books Recommended:

1. Python Cookbook: Recipes for Mastering Python 3 by Brian Kenneth Jones and David M. Beazley-O'Reilly Media
2. Beginning Python by Magnus Lie Hetland-Apress

Lab Course on Data Structure I and II

Array

1) Write a program to implement array with following operations:
    a) Insert Element
    b) Delete element from entered position
    c) Traverse array element.
    d) Count
    e) Search element
2) Write a programs that prints array elements in reverse order.
3) Write a program that finds only even elements in an array.
4) Write a program that finds only odd elements in an array.
5) Write a program that finds addition of matrices.
6) Write a program that finds multiplication of matrices.

Stack

1) Write a program to implement stack by using array. (Static Implementation of stack)
2) Write a program, which reverses the string by using stack.
3) Write a program to check entered string is palindrome or not by using stack.
4) Write a program to convert decimal number into binary number by using stack.
5) Write a program to count total number of vowels present in string by using stack.
6) Write a program which convert infix expression into prefix expression.
7) Write a program which convert infix expression into Postfix expression.
8) Write a program which check entered expression is valid or not.
9) Write a program for evaluation of postfix expression.
10) Write a program to calculate factorial of entered number by using recursion.
11) Write a program to calculate digit sum of entered number by using recursion.
12) Write a program to find face value of entered number by using recursion.

Queue

1) Write a program to implement linear queue by using array. (Static Implementation of queue)
2) Write a program to implement Circular queue.
3) Write a program to implement Priority queue.
4) Write a program to implement IRD (Input Restricted Deque)
5) Write a program to implement ORD (Output Restricted Deque)

Linked List

1) Write a program to implement singly linear linked list with its basic operations.
2) Write a program to implement stack by using linked list. (Dynamic implementation)
3) Write a program to implement queue by using linked list. (Dynamic implementation)
4) Write a program to implement doubly linear linked list with its basic operations.
5) Write a program to implement singly circular linked list with its basic operations.
6) Write a program to implement doubly circular linked list with its basic operations.

Tree

1) Write a program to implement binary search tree with tree traversal methods.
2) Write a program to implement BST with following operations:
    I) Insert Node
    II) Count Leaf nodes
    III) Count Non-Leaf nodes
    IV) Count Total nodes
3) Write a program to implement BST with following operations:
    I) Insert Node II) Find Maximum node
    III) Find Minimum Node IV) Search node
    V) Display only odd nodes VI) Display only even nodes
    VII) Display leaf nodes VIII) Find level of node
    IX) Find degree of node X) Delete Node

Graph

1) Write a program to represent undirected and directed graph by using Adjacency matrix.
2) Write a program to represent weighted graph by using Adjacency matrix.
3) Write a program to implement graph by using linked list and perform following operations:
    1) Insert Vertex (Node)
    2) Display Vertices
    3) Search Vertex
    4) Insert Edge
    5) Find adjacent vertices
    6) Display Graph
4) Write a program to implement breadth first search (BFS) traversal of graph.
5) Write a program to implement depth first search (DFS) traversal of graph.

Sorting and Searching

1) Write a program to implement simple exchange sort method.
2) Write a program to implement bubble sort method.
3) Write a program to implement insertion sort method.
4) Write a program to implement selection sort method.
5) Write a program to implement Shell sort method.
6) Write a program to implement linear searching technique for unsorted data.
7) Write a program to implement linear searching technique for sorted data.
8) Write a program to implement Binary search technique.

Lab Course on Software Testing:

1) Design test case **for Internet Banking Application**
2) Design test case for Gmail Login Functionality
3) Design test case **for college admission Application**
4) Design test case for online order processing.
5) Design test case for social networking sites.
6) Design test case for MS-word application
7) Design test case for simple calculator
8) Design test case for ball pen.
9) Design test case for Paint application.
10) Design test case for Online Flight Booking

Lab Course on DBMS using Oracle:

1. Create the following Databases.

Salesmen

SNUM	SNAME	CITY	COMMISSION
1001	Prashnat	Mumbai	12
1002	Rajesh	Surat	13
1004	Anandi	Mumbai	11
1007	Priya	Delhi	15
1003	Suchita	Pune	10
1005	Nayan	Baroda	14

Customers

CNUM	CNAME	CITY	RATING	SNUM
2001	Harsh	Baroda	100	1001
2002	Gita	Pune	200	1003
2003	Lalit	Mumbai	200	1002
2004	Govind	Delhi	300	1002
2006	Chirag	Surat	100	1001
2008	Prajkta	Delhi	300	1007
2007	Sushma	Mumbai	100	1004

Orders

ONUM	AMOUNT	ODATE	CNUM	SNUM
3001	18	10/3/2019	2008	1007
3003	767	15/3/2019	2001	1001
3002	1900	10/3/2019	2007	1004
3005	5160	20/4/2019	2003	1002
3006	1098	20/4/2019	2008	1007
3007	1713	10/5/2019	2002	1003
3008	75	10/5/2019	2004	1002
3010	4723	15/6/2019	2006	1001
3011	1309	18/3/2019	2004	1002

Solve the following queries using above databases and where clause range searching and pattern matching.

1. Produce the order no, amount and date of all orders.
2. Give all the information about all the customers with salesman number 1001.
3. Display the following information in the order of city, sname, snumand commission.
4. List of rating followed by the name of each customer in Surat.
5. List of snum of all salesmen with orders in order table without any duplicates.

Solve the following queries using above databases and group by clause.

1. Find out the largest orders of salesman 1002 and 1007.
2. Count all orders of October 3, 1997.
3. Calculate the total amount ordered.
4. Calculate the average amount ordered.
5. Count the no. of salesmen currently having orders.

Solve the following queries using above databases and formatted output and order by clause.

1. List all salesmen with their % of commission.
2. Display the no. of orders for each day in the descending order of the no. of orders in the following format.
   FOR dd-mon-yy, there are __ Orders.
3. Assume each salesperson has a 12% commission. Write a query on the order table that will produce the order number, salesman no and the amount of commission for that order.
4. Find the highest rating in each city in the form :
   For the city (city), the highest rating is : (rating)
5. List all in descending order of rating.
6. Calculate the total of orders for each day and place the result in descending order.

Solve the following queries using above databases and join.

1. Show the name of all customers with their salesman's name.
2. List all customers and salesmen who shared a same city.
3. List all orders with the names of their customer and salesman.
4. List all orders by the customers not located in the same city as their salesman.
5. List all customers serviced by salespeople with commission above 12%.

Solve the following queries using above databases and join and subquery.

1. Find all orders attributed to salesmen in 'London'.
2. List the commission of all salesmen serving customers in 'London'.
3. Find all customers whose cnum is 1000 above than the snum of Sejal.
4. Count the no. of customers with the rating above than the average of'Surat'.
5. List all orders of the customer 'Chirag'.

Solve the following queries using above databases and delete and update.

1. Remove all orders from customer Chirag from the orders table.
2. set the ratings of all the customers of Piyush to 400.
3. Increase the rating of all customers in Rome by 100.
4. Salesman Sejal has left the company. Assign her customers to Miti.
5. Salesman Miti has resigned. Reassign her number to a new salesman Gopal whose city is Bombay and commission is 10%.

Solve the following queries using above databases and alter table and table constraints..

1. How the onum field is forced to be an unquie?
2. Create an index to permit each salesman to find out his orders by date quickly.
3. Write a command to enforce that each salesman is to have only one customer of a given rating.
4. Write a command to add the item-name column to the order table.
5. Write a command to create the salesmen table so that the default commission is 10% with no NULLs permitted, snum is the primary key and all names contain alphabeticals only.
6. Give the commands to create our sample tables (salesmen, customer,orders) with all the necessary constraints like PRIMARY KEY, NOT NULL , UNIQUE, FOREIGN KEY.

Solve the following queries using above databases and view.

1. Create a view called Big orders which stores all orders larger than Rs. 4000.

2. Create a view Rate count that gives the count of no. of customers a teach rating.
3. Create a view that shows all the customers who have the highest ratings.
4. Create a view that shows all the number of salesmen in each city.
5. Create a view that shows the average and total orders for each salesmen after his name and number.
6. Create a cursor emp_cur,fetch record from emp table and check whether sal>10000 then update Grade = 'A' else if sal = > 5000 and sal<= 10000 then update Grade = 'B'
7. Write a procedure to find the table structure of a given number
8. Write a procedure on software table to calculate selling cost of all software of a specified person

Lab Course on Core Java:

1. WAP to demonstrate the use of various data types.
2. WAP which will check number for Armstrong, prime, palindrome & perfect number.
3. WAP USING arrays to sort player name along with timing of Athlete (sort using two dimensional array).
4. WAP to demonstrate the use of Access Control.(Public, private , protected).
5. WAP using static & non static data members.
6. WAP using Interface.
7. WAP to demonstrate use of Exception Handling.
8. WAP which will create user defined Exception.
9. WAP which will accept string and calculate how many vowels present in it.
10. WAP to implement any two collection classes.
11. WAP to implement InetAdress Class.
12. WAP Event classes and Listener interfaces.
13. WAP which will create following threads.
    a. Print even & odd numbers.
    b. Print Hello 15 times.
    c. Print the prime number.
14. WAP which will demonstrate overloading & Inheritance.
15. WAP to show demo of parameterized constructor.
17. WAP to append the contents of one file with another file.
18. WAP to develop a calculator using Applet (functions showing addition, subtraction, Multiplication and Division.
19. WAP which will insert student records into database having fields roll no, name, marks of five subjects, total marks and percentage and display the same.
20.WAP to implement GridLayout, CardLayout.

Lab Course on Web Technology using PHP

1) Write PHP code to check entered number is Armstrong or Not.
2) Write a menu driven program to perform following operations:
a) Check Number is Palindrome or not.
b) Check Number is Perfect or not.
c) Find face value of Entered number.
d) Check Number is Prime or not.
e) Check Number is Strong or not.
3) Write a PHP code to perform following operations:
a) Sort array element
b) Find Maximum and Minimum number in array
c) Merge two arrays in third array.
d) Swap two array elements
4) Write a program to overload the constructor.
5) Write a program which uses the static methods and static variables.
6) Write a program to implement different types of inheritance.
7) Write a program to implement interface.
8) Write a program to handle different types of exceptions.
9) Write a program which shows the use of 'final' keyword.

10) Write a program to copy the content of one file into another.
11) Write a program to merge two files into third file.
12) Design a web application to perform following task on employee table.
I) Add New II) Save III) Delete IV) Update V) Move First VI) Move Last
13) Design a web application that uses cookies and session object.

PUNYASHLOK AHILYADEVI HOLKAR SOLAPUR UNIVERSITY, SOLAPUR

NAAC Accredited 2015
'B' Grade (CGPA 2.62)

Name of the Faculty: Science and Technology

CHOICE BASED CREDIT SYSTEM

Syllabus: Computer Application

Name of the Course: BCA - III(Semester –V and VI)

(Syllabus to be implemented w.e.f. June 2024)

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Faculty of Science and Technology

Choice Based Credit System

(CBCS) (w.e.f.2024-25)Revised

Structure for BCA - III

Subject/ Core Course	Name and Type of the Paper		No. of Papers/ Practical	Hrs./ Week			Total Mark sper Paper	UA	CA	Credits
	Type	Name		L	T	P				
Class:	**BCA - III Semester – V**									
Ability Enhancement Course	(AECC)	English (Business English)	Paper II Part A	4	--	--	50	40	10	2.0
Core	DSC 1E	Data Communication and Networking	Paper IX	4	--	--	100	80	20	4.0
	DSC 1F	Data Warehousing and Data Mining	Paper X	4	--	--	100	80	20	4.0
	DSC 1G	.NET Core	Paper XI	4	--	--	100	80	20	4.0
	DSE 1	A-Data Science using Python **or** B-Kotlin Programming **or** C-ReactJS	Paper XII	4	--	--	100	80	20	4.0
Skill Enhancement Course	SEC 2	Advanced Java		4	--	--	100	80	20	4.0
Total Theory Semester-V				**24**			**550**	**440**	**110**	**22**
Class:	**BCA - III Semester – VI**									
Ability Enhancement Course	(AECC)	English (Business English)	Paper II Part B	4	--	--	50	40	10	2.0
Core	DSC 2E	Network Security	Paper XII	4	--	--	100	80	20	4.0
	DSC 2F	Data Visualization using PowerBI	Paper XIII	4	--	--	100	80	20	4.0
	DSC 2G	ASP.Net Core	Paper XIV	4	--	--	100	80	20	4.0

	DSE 2	A- Linux and Shell Programming **or** B- Mobile Application Development using Kotlin **or** C- Internet of Things	Paper XV	4	--	--	100	80	20	4.0
Total Theory Semester-VI				**20**			**450**	**360**	**90**	**18**
Practical's on		Project		--	--	5	100	80	20	4.0
		DSC 1F and 2F		--	---	5	100	80	20	4.0
		DSC 1G and 2G		--	--	5	100	80	20	4.0
		DSE 1A/B/C and 2A/B/C		--	--	5	100	80	20	4.0
Total (practical's)						**20**	**400**	**320**	**80**	**16**
Grand Total				**42**	**--**	**20**	**1400**	**1120**	**280**	**56**

% Theory paper: (Core Computer Science Course)
$The students can choose MOOCs/ NPTEL/SWAYAM/Pathshala/Add-on / Skill based courses of university/college initiated courses of same credits.
$ These courses are not compulsory, but after completion of these courses students get additional credits on their
Marklist.
$SEC Courses initiated by colleges

Abbreviations:

L: Lectures T: TutorialsP: Practicals

UA : University AssessmentCA : College Assessment CC: Core Course

AEC : Ability Enhancement Course DSE : Discipline Specific Elective PaperSEC : Skill

Enhancement Course

BCA-III (CBCS)						
Sem:	V					
Paper Type:	DSC 1E		**Paper No:**	Paper IX		
Paper Name:	**Data Communication and Networking**					
Credit:	04		**Theory:**	4 Hrs./week		
Marks:	**UA:**	80	**CA:**	20	**Total:**	100

Course Objectives:

1. To understand the structure of Data Communications System and its components.
2. Be familiarize with different network terminologies.

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

1. Familiarize with contemporary issues in network technologies.
2. Know the layered model approach explained in OSI and TCP/IP network models
3. Identify different types of network devices and their functions within a network.
4. Know the Basic routing mechanisms, IP addressing scheme and internet working concepts.
5. Familiarize with IP and TCP Internet protocols.
6. Understand major concepts involved in design of WAN, LAN and wireless networks.
7. Know the basics of network configuration and maintenance.
8. Know the fundamentals of network security issues.

Unit 1: Introduction to Computer Networks **[20]**

Network Definition, Network Topologies, Network Classifications, Network Protocol, Layered Network Architecture, Overview of ISO-OSI Reference Model, Overview of TCP/IPProtocol Suite.

Data Communication Fundamentals and Techniques:

Signals-Analog and Digital Signal, Data-Rate Limits, Digital to Digital Line Encoding Schemes, Pulse Code Modulation, Parallel and Serial Transmission, Digital to Analog Modulation, Multiplexing Techniques- FDM, TDM, Transmission Media, Switching: Circuit Switching, Message Switching, Packet Switching

Unit 2: Data Link Layer Functions and Protocols **[10]**

Design issues, Error Detection and Error Correction Techniques, Data-Link Control- Framing and Flow Control, Error Recovery Protocols-Stop and Wait ARQ, Go-Back-N ARQ, Point to Point Protocol on Internet.

Unit 3: Multiple Access Protocol and Network Layer **[10]**

Design issues, CSMA/CD Protocols, Ethernet LANS; Connecting LAN and Back -Bone Networks- Repeaters, Hubs, Switches, Bridges, Router and Gateways, Networks Layer Functions and Protocols, Routing, Routing Algorithms, Network Layer Protocol of Internet - IP Protocol, Internet Control Protocols.

Unit 4: Transport, Session, Presentation and Application Layer Protocol **[20]**

Transport Services- Error and Flow Control, Connection Establishment and Connection Release, Flow Control & Buffering, TCP/IP protocol suite, Concept of- TCP, UDP, IP, FTP, DNS, Telnet, SMTP, POP, HTTP, WWW, ARP, RARP.

Reference Books:

- B. A. Forouzan: Data Communications and Networking, Fourth edition, THM Publishing Company Ltd 2007.
- S. Tanenbaum: Computer Networks, Fourth edition, PHI Pvt. Ltd 2002

BCA-III (CBCS)						
Sem:	V					
Paper Type:	DSC 1F		**Paper No:**	Paper XV		
Paper Name:	**Data Warehousing and Data Mining**					
Credit:	04		**Theory:**	4 Hrs./week		
Marks:	**UA:**	80	**CA:**	20	**Total:**	100

Course Objectives:

1. To understand the principles of Data warehousing and Data Mining.
2. To understand the Architecture of a Data Mining system.
3. To perform classification, association, and prediction of data.

Course Outcomes: The students should be able to:

1. Identify data mining problems and implement the data warehouse.
2. Write association rules for a given data pattern.
3. Choose classification and clustering solutions.

Unit 1: Introduction to Data Warehouse and Data Mining [15]

Differences between Operational Database Systems and Data Warehouses, Data Warehouse Architecture, Data Warehouse Components, A Multidimensional Data Model, Schemas, Data Warehouse Implementation, Data cube Technology, OLAP operations, Data mining query language, Data Mining:- What is data mining, Evolution, KDD, What kind of data, Architecture, data mining views, Data Mining Functionalities, Issues in Data Mining.

Unit 2: Data Preprocessing and Association Rule mining [15]

Data Preprocessing:An Overview, Extract, Transform, Load (ETL) Processes, Data Cleaning, Data Integration, Data Transformation and Data Discretization, Data Reduction, Frequent Patterns,

Associations, and Correlations: Market Basket Analysis, Frequent Itemsets, Closed Itemsets, and Association Rules, Frequent Itemset Mining Methods-Apriori Algorithm: Finding Frequent Itemsets, Generating Association Rules from Frequent Itemsets, Improving the Efficiency of Apriori, A Pattern-Growth Approach for Mining Frequent Itemsets, Mining Multilevel and multidimensional Association Rules, Constraint-Based Frequent Pattern Mining

Unit 3: Supervised Learning Technique [15]

supervised and unsupervised learning, What Is Classification? What is regression, difference

between classification and regressing, General Approach to Classification, Issues regarding Classification and Predication, Binary and Multiclass Classification, Types of classifications, Classification by Decision tree induction, Bayesian Classification, Classification by Back propagation, Logistic regression, k-Nearest-Neighbor Classifiers, SVM, Introducing Ensemble Methods-Bagging, Boosting, AdaBoost, Random Forests, Other classification methods, Prediction: regression. Model Evaluation and Selection-Metrics for Evaluating Classifier Performance, Cross-Validation, underfitting and overfittin.

Unit 4: Unsupervised Learning Technique and Applications [15]

Clustering: What is Cluster Analysis, Types of data in Cluster Analysis, A Categorization of Major Clustering Methods., Partitioning Methods, Hierarchical Methods, Density-Based Methods, Model-Based Clustering Methods: Statistical Approach, Neural Network Approach, Outlier Analysis, Applications and Trends in Data Mining: Data Mining Applications, Data Mining for Financial Data Analysis, Data Mining for Retail and Telecommunication Industries, Data Mining in Science and Engineering, Data Mining for Intrusion Detection and Prevention, Data Mining and Recommender Systems, Spatial Data Mining. Text Data Mining, Multimedia Data Mining, Web Data Mining, Privacy, Security, and Social Impacts of Data Mining, Data Mining and Intelligent Query Answering, Trends in Data Mining.

Reference Books:

- Data Mining – Concepts and Techniques – Jiawei Han, Michelinen Kamber, Morgan Kaufmann Publishers, Elsevier, 2nd Edition, 2006.
- Introduction to Data Mining, Pang – Ning Tan, Vipin Kumar, Michael Steinbanch, Pearson Education.
- Data Warehouse Fundamentals, Pualraj Ponnaiah, Wiley Student Edition.

BCA-III (CBCS)						
Sem:	V					
Paper Type:	DSC 1G		**Paper No:**	Paper XI		
Paper Name:	**Dot NET Core**					
Credit:	04		**Theory:**	4 Hrs./week		
Marks:	**UA:**	80	**CA:**	20	**Total:**	100

Course Objectives:

1. To understand how to design, implement, test, debug, and document programs that use basic data types and computation, simple I/O, conditional and control structures, string handling and functions in C#.
2. To understand the importance of Classes and objects along with constructors, Arrays and Vectors in C#.
3. Discuss the principles of inheritance, interface and demonstrate though problem analysis assignments how they relate to the design of methods, abstract classes and interfaces and packages in C#.
4. To understand importance of Multi-threading and different exception handling mechanisms in C#.
5. To understand basic idea about how to design GUI base windows application using C#.

Course Outcomes: Students will be able to:

1. Gain proficiency in designing, implementing, testing, debugging, and documenting programs utilizing fundamental data types, computation, basic I/O, conditional and control structures, string manipulation, and functions within the C# programming language.
2. Appreciate the significance of classes, objects, constructors, arrays, and vectors within the C# framework.
3. Explore the principles of inheritance and interfaces, demonstrating their application through problem analysis assignments and their relevance to method design, abstract classes, interfaces, and packages in C#.
4. Recognize the importance of multi-threading and various exception handling mechanisms in C# programming.
5. Acquire a foundational understanding of designing GUI-based Windows applications using C#.

Unit 1: Introduction to C# **[20]**

Understanding .NET- The .NET Framework, .NET Core, Download and install C# Development Environments - Visual Studio, Visual Studio Code, building console apps using Visual Studio 2022 and Building console apps using Visual Studio Code, C# Basics- Variables and Data Types, Reference and Value Types-Nullable types, Elvis operator, Null coalescing operator, Boxing and unboxing, Keywords, Initialization, Type Inference, Console Input and Output., Operators, Operator precedence, Type conversion, C# statements- Branching, Jumping, Looping, Complex data types- Enums, Arrays, Tuples.

Unit 2: Object Oriented Programming **[20]**

Classes and object-Declaration, Access modifiers, Data, Methods, Method parameters, Constructors, Deconstruct, Method overloading, Properties, Local and global variable and methods, Static classes, methods and members, nested classes, Indexers, Partial types and methods, Structs and Records, Inheritance- Base and derived classes, advantages, Types. Constructors in inheritance. Abstract classes, sealed class, Interfaces - Defining and implementing, Default interface methods, Interface inheritance, .NET interfaces, Polymorphism- Virtual methods. Method overriding, operator overloading, Abstract methods, Sealed types.

Unit 3: Threading, Exception and Resource Management **[12]**

Exception- about exception, Exceptions Hierarchy, Throwing and Catching Exception, The try-finally Construct, IDisposable and the "using" Statement, Advantages of Using Exceptions, inbuild exception, custom exception, Threading-about threading, Thread Name, Thread Priority, and Thread State, Foreground and background threads in C#, Multithreading - An Overview, The Thread Class, ThreadPool Threads, Collections- Generic collections, Concurrent collections, Specialized collections, Performance considerations, Resource Management-Finalizers, Garbage Collection, IDisposable, The using statement, Serialization-Attributes, JSON serialization, Binary serialization, XML serialization

Unit 4:Delegate, Event and LINQ **[8]** Delegates-Multicast delegates, generic delegates, Action<T>, Predicate<T>, Func<T> , Lambdas-Expression and statement lambdas, Parameters, Return type, Captures, Events- Defining, Raising, Standard and custom events, LINQ- Enabling features, LINQ expression, LINQ pattern, Joins, Aggregations, Basic of Windows application.

Reference Books:

- Nagel, Karli Watson, Jay Glynn, Morgan Skinner, Bill Evjen.
- Inside C# - Microsoft Press by Tom Archer, Andrew Whitechapel.
- Programming Microsoft Visual C# 2005 - The Language (Microsoft Press) by Donis Marshal

BCA-III (CBCS)						
Sem:	V					
Paper Type:	DSE 1A		**Paper No:**	Paper XI		
Paper Name:	**Data Science using Python**					
Credit:	04		**Theory:**	4 Hrs./week		
Marks:	**UA:**	80	**CA:**	20	**Total:**	100

Course Outcomes:

1. Know basic data types in Python.
2. Know operators, and how to clean and merge datasets.
3. Know pandas library, the main methods for DataFrames.
4. Know how to import data in Python.
5. Know how to work in Spyder.

Unit-1: **Introduction to Data Science:** Evolution of Data Science – Data Science life cycle, Data Science Roles – Stages in a Data Science Project – Applications of Data Science in various fields – Data Security Issues.

Data Collection Strategies – Introduction to Collection of Data, Primary and Secondary Data, Methods of Collecting Primary Data, Methods of Secondary Data, Statistical Errors, Rounding off Data.

Data Pre-Processing Overview – Data Cleaning – Data Integration and Transformation – Data Reduction – Data Discretization, Outlier analysis, Testing and Training

Model design and Development: Model Evaluation using Visualization – Residual Plot – Distribution Plot –, Measures for In-sample Evaluation – Prediction and Decision Making. Generalization Error – Out-of-Sample Evaluation Metrics – Cross Validation – Over fitting – Under Fitting and Model Selection – Prediction by using Ridge Regression – Testing Multiple Parameters by using Grid Search.

Unit-2: I**ntroduction to NumPy:** Arrays and Vectorized Computation- The NumPy ndarray- Creating ndarrays- Data Types for ndarrays- Arithmetic with NumPy Arrays- Basic Indexing and Slicing - Boolean Indexing-Transposing Arrays and Swapping Axes.

Universal Functions: Fast Element-Wise Array Functions- Mathematical and Statistical

Methods-Sorting Unique and Other Set Logic.

I**ntroduction to pandas:** Series, DataFrame, Creating Data Frame from an Excel, .csv file, python Dictionary, python List and Tuples, Operations on Data Frames: Dropping Entries Indexing, Selection, and Filtering- Function Application and Mapping- Sorting and Ranking. Summarizing and Computing Descriptive Statistics- Unique Values, Value Counts, and Membership. Reading and Writing Data in Text Format.

Data visualization using Matplotlib and Seaborn: Basic plot-Line, Bar, scatter, subplot, Statistics plots-Box, Histogram, errorbar and pie, 3D plots-scatter, surface, triangular, Heatmap, Density Plots, CatPlot (Categorical Plot), Joint Distribution Plot.

Books Recommended:

1. Python Data Analytics, Fabio Nelli, Apress

2. Wes McKinney, "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython",O'Reilly, 2nd Edition,2018. Python for data science for dummies 2nd Edition, John Paul Mueller, Luca Massaron, and Wiley

3. Jake VanderPlas, "Python Data Science Handbook: Essential Tools for Working with Data",O'Reilly, 2017.

4. Programming through Python, M. T. Savaliya, R. K. Maurya, G. M. Magar, STAREDU Solutions

5. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Davy Cielen, Arno D.B. Meysman, et al., Mining

BCA-III (CBCS)						
Sem:	V					
Paper Type:	**DSE-1 B**		**Paper No:**	Paper XI		
Paper Name:	**Kotlin Programming**					
Credit:	04		**Theory:**	4 Hrs./week		
Marks:	**UA:**	80	**CA:**	20	**Total:**	100

Course Objective: -

Students will try to learn:

1. The Kotlin programming language.
2. Familiarize with Kotlin syntax, features, and basic programming concepts.
3. Explore language features such as type inference, null safety, and extension functions, and understand their importance in Kotlin programming.
4. Provide a solid foundation in object-oriented programming (OOP) concepts within the context of Kotlin which covers classes, objects, inheritance, interfaces, and other OOP principles, and demonstrate their implementation in Kotlin.
5. To introduce functional programming concepts in Kotlin, including higher-order functions, lambda expressions, and functional manipulation of collections and understand how functional programming enhances code readability, maintainability, and reusability.
6. Equip the skills to write Kotlin code for simple applications.
7. Lay the foundation for further exploration and application of Kotlin in software development.

Course Outcomes: -

Students will be able to:

8. Gain a comprehensive understanding of the Kotlin programming language.
9. Acquire familiarity with Kotlin syntax, features, and fundamental programming concepts.
10. Explore and grasp the significance of language features such as type inference, null safety, and extension functions in Kotlin programming.
11. Develop a strong foundation in object-oriented programming (OOP) principles within the Kotlin context, covering classes, objects, inheritance, interfaces, and other related concepts.

12. Introduce functional programming concepts in Kotlin, including higher-order functions, lambda expressions, and functional manipulation of collections, while understanding their impact on code readability, maintainability, and reusability.
13. Develop the skills necessary to write Kotlin code for simple applications.
14. Establish a groundwork for further exploration and utilization of Kotlin in software development endeavors.

Unit 1: - Introduction to Kotlin (15)

Introduction to Kotlin, Download and Install JDK, Download and Install IntelliJ Idea, Creating New Kotlin Project, Creating Kotlin File, main Function, Run Kotlin Application, Single Line and MultiLine Comment, Data Type- basic data types (Int, String, Boolean, etc.), Variable using var and val, type inference, Operators and expressions: Arithmetical Operators, Comparison Operators ,Equality and Inequality Operators ,Logical Operators ,Increment and Decrement Operators ,Augmented Assignments etc., Control flow: if Expression, if else Expression ,when Expression, for Loop ,while Loop ,do while Loop ,Break and Continue, Getting User Input using readLine, Getting User Input using Scanner, String ,String Concatenation ,String Literals - Escaped String ,String Literals - Raw String ,String Templates

Unit 2: - Functions and Lambdas (15)

Defining functions: function syntax, parameters, return types., Function without Parameters, Function with Parameters, Function with Default Argument, Function with Named Argument, Higher-order functions: passing functions as arguments, returning functions from functions. Lambda expressions: syntax, usage, capturing variables., Anonymous Function, Null Safety- Nullable and non-nullable types in Kotlin., Safe calls (?.), Elvis operator (?:), and the !! operator., Handling nullability: safe casts (as?), let, run, apply, also functions

Unit 3: - Collections (10)

Introduction to Kotlin collections: array lists, sets, maps or Dictionary, Basic operations on collections: iterating, adding, removing elements., Functional programming with collections: map, filter, reduce.

Unit 4: - Object-Oriented Programming (20)

Classes and objects: class declaration, properties, methods., Constructors: primary constructors, secondary constructors., Getter and Setter, Inheritance: superclass, subclass, Inheritance with Constructor, overriding methods., Properties and Function, Super, Visibility Modifiers, Abstract Class and Method, Interfaces: defining interfaces, implementing interfaces in classes., Data Class, Object Destructuring, Exception Handling: try-catch block, throw keyword, finally block., Kotlin's approach to checked and unchecked exceptions., File I/O:-Reading from and writing to files., Working with text files: reading lines, writing to files., Handling file exceptions.

References:

1. "Kotlin Programming: The Big Nerd Ranch Guide" by Josh Skeen and David Greenhalgh
2. "Kotlin in Action" by Dmitry Jemerov and Svetlana Isakova Online Kotlin documentation and tutorials.

BCA-III (CBCS)							
Sem:	V						
Paper Type:	**DSE-1 C**			**Paper No:**	Paper XI		
Paper Name:	**ReactJS**						
Credit:	04			**Theory:**	4 Hrs./week		
Marks:	**UA:**	80	**CA:**	20	**Total:**	100	

Course Objectives:

1. Understand Single Page React application is different than traditional web development frameworks.
2. Develop an application from scratch.
3. Understand the benefits of unidirectional data flow.

Course Outcomes:

1. To understand basic fundamentals of Front-end Technologies
2. To apply the advanced concepts of ReactJS and WebAPI
3. To develop ReactJS applications.

Unit 1: Introduction to ReactJS **[20]**

Introduction, Workflow, Scope, Pros and Cons, Difference between JS and JSX, React Components overview, Child Components, JSX expressions,

Building Blocks of ReactJS: JSX, Components, State and Props, Conditional Rendering, Why JSX, Advantages of JSX, Expressions in JSX, Implementation of JSX, Creating a react component with jsx

Environment Setups: Node setup, How to use NPM, Npm and Setting Environment for ReactJS projects, How to create package.json and purpose, IDE for ReactJS, ReactJS browser plugins overview.

Components: Types of components, Functional component vs Class Component, Converting Functional Components to Class Components, Component Life Cycles and its different methods.

Unit 2: Conditional Rendering and List **[10]**

if-else Statement, logical andand operator, operators, Preventing Component from Rendering, Switch case operator

List and Keys: react key prop, map function to iterate the List, References, use Refs, Create

Refs, access Refs, Event Binding types: Bind () method, Arrow function

Props and State: What is a state, use and role of the state, what are props, Props validation, Passing data between multiple components, Managing Component State

Unit 3: Handling Events and Forms [10]

Lists of Form components, Setup Controlled and Uncontrolled form components, Control Input elements, Form Submission and Validation, how to set default values on all formats of Input elements, Form validations, writing Styles, Animations overview, Event, Event Binding, Event Handlers, Common React Events, Key Events, Event Pooling, Synthetic Event.

Unit 4: Routing and State Management [20]

Introduction to React Router, History of Router, Single Page Application Overview, configure React Router, Load the router library, Navigating between Routes, Route Parameters and Nested Routes, Dynamic Routing, Nesting Routes, Invalid URL, Handle Conditional statement in JSX

State Management: Local State vs. Global State, State Lift-Up, Context API for Global State

Redux: Introduction to Redux, Redux Architecture- Actions, Reducers, and Store, Provider Component, Dispatchers, View Controllers, Connecting React with Redux

Hooks: Introduction to Hooks, The useState hook, useEffect hook, Custom hook, useRef hook, useMemo hook, The useContext hook, The useReducer hook, Another Hooks.

Reference Books:

- Introduction to React By Cory Gackenheimer, Apress
- React and React Native: A complete hands-on guide to modern web and mobile development with React.js By Adam Boduch, Roy Derks
- React 16 Essentials: A fast-paced, hands-on guide to designing and building scalable and maintainable web apps with React 16 By Artemij Fedosejev, Ada

BCA-III (CBCS)						
Sem:	V					
Paper Type:	**SEC-2**		**Paper No:**	Paper XI		
Paper Name:	**Advanced Java**					
Credit:	04		**Theory:**	4 Hrs./week		
Marks:	**UA:**	80	**CA:**	20	**Total:**	100

Course Objectives:

1. To learn GUI programming using swing Technology
2. To study database programming using Java.
3. To study web development concept using Servlet and JSP
4. To learn socket programming concept using Networking.

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

1. Design, create, build, and debug Java applications.
2. Explore Integrated Development Environment (IDE).

Unit 1: Networking and Database **[15]**

Networking: Basics, networking classes and interfaces, using java.net package, doing **Working with databases:** Steps for Connecting to databases, Types of Drivers, Handling Exceptions, Creating and Using Statement Objects, Using Statements to Insert, Update, Delete Data into a Database, Using the ResultSet Class, Data navigation, Prepared Statements, Callable Statements.

Unit 2: Introducing Servlet **[18]**

Advantages of Servlet over CGI, Features of Servlet, Introducing Servlet API, Javax.servlet package, Javax.servlet.http package, Introducing Servlet, Advantages of Servlet over CGI, Features of Servlet, Servlet life Cycle,Init(),Service(),Destroy(),Working with GenericServlet and HttpServlet, RequestDispatcher interface, Include() and forward() ,Use of RequestDispatcher, Session in Servlet, Introducing session, Session tracking mechanism, Cookies, Advantages and disadvantages,use of cookies, Hidden form filed, Advantages and

disadvantages, use of Hidden form filed, URL rewritten, disadvantages, use of URL rewritten, HttpSession, Advantages and disadvantages, use of URL HttpSession

Unit 3: Java Server Pages **[12]**

Introduction, Advantages of JSP over Servlet, JSP architecture , JSP life cycle, Implicit objects in JSP- request, response, out, page, page Context,
application, session, config, exception, JSP tag elements- Declarative, Declaration, scriplet, expression, action. Java Bean- Advantages and Disadvantages, Use Bean tag- setProperty and getProperty, Bean In Jsp, JSTL core tag: General purpose tag, conditional tag, networking tag, JSTL SQL tags, JSTL formatting tags, JSTL xml tags, Custom tag: empty tag, body content tag, iteration tag, simple tag , Introducing internationalization and Java: local class, ResourceBundle class, Handling different Error and Exceptions.

Unit 4: Hibernate and Spring **[15]**

Introduction Hibernate (HB), Architecture of HB, Application of HB, HB with annotation, HB web application, Inheritance mapping: Table per Hierarchy, (TPH), TPH using annotation, Table Per, Concrete (TPC), TPC using annotation, Table Per Subclass (TPS), TPS using annotation, Collection mapping: Mapping list, one to many by list, one to many by bag, one to many by set, one to many by map.

Spring: Introduction to spring, Spring modules, Spring application, Dependency injection: constructor Injection (CI), CI dependant object, CI with collection, CI with map, CI inheriting bean, Spring JDBC: JDBC template, PreparedStatement, ResultsetExactor, RowMapper, NamedParameter, Simple JDBC template, Spring with Hibernate.

Reference books:

- Java the complete Reference by Herbert Schildt
- Java Servlet Programming by Jasan Hunter
- Java Server Programming by Bayross and Shah
- JDBC, Servlet and JSP Black Book- Santosh Kumar K.
- Hibernate- Sharanam Shah and Vaishali Shah
- Spring Persistence with Hibernate- Paul Tepper Fisher, Brian D Murphy

BCA-III (CBCS)					
Sem:	VI				
Paper Type:	**DSC 2E**		**Paper No:**	Paper XII	
Paper Name:	**Network Security**				
Credit:	04		**Theory:**	4 Hrs./week	
Marks:	**UA:**	80	**CA:**	20	**Total:** 100

Course Objective: -

Students will try to learn:

1. Introduce students to the fundamental concepts of security, including its necessity, approaches, principles, and various types of security attacks.
2. Provide an overview of cryptography, including plaintext and ciphertext, substitution and transposition techniques, encryption and decryption processes.
3. Introduce symmetric and asymmetric key cryptography algorithms, steganography, key range, key size, and potential types of cryptographic attacks.
4. Investigate Internet security protocols such as Secure Socket Layer/Transport Layer Security (SSL/TLS), Secure Electronic Transaction (SET), and email security protocols like PGP and S/MIME.
5. Cover user authentication basics, including passwords, smart cards, and biometrics, as well as authentication protocols like Kerberos as well as discuss network security principles, firewalls, types of firewalls, and IP security mechanisms to safeguard networks against unauthorized access and malicious activities.

Unit 1: - Fundamental Security Concepts (10)

Security Concepts: Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks – Active and Passive, Security services, Security Mechanisms, A model for Network Security , Access Control Mechanisms: Access Matrix, ACL and capabilities, Access Control Models,.

Unit 2: - Cryptography Concepts and Techniques (20)

Introduction to Cryptography, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, possible types of attacks, Symmetric Key Cryptographic Algorithms: Algorithm Types and Modes, An overview of Symmetric Key Cryptography, DES, International Data Encryption Algorithm (IDEA), RC5, Blowfish, AES, Asymmetric Key Cryptography: Brief History of Asymmetric Key

Cryptography, An overview of Asymmetric Key Cryptography, The RSA Algorithm, Symmetric and Asymmetric Key Cryptography Together

Unit 3: - Digital Signatures and Internet Security Protocols (20)

Digital Signatures: Introduction, Message digests, MD5, SHA-512,MAC, HMAC, Knapsack Algorithm, Elliptic curve Technology, ELGamal Algorithm. , Internet Security Protocols: Secure Socket Layer/TLS, Secure Electronic Transaction, SSL versus SET, E-mail Security- PGP, S/MIME.,

Unit 4: - User Authentication and Network Security (10)

User Authentication and Kerberos: Authentication basics, Passwords, use of smart cards, Biometrics, Kerberos., Network Security: Firewalls, types of firewalls, IP Security, Intrusion : Intruders, Audit Records, Intrusion Detection, honey pots.

Reference Books:

1. Cryptography and Network Security by Atul Kahate, Tata McGraw-Hill
2. Cryptography and Network Security by Behrouz A. Forouzan, Debdeep Mukhopadhyay, Special Indian Edition, Tata McGraw-Hill.
3. Network Security Essentials: Applications and Standards by William Stallings, Pearson Education.
4. Fundamentals of Computer Security Technology: Edward Amoroso, Prentice-Hall.
5. Cryptography and Data Security: Dorothy E. Denning, Addison-Wesley.
6. Cryptography -Theory and Practice: Douglas R. Stinson, CRC Press.
7. Building Internet Firewalls: D. Brent Chapman and Elizabeth D. Zwicky, O'Reilly andAssociates.

BCA-III (CBCS)							
Sem:	VI						
Paper Type:	**DSC 2F**			**Paper No:**	Paper XII		
Paper Name:	**Data Visualization using PowerBI**						
Credit:	04			**Theory:**	4 Hrs./week		
Marks:	**UA:**	80	**CA:**	20	**Total:**	100	

Course Objective: -

Students will try to learn:

1. To gain proficiency in navigating the Power BI interface, understanding its components, and utilizing its basic features for data visualization.
2. To learn how to import, clean, and transform data from various sources into formats suitable for visualization in Power BI.
3. To understand the importance of data modeling techniques such as creating relationships, measures, and calculated columns for effective analysis.
4. To master the creation of various types of visualizations, including bar charts, line charts, scatter plots, and maps, using Power BI's intuitive tools.
5. To explore customization options to enhance visual appeal and effectively convey insights to stakeholders.
6. To learn to design interactive dashboards by combining multiple visualizations, slicers, filters, and drill-down functionalities.
7. To understand principles of dashboard layout, organization, and storytelling to create engaging and actionable presentations of data.
8. To understand the process of sharing Power BI reports and dashboards securely with colleagues and stakeholders.

Unit 1: - Introduction to Power BI **(10)**

Overview of Business Intelligence, BI Uses and Users, Various BI Tools, Why Power BI, Introduction to Power BI, Features of Power BI, Power BI Components, Building Blocks of Power BI, Architecture of Power BI, Power BI Desktop Installation, Loading and Transforming dataset: Data Sources-File Sources, Databases, Azure, Other Sources, Loading Data-Web Pages, CSV Files, Text Files, XML Files, Excel, Microsoft Access Databases, SQL Server and other databases, Refreshing Data.

Unit 2: - Creating a Data Model **(10)**

Data Modeling in the Power BI Desktop Environment-The Power BI Desktop Data View, Data Models, Managing Power BI Desktop Data- Manipulating Tables, Manipulating Columns, Power BI Desktop Data Types, Formatting Power BI Desktop, Data Currency Formats, Preparing Data for Dashboards, Categorize Data, Apply a Summarization, Define Sort by Columns, Sorting Data in Power BI Desktop Tables, Adding Hierarchies, Designing a Power BI Desktop Data Model-Data View and Relationship View, Creating and Deleting Relationships Manually and Automatically

Unit 3: - Transforming Datasets **(20)**

Editing Data After a Data Load, Transforming Data Before Loading, Dataset Shaping- Renaming Columns, Reordering Columns, Removing Columns, Merging Columns, Duplicating Columns, Splitting Columns, Removing Records, Removing Duplicate Records, Sorting Data, Reversing the Row Order, Filtering Data-Selecting Specific Values, Finding Elements in the Filter List, Filtering Text Ranges, Filtering Numeric Ranges, Filtering Date and Time Ranges, Data Cleansing: Viewing a Full Record, Changing Data Type , Detecting Data Types, Replacing Values, Transforming Column Contents, Filling Down, Using the First Row As Headers, Grouping Records, Extending Data, Appending Data, Merging Data- Adding Data, Aggregating Data During a Merge Operation, Extending the Data Model with Calculated Columns, Creating Custom Columns, Index Columns, Types of Join- Joining on Multiple Columns, Preparing Datasets for Joins, Correct and Incorrect Joins, Examining Joined Data, The Expand and Aggregate Buttons.

Unit 4: - Power Query Editor and Data Visualizations **(20)**

Power Query Editor - What is DAX, Different type of DAX functions-Aggregate functions, Date functions, Logical functions, Math functions, String functions, Trigonometric functions and other functions. Adding Measures to the Data Model Basic Aggregations in Measures, Using Multiple Measures, Cross-Table Measures, More Advanced Aggregations, Filtering Data in Measures, Analyzing Data over Time. Data Visualizations Charts in Power BI-Types of charts, Maps in Power BI, Table and Matrix in Power BI, Subtotal and Total in Matrix, Cards and Filters in Power BI, Conditional Formatting, Slicers in Power BI- slicers, adding a Slicer, Applying Slicers, clearing a Slicer, deleting20 34 / 52 a Slicer, modifying a Slicer, Formatting Slicers-Slicer Orientation, Modifying the Outline, Adjusting Selection Controls, Setting the Exact Size and X and Y coordinates of a Slicer, Slicer Header, Slicer Items Designing Power BI Dashboards and Reports Dashboards, reports, Dashboards versus reports, Dashboard design-What is KPI, Whento use KPI, Requirements for KPI, KPI Visualizations, Visual selection, Layout, Navigation pane, Full screen mode, Supporting tiles, Custom date filters, Single- dashboard, Multiple-dashboard, Organizational dashboards, Multiple datasets Dashboard tiles- Tile details and custom links, Images and text boxes, SQL Server Reporting Services. Deploying the Power BI Report Server Live

Dashboard pages, Live report pages, Mobile-optimized dashboards Case study – Superstore, IPL Analysis, Product Sales Data Analysis, Marketing Campaign Insights Analysis, Financial Performance Analysis, Loan Application Analysis

Reference Books: -

1. Pro Power BI Desktop-Free interactive data analysis with Microsoft Power BI by AdamAspin, Apress
2. Introducing Microsoft Power BI by Alberto Ferrari and Marco Russo, Microsoft Press
3. Mastering Microsoft Power BI by Brett Powell, Packt BIRMINGHAM– MUMBAI
4. Microsoft Power BI Complete Reference by Devin Knight, Brian Knight, Mitchell Pearson, Manuel Quintana, Brett Powell, Packt
5. Learn Power BI by Greg Deckler,Packt

BCA-III (CBCS)							
Sem:	VI						
Paper Type:	**DSC 2G**			**Paper No:**	Paper XII		
Paper Name:	**ASP.Net Core**						
Credit:	04			**Theory:**	4 Hrs./week		
Marks:	**UA:**	80	**CA:**	20	**Total:**	100	

Course Objective: -

Students will try to learn:

1. Demonstrate the creation of ASP.NET Core MVC Web Applications using .NET 8, covering project file structure, main method, hosting options, and configuration files.
2. Cover Models, Controllers, Views, and Dependency Injection in ASP.NET Core MVC applications.
3. Introduce Entity Framework Core and guide the installation process also explain DbContext in Entity Framework Core and database connection string configuration and database operations.
4. Discuss Transactions, Migration, and Database Seeding in Entity Framework Core.
5. Introduce Partial Views, View Components, and Razor View Engine.
6. Cover Action Results, Routing, Model Binding, HTML Helpers and Tag Helpers in ASP.NET Core MVC.
7. Explain Data Annotations and Model Validations, including custom validations and remote validation.
8. Discuss different methods of State Management, including Cookies and Sessions.

Unit 1: - Introduction to ASP.Net Core MVC (10)

Overview of Microsoft Web Technologies, Introduction to ASP.NET Core Framework. NET Core Environment Setup, Download and Install Visual Studio 2022, Download and Install .NET Core SDK, Download and Install SQL Server 2022, Download and Install SSMS, Creating ASP.NET Core Web Application using .NET 8, NET Core Project File Structure, NET Core Main Method, NET Core InProcess Hosting, OutOfProcess Hosting, LaunchSettings.json File, AppSettings.json file, Middleware Components, Web Root (wwwroot) Folder, Static Files Middleware, Configuring Default Page, Developer Exception Page Middleware Command Line Interface, Project Templates in ASP.NET Core Application, Introduction to ASP.NET Core MVC Framework, Set up MVC in ASP.NET Core, Models, Controllers and Views in ASP.NET Core MVC, ASP.NET Core Dependency Injection, Creating ASP.NET Core

Application using MVC

Unit 2: - Entity Framework Core (15)

Introduction to Entity Framework Core, How to Install Entity Framework Core, DbContext in Entity Framework Core, Database Connection String in Entity Framework Core, CRUD Operations in Entity Framework Core, Entity States in Entity Framework Core, Data Annotation Attributes in Entity Framework Core- Table Attributes, Column Attributes, Key Attribute, ForeignKey Attribute, Index Attribute, InverseProperty Attribute, NotMapped Attribute, Required Attribute, MaxLength and MinLength Attribute, Database Generated Attribute, TimeStamp Attribute, ConcurrencyCheck Attribute, Relationships in Entity Framework Core- One-to-One Relationships, One-to-Many Relationships, Many-to-Many Relationships, Self-Referencing Relationship, Asynchronous Programming with Entity Framework Core, Disconnected Entities in Entity Framework Core, Stored Procedures in Entity Framework Core, Transactions in Entity Framework Core, Migration in Entity Framework Core, Database Seedd in Entity Framework Core, Entity Framework Core Database First Approach

Unit 3: Model, View, Controller and Routing (20)

ViewData, ViewBag, Strongly Typed View, ViewModel, TempData, Post-Redirect-Get (PRG) Pattern Example, Layout View, Sections in Layout View, ViewStart, ViewImports, Partial Views, Different Ways to Render Partial View, View Components, Razor View Engine and Razor Syntax, How to Install and use Bootstrap in ASP.NET Core MVC, Action Results in ASP.NET Core MVC- Action Results, View Result, Partial View Result, JSON Result, Content Result, File Result, Redirect Results, Status Results, Object Result, EmptyResult , Routing in ASP.NET Core MVC, Custom Routing, Custom Route Constraints in Web Application, Attribute Routing, Attribute Routing using Tokens, Attribute Routing vs Conventional Routing, Model Binding in ASP.NET Core MVC, Model Binding using- FromForm, FromQuery, FromRoute, FromHeader, FromBody, Complex Type, Custom Model Binding in ASP.NET Core MVC

Unit 4: - HTML, Tag Helper, Data Annotation Validation and State management (15)

HTML Helpers for-TextBox, TextArea, DropDownList, RadioButton, Check Box, ListBox, Password, Hidden, Custom HTML Helper in ASP.NET Core MVC, Creating Form Using HTML Helpers, Different Ways to Generate Links in ASP.NET Core MVC, Tag Helpers for- Image Tag , Environment Tag, Navigation Menus, Form Tag, Partial Tag, Creating Custom Tag Helper, View Component Tag Helper, Cache Tag Helper, Data Annotations, Model Validations, Data Annotation Attributes- Custom Data Annotation, Remote Validation, Blacklist and Whitelist Checks using Data Annotation, Displaying and Formatting Attributes, Real-Time Examples of Data Annotations in ASP.NET Core MVC, Cookies, Encrypt Cookies, Persistent vs Non-Persistent Cookies, Sessions, In-Memory vs Distributed Sessions, Differences Between Cookies and Sessions, Upload File, Restrict Uploaded File Size, Restrict Uploaded File Type, Save Uploaded file to Database, Display Images, Delete Images, Upload Multiple Files, Export

Data to Excel File, Import Excel Data to Database, Generate PDF, Generate Password Protected PDF, Convert HTML to PDF, Send Email with Attachment

Reference Books:

1. Pro ASP.NET Core MVC by ADAM FREEMAN, Apress
2. ASP.NET MVC with Entity Framework and CSS by Lee Naylor, Apress
3. ASP.NET MVC Core 2.0 Cookbook by Engin Polat & Stephanen Belkheraz, Packt Publishing;
4. ASP.NET Core in Action by Andrew Lock
5. ASP.NET Core Application Development: Building an application in four sprints (Developer Reference) by James Chambers, David Paquette & Simon Timms, Microsoft Press
6. Pro ASP.NET Core MVC by ADAM FREEMAN, Springer Nature
7. Professional ASPNET MVC 5 by Galloway Jon. Matson David. Wilson Brad. Allen K Scott, wiley

BCA-III (CBCS)							
Sem:	VI						
Paper Type:	**DSE 2-A**			**Paper No:**	Paper XII		
Paper Name:	**Linux and Shell Programming**						
Credit:	04			**Theory:**	4 Hrs./week		
Marks:	**UA:**	80	**CA:**	20	**Total:**	100	

Course Objective: -

Students will try to learn:

1. To introduce Basic Linux general purpose Commands
2. To learn different editor
3. To learn shell script concepts.
4. To learn file management and permission advance commands.
5. To learn awk, grap, perl scripts.

Unit 1: - Introduction of Linux (10)

History of Linux, Architecture of Linux system & features, Kernel, Shell & its type, Difference between Windows and Linux. Linux Distributions, Working environments: KDE, GNOME, Xface4, Hardware requirement, Installation procedure of Linux, Create partitions, Configuration of X system Users & Groups Management:- Create Users, Create groups, Special groups, Assigning permissions to users and Groups, File and Directory permissions- chmod, chown, chgrp., Linux File System:-Hierarchy of File system, File System parts- Boot Block, Super Block, Inode, Block, Data Block, File types, Devices and Drives in Linux, Mounting devices (CD/DVD, usb, hard drive partition), file system

Unit 2: - Linux Command (15)

Linux commands File and directory Management Commands:-mkdir, rmdir, cd and pwd, file, ls, cat, more, less, File and Directory Operations: find, cp, mv, rm, ln etc, Printing the files - lpr, lpq, lprm etc., Filter Commands & Editor:- Filters: head, tail , pr, cut, paste, sort, uniq, tr, grep, egrep, fgrep, sed., Communication commands:- mesg, talk, write, wall, mail., Text Editors- vi, vim, Archive and File compression commands,

Unit 3: - Linux System Management and Administration (15)

Process Management: Shell process, Parent and children, Process status, System process, Multiple jobs in background and foreground, Changing process priority with nice. Listing processes, ps, kill, premature termination of process., Disk management and System Administration:-Disk Partitioning- RAID, LVM

etc., disk related Management Tools- Fdisk, Parted etc. , Boot Loaders- GRUB, LILO, Custom Loaders, System administration – Role of system administrator, identifying administrative tasks & files, Configuration and log files, Chkconfig, Security Enhanced Linux, Installing and removing packages with rpm command, Understanding various Servers:- DHCP, DNS, Squid, Apache, Telnet, FTP, Samba.

Unit 4: - Shell Programming (20)

Introduction to shell scripting, Writing and executing simple shell scripts, Variables, data types, and operators in shell scripting, Meta characters, Control structure, Loop structure and case statement, Writing and using shell functions, Passing arguments to shell scripts and functions, Returning values from functions, I/O and Redirection, Piping,

Course Outcomes: -

Students will be able to:

1. Identify the basic Linux general purpose commands.
2. Apply and change the ownership and file permissions using advance Linux commands.
3. Use the awk, grep, perl scripts.
4. Implement shell scripts.
5. Apply basic of administrative task.

Reference Books :

1. Official Red Hat Linux Users guide by Redhat, Wiley Dreamtech India
2. Beginning Linux Programming by Neil Mathew & Richard Stones, Wiley Dreamtech India
3. Red Hat Linux Bible by Cristopher Negus, Wiley Dreamtech India
4. UNIX Shell Programming by Yeswant Kanethkar, BPB
5. Shell Scripting: Expert Recipes for Linux, Bash, and More" by Steve Parker
6. Classic Shell Scripting" by Arnold Robbins and Nelson H.F. Beebe
7. Learning the bash Shell: Unix Shell Programming" by Cameron Newham and Bill Rosenblatt

BCA-III (CBCS)						
Sem:	VI					
Paper Type:	DSE 2 B		**Paper No:**	Paper XVI		
Paper Name:	Mobile Application Development using Kotlin					
Credit:	04		**Theory:**	4 Hrs./week		
Marks:	**UA:**	80	**CA:**	20	**Total:**	100

Course Objective: -

Students will try to learn:

1. Student will understand Set up of Android Studio and understand a development environment for Android app development using Kotlin.
2. Students will able to Create interactive and user-friendly UIs by designing layouts, using XML, and incorporating UI elements such as buttons, text fields, and lists.
3. Students will able to understand user interaction by handling touch events, button clicks, and using dialogs to provide feedback and information.
4. Students will able to Develop multi-screen apps by understanding the concepts of activities, intents, and fragments for effective navigation
5. Student understands data passing between screens using intents, bundles, and fragment communication.
6. Students will learn about data storage options in Android, including SharedPreferences for simple data storage and SQLite databases for more complex data management.
7. Student will Understand the process of preparing and deploying an Android app to the Google Play Store.

Course Outcomes

Students will be able to:

8. Understand the setup process of Android Studio and establish a development environment for Android app development using Kotlin.
9. Create interactive and user-friendly user interfaces (UIs) by designing layouts using XM and incorporate various UI elements such as buttons, text fields, and lists to enhance the user experience.
10. Implement responsive UI elements that enhance user engagement and interaction.
11. Implement effective navigation between screens to provide seamless user experiences within the app.
12. Understand and implement data passing between screens using intents, bundles, and fragment communication.

13. Implement data storage solutions that suit the requirements of the app and ensure data integrity and security.
14. Learn the steps involved in deploying the app to the Google Play Store and ensure compliance with store guidelines and policies.

Unit 1: - Introduction to Android App Development (20)

Overview of Android app development and Android Architecture, Setting up Android Studio., Creating a simple "Hello World" app., Understanding the Android project structure., User Interface (UI) Design and Layouts, UI Design Principles-Material Design guidelines, UI best practices, Layouts and Views and view groups-Introduction to Android layouts, Creating layouts in XML, Designing Interactive views-Buttons, text fields, labels, list, checkbox, spiner, radio and radio group, Handling user interaction, Building Complex UIs-Lists and RecyclerView, Custom UI components

Unit 2: - User Interaction and Navigation (15)

Handling User Input-Responding to button clicks and touch events, Toast messages and dialogs, android components activity, service, content provider, broadcost receiver, Navigating Between components, Creating multiple screens in your app, Fragment-Based UI, Introduction to Fragments, Building flexible UIs with fragments, Passing Data Between Screens, Sending data with Intents, Fragment communication

Unit 3: - Data Storage and Retrieval (15)

Introduction to Data Storage, Overview of data storage options in Android, Choosing the right storage method, Shared Preferences, Using Shared Preferences for simple data storage, Saving and retrieving data, SQLite Database, Introduction to SQLite databases in Android, Performing database operations, Integrating Data Storage, Incorporating data storage into your app, Best practices for data handling

Unit 4: - App Deployment and Finalization (10)

App Deployment, Preparing the app for the Google Play Store, Generating a signed APK, App Finalization, App optimization and performance, Gathering user feedback and making improvements, App Presentation, Final app project presentations, Sharing experiences and challenges, Creating SMS sending application, email application, google map application

Reference Books: -

1. "Android App Development" by Reto Meier
2. "Kotlin for Android App Development" by Peter Sommerhoff

3. "Android User Interface Design: Turning Ideas and Sketches into Beautifully Designed Apps" by Ian G. Clifton
4. "Android Programming for Beginners" by John Horton
5. "Android Fragments" by Dave MacLean
6. "Android Programming: The Big Nerd Ranch Guide" by Bill Phillips and Chris Stewart
7. "Android SQLite Essentials" by Sunny Kumar Aditya

BCA-III (CBCS)							
Sem:	VI						
Paper Type:	**DSE 2C**			**Paper No:**	Paper XII		
Paper Name:	**Internet of Things**						
Credit:	04			**Theory:**	4 Hrs./week		
Marks:	**UA:**	80	**CA:**	20	**Total:**	100	

Course Objectives:

1. To understand the concept of IoT and its applications.
2. To reduce import dependency on IoT components and promote indigenization.
3. To Understand the the fundamentals of IoT with its architecture, and protocols.
4. It also covers the overview and programming of two widely used IoT platforms Arduino and Raspberry Pi.

Course Outcome: Students should be able to

1. Demonstrate the architecture and functioning of IoT systems including the sensors and microcontrollers with their interfacing and software needs considering application areas.
2. Diagnose the various IoT protocols with detailing of their elements and overall functioning within IoT systems for efficient communication.
3. Design an IoT system to take the benefit of the Clouds for computing and storage considering security issues.
4. Leverage the benefits of IoT technologies for automating the various real-life challenges in various application areas.
5. Develop the software components of the IoT system using Arduino/Raspberry Pi Programming.

Unit 1: Introduction **[20]**

What is IoT, why important, Elements of an IoT ecosystem, Technology drivers, Business drivers, Trends and implications, Application areas of IoT, Characteristics of IoT, Things in IoT, IoT stack, Enabling technologies, IoT challenges, IoT levels, IoT and cyber-physical system, IoT and WSN, Overview of Governance, Privacy and Security Issues.

IOT ARCHITECTURE: IoT Open source architecture (OIC)- OIC Architecture and Design principles- IoT Devices and deployment models- IoTivity : An Open source IoT stack - Overview-

IoTivity stack architecture- Resource model and Abstraction.

Sensors, Microcontrollers, and Their Interfacing: Sensor interfacing, Types of sensors, Controlling sensors, Microcontrollers, ARM.

Unit 2: Protocols **[10]**

Protocol Standardization for IoT – Efforts – M2M and WSN Protocols – SCADA and RFIDProtocols – Issues with IoT Standardization – Unified Data Standards – Protocols – IEEE802.15.4–BACNet Protocol– Modbus – KNX – Zigbee– Network layer – APS layer – Security.

Unit 3: Cloud and Applications **[10]**

IoT and cloud, Fog computing, Security in cloud, Case study.

Application Building with IoT: Various applications of IoT: Food, Healthcare, Lavatory maintenance, Water quality, Warehouse, Retail, Driver Assistance, and Collision impact**,** IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications. Study of existing IoT platforms /middleware, IoT- A, Hydra etc.

Unit 7: Arduino and Raspberry Pi **[20]**

Arduino: Architecture, Programming and Application Raspberry Pi : Architecture, Programming and Application.

IoT Security: Various security issues and needs, architecture, requirements, challenges and algorithms.

Reference Books:

- Internet of Things, Vasudevan, Nagrajan and Sundaram, Wiley India
- IoT Fundamentals, David Hence at el, Cisco Press
- 21 IoT Experiments, Yashavant Kanetkar, Shrirang Korde, BPB
- IoT Based Projects, Rajesh Singh at el, BPB
- Internet of Things with ARDUINO and BOLT, Ashwin Pajankar, BPB

Credits: Practical's – (4)

Total Lectures: 60 Hrs. Contact Hrs. (Pr):5

University Evaluation: 80 Marks Internal Evaluation: 20 Marks

Project (Practical-IV)

Instructions:

1. Team size for major project not exceed than two students.
2. Real time and live project followed by Presentation and Viva-Voce.

Credits: Practical's – (4)

Total Lectures: 60 Hrs. **Contact Hrs. (Pr):5**

University Evaluation: 80 Marks **Internal Evaluation: 20 Marks**

DSC1F (Tools: Weka or Python)

Practical based on DSC 1E (Data warehousing and Mining)

1. What options are available on main panel? Describe and create the arff and csv file format.
2. Load iris dataset. How many instances are this dataset have? How many attributes? What is the range of possible values of the attribute petallength?
3. Explore the dataset using various visualization tools available in Weka.
4. Analyze basic statistics of the dataset such as mean, median, standard deviation, etc.
5. Identify any missing values and outliers in the dataset.
6. Preprocess the dataset by handling missing values, outliers, and noise.
7. Perform attribute selection to identify the most relevant features.
8. Normalize, discretize or standardize numerical attributes.
9. Load the weather.nominal dataset. Use the following filters in weka.
    a. unsupervised.instance.RemoveWithValues to remove all instances in which the humidity attribute has the value high.
    b. Convert numeric value to nominal
    c. Convert nominal to string
    d. Discretizes data
10. Load the iris dataset. Use the following filter of weka.
    a. Add noise to last column (i.e. Class).
    b. Randomize the data
    c. Normalize the data
    d. Reorder the data
11. Implement and evaluate different classification algorithms available in Weka (e.g., Decision Trees, Naive Bayes, k-Nearest Neighbors).
12. Train the models using the training set and evaluate their performance on the testing set using various metrics like accuracy, precision, recall, and F1-score.
13. Compare the performance of different classifiers and identify the most suitable one for the dataset.
14. Apply regression algorithms available in Weka (e.g., Linear Regression, Polynomial Regression).
15. Train the regression models using the training set and evaluate their performance on the testing set using metrics like Mean Squared Error (MSE) and R-squared.
16. Implement and evaluate clustering algorithms available in Weka (e.g., k-Means, Hierarchical Clustering).
17. Explore different clustering techniques and their impact on clustering quality.

18. Visualize the clusters using Weka's visualization tools and analyze their characteristics.
19. Apply association rule mining algorithms (e.g., Apriori) to identify interesting patterns in the dataset.
20. Adjust parameters such as minimum support and confidence to control the quality of the discovered rules.
21. Interpret and analyze the discovered rules to gain insights into the dataset.
22. Implement feature selection techniques to identify the most informative features in the dataset.
23. Evaluate the impact of feature selection and dimensionality reduction on the performance of classification or clustering algorithms.
24. Implement ensemble learning techniques (e.g., Random Forest, AdaBoost) using Weka's ensemble classifiers.
25. Compare the performance of ensemble methods with individual classifiers and analyze the benefits of ensemble learning.

Practical based on DSC 1G (Dot NET Core)

1. Create a simple console application that prints "Hello, World!" to the console.
2. Compile and run the application using the .NET Core CLI or Visual Studio.
3. Perform basic arithmetic operations (addition, subtraction, multiplication, division) on numeric variables.
4. Use string interpolation or concatenation to display variable values.
5. Implement conditional statements (if-else, switch-case) to control program flow based on different conditions.
6. Use loops (for, while, do-while) to iterate over arrays, collections, or sequences of data.
7. Create nested loops and conditional statements for more complex control flow logic.
8. Declare and initialize arrays of different data types.
9. Access array elements using index notation and perform array manipulation operations (sorting, searching, etc.).
10. Define and call methods with different access modifiers (public, private, protected).
11. Pass parameters to methods and return values from methods.
12. Overload methods with different parameter types and number of parameters.
13. Create classes and objects to represent real-world entities.
14. Create class library for checking Odd number, even number, prime number, Armstrong number etc.
15. Implement encapsulation, inheritance, and polymorphism concepts in C#.
16. Use constructors, properties, and methods to define the behavior of objects.
17. Implement try-catch blocks to handle exceptions and prevent application crashes.
18. Throw custom exceptions to handle specific error conditions.
19. Create custom exception and use them.
20. Use finally blocks to execute cleanup code regardless of whether an exception is thrown.
21. Read from and write to text files using StreamReader and StreamWriter classes.

22. Implement file input/output operations such as reading, writing, appending, and deleting files.
23. Handle file exceptions and ensure proper resource management using IDisposable interface.
24. Use LINQ queries to perform filtering, sorting, grouping, and aggregation operations on collections.
25. Create and manage multiple threads using the Thread class or ThreadPool.

Practical based on DSE 1 B (Kotlin Programming)

1. Compile and run the program using the Kotlin compiler or an Integrated Development Environment (IDE) like IntelliJ IDEA.
2. Declare variables of different data types (Int, Float, Double, String, Boolean).
3. Perform basic arithmetic operations (addition, subtraction, multiplication, division) on numeric variables.
4. Implement conditional statements (if-else, when) to control program flow based on different conditions.
5. Use loop structures (for, while, do-while) to iterate over arrays, lists, or ranges of values.
6. Create nested loops and conditional statements for more complex control flow logic.
7. Define and call functions with different parameters and return types.
8. Implement higher-order functions that take other functions as parameters or return functions as results.
9. Use lambda expressions to define anonymous functions and pass them as arguments to other functions.
10. Create and manipulate collections such as lists, sets, maps, and arrays in Kotlin.
11. Perform common operations on collections, such as iteration, filtering, mapping, and folding.
12. Use Kotlin's standard library functions (e.g., filter, map, reduce) to process collections efficiently.
13. Define classes and objects to represent real-world entities in Kotlin.
14. Implement inheritance, polymorphism, and encapsulation principles in Kotlin classes.
15. Use data classes and object declarations to create simple data structures and singleton objects.
16. Understand Kotlin's null safety features and how to handle nullable types using safe calls (?.), null checks, and the elvis operator (?:).
17. Use the safe cast operator (as?) and smart casts to safely cast nullable types to non-nullable types.
18. Create extension functions and extension properties to add new functionality to existing classes without modifying their source code.
19. Overload operators such as plus (+), minus (-), times (*), and equals (==) to define custom behavior for user-defined types.
20. Read from and write to files using Kotlin's standard library functions for file I/O.
21. Handle exceptions and errors using try-catch blocks and the throw keyword to gracefully handle unexpected conditions in Kotlin programs.
22. WAP which will demonstrate use of abstract class.
23. WAP to demonstrate interface and Multiple interfaces in kotlin.
24. WAP to implement following data classes-

a. equals (), b. Hash Code () c. To String () d. Copy ().

25. WAP to which demonstrate all exception handler

Practical Assignments on DSE 1C: (ReactJS)

1. How useEffect works in ReactJS?
2. How to concatenate unicode and variable in ReactJS?
3. How to pass data from one component to other component in ReactJS?
4. How to set input box to be a floating number in ReactJS?
5. How to use useState in arrow function instead of hook?
6. How to add theme to your React App?
7. How to add a function in JSX?
8. How to fetch data from APIs using Asynchronous await in ReactJS?
9. How to get cell value on React-Table?
10. How to access nested object in ReactJS?
11. How to set default value in select using ReactJS?
12. How to change the position of the element dynamically in ReactJS?
13. How to solve too many re-renders error in ReactJS?
14. How to publish a ReactJS component to NPM?
15. How to change the navbar color when you scroll in ReactJS?
16. How React Native is different from ReactJS?
17. How to Create a Countdown Timer Using ReactJS?
18. How to Create a Navigation Bar with Material-UI?
19. How to locally manage component's state in ReactJS?
20. How to add a CSS class whenever the component is updated in ReactJS?
21. How to Create a Toggle Switch in React as a Reusable Component?
22. How to create a custom progress bar component in React.js?
23. How to pass data from one component to other component in ReactJS?
24. How to convert functional component to class component in ReactJS?
25. How to put ReactJS component inside HTML string?
26. How to use CssBaseLine Component in ReactJS?
27. How to use ToggleButtonGroup Component in ReactJS?
28. How to use componentWillMount() in React Hooks?
29. How to use Link Component in ReactJS?
30. How to use BottomNavigation Component in ReactJS?
31. How to use Breadcrumbs Component in ReactJS?
32. How to use TextField Component in ReactJS?

33. How to use Portal Component in ReactJS?
34. How to use ScopedCssBaseline Component in ReactJS?
35. How to use Popper Component in ReactJS?
36. How to use Slide Component in ReactJS?
37. How to use Grow Component in ReactJS?
38. How to use Collapse Component in ReactJS?
39. How to use Fade Component in ReactJS?
40. How to use Zoom Component in ReactJS?
41. How to use Popover Component in ReactJS?
42. How to apply validation on Props in ReactJS?
43. What is prop drilling and how to avoid it?
44. How to create new elements with ReactJS mapping props?
45. How to pass multiple props in a single event handler in ReactJS?
46. How to send state/props to another component in React with onClick?
47. How to create a translucent text input in ReactJS?
48. How to set an object key inside a state object in React Hooks?

DSC 2F

Practical based on DSE 2 C (Data Visualization using Power BI)

1. Import data from different sources such as Excel, CSV, databases, and web services into Power BI.
2. Perform data cleaning tasks such as removing duplicates, handling missing values, and formatting data types.
3. Apply data transformation operations such as splitting columns, merging queries, and creating calculated columns.
4. Create relationships between different tables in the dataset.
5. Define hierarchies and drill-down paths to facilitate data exploration.
6. Implement calculated measures and calculated columns using DAX (Data Analysis Expressions).
7. Design interactive reports and dashboards using a variety of visualization types (e.g., bar charts, line charts, pie charts, maps, tables).
8. Customize the appearance of visualizations by adjusting formatting options such as colors, fonts, and labels.
9. Demonstrate at list 5 Maps
10. Demonstrate Table and Matrix with total and subtotal.
11. Demonstrate drill operations on any visuals
12. Incorporate slicers, filters, and bookmarks to enable users to interactively explore data.
13. Implement advanced calculations using DAX functions (e.g., CALCULATE, FILTER, RELATED, SUMMARIZE).

14. Perform time intelligence calculations such as year-over-year growth, moving averages, and cumulative totals.
15. Utilize AI-powered features such as Q&A (natural language querying) and Quick Insights to uncover hidden patterns and trends in data.
16. Create interactive tooltips and drill-through pages to provide additional context and detail to visualizations.
17. Demonstrate at list 5 Maps
18. Demonstrate any 10 visuals with formatting.
19. Demonstrate how to insert different objects and adding action to them.
20. Demonstrate different levels of filters.
21. Design sample report and dashboard.
22. Implement dynamic filtering and highlighting to focus attention on relevant data points.
23. Publish reports and dashboards to the Power BI Service for sharing with colleagues and stakeholders.
24. Configure row-level security to restrict access to sensitive data based on user roles and permissions.
25. Explore and utilize custom visuals available in the Power BI marketplace to enhance data visualizations.

Practical based on DSC 2G (ASP.NET Core)

1. Create a new ASP.NET Core MVC project using Visual Studio or the .NET CLI.
2. Explore the project structure and understand the role of important files such as Startup.cs, Program.cs, and the Views folder.
3. Define model classes representing entities in the application domain.
4. Generate scaffolded controllers and views using Entity Framework Core for CRUD operations on the model classes.
5. Customize the generated views and controllers to meet specific requirements.
6. Define custom routes using attribute routing and convention-based routing.
7. Implement route constraints to restrict the format of URL parameters.
8. Deminstrate how routing works and how URLs map to controller actions.
9. Create HTML forms for user input and data submission.
10. Implement form validation using data annotations and ModelState.IsValid.
11. Bind form data to model properties using model binding techniques.
12. Set up a database context and configure entity classes for use with Entity Framework Core.
13. Perform database migrations to create or update the database schema based on changes to the model
14. Demonstrate HTML helper tag.
15. Demonstrate Tag helper.
16. Data Annotation Attributes and Relationships in Entity Framework Core
17. Perform CRUD operations using Stored Procedures in Entity Framework Core
18. Demonstrate migration concept and add sample record using seeding technique

19. Demonstrate routing concept in ASP.Net core MVC
20. Demonstrate different ActionResult concept.
21. Implement CRUD operations (Create, Read, Update, Delete) using Entity Framework Core methods.
22. Create RESTful API endpoints for accessing application data using ASP.NET Core MVC controllers.
23. Implement HTTP methods (GET, POST, PUT, DELETE) to perform CRUD operations on resources.
24. Use attribute routing and model binding to define API routes and handle incoming requests.
25. Validate form input and handle form submissions before sending requests to the server.

Practical based on DSE 2 A (Linux and Shell Programming)

1. Navigate the file system using commands such as ls, cd, pwd, mkdir, and rmdir.
2. View file contents using commands such as cat, less, and head.
3. Manipulate files and directories using commands such as cp, mv, rm, and touch.
4. Use commands such as chmod and chown to change file permissions and ownership.
5. Understand the concept of file permissions (read, write, execute) for users, groups, and others.
6. View running processes and their resource usage using commands such as ps, top, and htop.
7. Manage processes using commands such as kill, killall, pkill, and pgrep.
8. Monitor system performance using commands such as df, du, and free.
9. Configure network settings using commands such as ifconfig, ip, and netstat.
10. Manage users and groups using commands such as useradd, usermod, groupadd, and passwd.
11. Shell Programming:
12. Write and execute a simple shell script using a text editor and the bash interpreter.
13. Use variables to store and manipulate data within a shell script.
14. Write comments to document the purpose and functionality of shell scripts.
15. Implement conditional statements in shell.
16. Use loop constructs (for, while) to iterate over lists of items or perform repetitive tasks.
17. Define and call functions within shell scripts to encapsulate reusable code blocks.
18. Pass arguments to functions and return values using the return statement.
19. Organize shell scripts into modular components for better code organization and maintainability.
20. Read input from files and process data using commands such as cat, grep, awk, and sed.
21. Use redirection operators (>, >>, <) to redirect input and output streams between files and commands.
22. Implement error handling mechanisms in shell scripts to gracefully handle unexpected conditions.
23. Use exit codes and error messages to communicate errors to users and other scripts.
24. Write Linux script for checking given number is prime, Armstrong and palindrome.
25. Write Linux script to display Fibonacci sequence up to n numbers.

Practical based on DSE 2 B (Mobile Application Development using Kotlin)

1. Set up a new Android project in Android Studio with Kotlin support.
2. Create a simple "Hello World" app that displays a greeting message on the screen. Run the app on an emulator or physical device to verify functionality.
3. Design and implement a user interface with various UI components such as TextViews, EditTexts, Buttons, and ImageViews.
4. Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button
5. Use layout files (XML) to define the structure and appearance of the app's screens. Apply styling and theming to enhance the visual appeal of the app.
6. Create a screen that has input boxes for User Name, Password, Address, Gender(radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button (use any layout)
7. Implement event listeners to handle user interactions with UI elements (e.g., button clicks, text input).
8. Validate user input and provide feedback to users for incorrect or incomplete input.
9. Use input controls such as EditTexts and Spinners to capture user data.
10. Implement navigation between different screens or fragments within the app.
11. Use activities, fragments, and intents to navigate between screens and pass data between them.
12. Create a multi-screen app with a navigation drawer, tab layout, or bottom navigation bar for seamless navigation.
13. Integrate data storage solutions such as SharedPreferences, SQLite databases, or Room Persistence Library to store and retrieve app data.
14. Implement CRUD (Create, Read, Update, Delete) operations for managing data records within the app.
15. Fetch and display remote data from RESTful APIs using libraries like Retrofit or Volley.
16. Handling Device Features:
17. Design an android application Send SMS using Intent
18. Design an android application for menu.
19. Access and utilize device features such as camera, location, sensors, and permissions.
20. Implement functionality to capture photos or videos using the device's camera.
21. Retrieve the user's current location and display it on a map using Google Maps API.
22. Prepare the app for deployment by optimizing performance, reducing APK size, and ensuring compliance with platform guidelines.
23. Create a user registration application that stores the user details in a database table.
24. Generate signed APKs or app bundles for release and upload them to Google Play Store or other app distribution platforms.
25. Manage app releases, updates, and user feedback through the app store console and monitoring tools.

DSE 2-C

Practical Assignments on DSE 2B: (Internet of Things)

1. Write a program to sense the available networks using Arduino.

2. Write a program to measure the distance using ultrasonic sensor and make LED blink using Arduino.

3. Write a program to detect the vibration of an object with sensor using Arduino.

4. Write a program to connect with the available Wi-Fi using Arduino.

5. Write a program to sense a finger when it is placed on the board Arduino.

6. Write a program to get temperature notification using Arduino.

7. Write a program for LDR to vary the light intensity of LED using Arduino.

8. Write a program to install MySQL database in Raspberry pi.

9. Write a program to work with basic MySQL queries by fetching data from database in Raspberry pi.

10. Write a program to switch light on when the input is 1 and switch the light off when the input is 0 using Raspberry pi.

UA

Punyashlok Ahilyadevi Holkar Solapur University, Solapur.
Faculty of Science and Technology.
Nature of Question Paper
BCA- III w.e.f. AY 2024-25

Time: 3 Hrs **Total Marks: 80**

Instructions

1) All Questions are compulsory 2) Figure to right indicate full marks.

Q.1 A) Choose correct alternative. (MCQ) **10 Marks**

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

Q. 1) B) Fill in the blank/Definition/One sentence answer/ One word answer / Give the name /Predict the product etc. **06 Marks**

1.
2.
3.
4.
5.
6.

Q. 2) Solve any Eight of the following. **16 Marks**

a)
b)
c)
d)
e)
f)
g)
h)
i)
j)

Q. 3) A) Attempt any Two of the following. **10 Marks**

a)
b)
c)

Q. 3) B) Short note/Solve **06 Marks**

Q. 4 A) Attempt any Two of the following. **08 Marks**

a)

b)

c)

Q. 4) B) Describe/Explain/Solve **08 Marks**

Q. 5) Attempt any Two of the following. **16 Marks**

a)

b)

c)

CA

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Faculty of Science and Technology. Nature of Question Paper for CBCS Pattern

BCA- III w.e.f. AY 2024-25

Time: **Internal Evaluation: 20 Marks**

Choose any two of the following Home Assignment / Unit Test / Tutorial /Seminar

---

Pattern of Examination

External Evaluation + Internal Evaluation

80 Marks + 20 Marks = 100 Marks

Passing Criteria –

1. Written Exam – 32 out of 80

2. Continuous Assessment (CA) – 08 out of 20

Format for Title Page and For Embossing

PROJECT REPORT

ON

TITLE OF THE SYSTEM

Submitted in partial fulfillment for the award of degree of

BACHELOR IN COMPUTER APPLICATION

IN

COMPUTER APPLICATION

To the

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Submitted By

NAME OF STUDENTS

Name of College

Department of Computer Science

Under The Guide

Name of Guide

20___ - 20__

The Guidelines regarding the documentation and scope of project are mentioned here below:

For Application Level Projects

Project Report should be submitted in following format for Commercial Application Projects viz. Payroll, Sales, Purchase, Inventory, Book Shop, Examination system etc.

3.

- Blank Pages at beginning
- Title Page
- Certificate Page
- Certificate from Guide and Head of Department
- Declaration
- Acknowledgement Page
- Index with printed Page with Numbers

1. CHAPTER 1: INTRODUCTION
    1.1. Existing System and Need for System
    1.2. Scope of Work
    1.3. Operating Environment – Hardware and Software
    1.4. Detail Description of Technology Used
2. CHAPTER 2: PROPOSED SYSTEM
    2.1. Proposed System
    2.2. Objectives of System
    2.3. User Requirements
3. CHAPTER 3 : ANALYSIS & DESIGN
    3.1. Data Flow Diagram (DFD)
    3.2. Entity Relationship Diagram (ERD)
    3.3. Data Dictionary
    3.4. Table Design
    3.5. Code Design
    3.6. Menu Tree
    3.7. Input Screens
    3.8. Report Formats
    3.9. Test Procedures and Implementation
4. CHAPTER 4: USER MANUAL
    4.1. User Manual
    4.2. Operations Manual / Menu Explanation
    4.3. Forms and Report Specifications

4.4. Drawbacks and Limitations
4.5. Proposed Enhancements

5. CHAPTER 5: Conclusions
6. Bibliography
7. ANNEXURES :
   7.1. ANNEXURE 1 : INPUT FORMS WITH DATA
   7.2. ANNEXURE 2 : OUTPUT REPORTS WITH DATA
   7.3. ANNEXURE 3 : SAMPLE CODE

- Blank Pages at the end.

For System-Level Projects

Project Report should be submitted in following format for Commercial Application Projects viz. Payroll, Sales, Purchase, Inventory, Book Shop, Examination system etc.

- 2 Blank Pages at beginning
- Title Page
- Certificate from Company
- Certificate from Guide and Head of the Department
- Declaration
- Acknowledgement
- Index with printed Page Numbers

1. CHAPTER 1: INTRODUCTION
   1.1. Existing System and Need for System
   1.2. Scope of Work
   1.3. Operating Environment – Hardware and Software
   1.4. Detail Description of Technology Used
2. CHAPTER 2: PROPOSED SYSTEM
   2.1. Proposed System
   2.2. Objectives of System
   2.3. User Requirements
3. CHAPTER 3: ANALYSIS & DESIGN
   3.1. Object Diagram
   3.2. Class Diagram
   3.3. Use Case Diagrams
   3.4. Module Hierarchy Diagram

3.5. Component Diagram
3.6. Module Specifications
3.7. Interface Diagram (in case of WAP and Embedded Systems)
3.8. User Interface Design (Screens etc.)
3.9. Table specifications (in case back end is a database)
3.10. Test Procedures and Implementation

4. CHAPTER 4: USER MANUAL
   4.1. User Manual
   4.2. Operations Manual / Menu Explanation
   4.3. Program Specifications / Flow Charts
   4.4. Drawbacks and Limitations
   4.5. Proposed Enhancements
5. CHAPTER 5: Conclusions
6. Bibliography
7. ANNEXURES :
   7.1. ANNEXURE 1 : USER INTERFACE SCREENS
   7.2. ANNEXURE 2 : OUTPUT REPORTS WITH DATA (if any)
   7.3. ANNEXURE 3 : SAMPLE PROGRAM CODE (which will prove sufficient