

ST. JOSEPH'S UNIVERSITY

BENGALURU-27



**DEPARTMENT OF COMPUTER SCIENCE AND COMPUTER
APPLICATIONS**

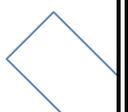
SYLLABUS FOR UNDERGRADUATE PROGRAMME BCA

For Batch 2021-2024



The objectives of the Programme are:

1. The primary objective of this program is to provide a foundation of computing principles and business practices for effectively using/managing information systems and enterprise software.
2. It helps students analyze the requirements for system development and exposes students to business software and information systems.
3. This course provides students with options to specialize in legacy application software, system software or mobile applications
4. To produce outstanding IT professionals who can apply the theoretical knowledge into practice in the real world and develop standalone live projects themselves
5. To provide opportunity for the study of modern methods of information processing and its applications.
6. To develop among students the coding skills and the problem- solving skills through programming
7. To prepare students who wish to go on to further studies in computer science and related subjects.
8. To acquaint students to work effectively with a range of current, standard, Office Productivity software applications



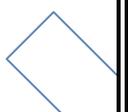
Program Outcomes: **BCA (3 Years) Degree**

1. **Discipline knowledge:** Acquiring knowledge on basics of Computer Science and ability to apply to design principles in the development of solutions for problems of varying complexity
2. **Problem Solving:** Improved reasoning with strong mathematical ability to Identify, formulate and analyze problems related to computer science and exhibiting a sound knowledge on data structures and algorithms.
3. **Design and Development of Solutions:** Ability to design and development of algorithmic solutions to real world problems and acquiring a minimum knowledge on statistics and optimization problems. Establishing excellent skills in applying various design strategies for solving complex problems.
4. **Programming a computer:** Exhibiting strong coding skills required to program a computer for various issues and problems of day-to-day applications with thorough knowledge on programming languages of various levels.
5. **Application Systems Knowledge:** Possessing a sound knowledge on computer application software and ability to design and develop apps for applicative problems.
6. **Modern Tool Usage:** Identify, select and use a modern scientific and IT tools and techniques for modeling, prediction, data analysis and solving problems in the area of Computer Science and making them mobile based application software.
7. **Communication and Soft Skills:** Students will acquire a reasonably good communication skills in both oral and written.
8. **Project Management:** Practicing of existing projects and becoming independent to launch own project by identifying the requirement in multidisciplinary fields.
9. **Ethics on Profession, Environment and Society:** Exhibiting professional ethics to maintain the integrity in a working environment and also have concern on societal impacts due to computer-based solutions for problems.
10. **Lifelong Learning:** Should become an independent learner. Thus, learn to learn unlearn and relearn.
11. **Motivation to take up Higher Studies:** Inspiration to continue education towards advanced studies in Computer Science.

Additional Program Outcomes: **BCA Degree (Hons)**

The Bachelor of Computer Application (BCA (Hons)) program enables students to attain following additional attributes besides the afore-mentioned attributes, by the time of graduation:

1. Apply standard Software Engineering practices and strategies in real -time software project development
2. Design and develop computer programs/computer -based systems in the areas related to AI, algorithms, networking, web design, cloud computing, IoT and data analytics.
3. Acquaint with the contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems
4. The ability to apply the knowledge and understanding to the analysis of a given information handling problem.
5. The ability to work independently on a substantial software project and as an effective team member.



DEPARTMENT OF COMPUTER SCIENCE AND COMPUTER APPLICATIONS(UG)
(2024-2027)

Semester 1	Code Number	Title	No. of Hours of Instructors	Number of Hours of teaching perweek	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CA 1124	Fundamentals of Computers	42	03	03	40	60	100
Theory	CA 1224	Programming in C	42	03	03	40	60	100
Theory	CA 1324	Mathematical Foundations	42	03	03	40	60	100
Theory	CA 1424	Unix Programming	42	03	03	40	60	100
Practical	CA 1P124	Information Technology and Unix Lab	52	02	1	25	25	50
Practical	CA 1P224	C Programming Lab	52	02	1	25	25	50

Total Number of credits:13+ 13(Total Credit of OE, Languages, CAE, Voc, AECC, SEC)

Semester 2	Code Number	Title	No. of Hoursof Instructors	Number of teaching hrs /week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CA2121	Data Structures using C	42	03	03	40	60	100
Theory	CA2221	Object Oriented Concepts using JAVA	42	03	03	40	60	100
Theory	CA2321	Discrete Mathematical Structures	42	03	03	40	60	100
Practical	CA2P1	Data Structures Lab	52	04	2	25	25	50
Practical	CA2P2	JAVA Lab	52	04	2	25	25	50

Total Number of credits:13+13(Total Credit of OE, Languages, CAE, VoC, AECC, SEC)

Semester 3	Code Number	Title	No. of Hours of Instructions	Number of teaching hrs /week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CA3122	Data Base Management Systems	42	03	03	40	60	100
Theory	CA3222	C# and DOT NET Framework	42	03	03	40	60	100
Theory	CA3322	Computer Communication and Networks	42	03	03	40	60	100
Practical	CA3P1	DBMS Lab	52	04	2	25	25	50
Practical	CA3P2	C# and DOT NET Framework Lab	52	04	2	25	25	50

Total Number of credits:13+13(Total Credit of OE, Languages, CAE, Voc, AECC, SEC)

Semester 4	Code Number	Title	No. of Hours of Instructions	Number of teaching Hrs /Week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CA4122	Python Programming	42	03	03	40	60	100
Theory	CA4222	Computer Multimedia and Animation	42	03	03	40	60	100
Theory	CA4322	Operating System Concepts	42	03	03	40	60	100
Practical	CA4P1	Python programming Lab	52	04	2	25	25	50
Practical	CA4P2	Multimedia and Animation Lab	52	04	2	25	25	50

Total Number of credits:13+13(Total Credit of OE, Languages, CAE, Voc, AECC, SEC)

Semester 5	Code Number	Title	No. of Hours of Instructions	Number of teaching Hrs /week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total Marks
Theory	CA5123	E Commerce	42	03	03	40	60	100
Theory	CA5223	Internet Technologies	42	03	03	40	60	100
Theory	CA5323	Cyberlaw and Cybersecurity	42	03	03	40	60	100
Theory	CADE5423	Cloud Computing	42	03	03	40	60	100
Theory	CAV01	Vocational 1 AWS - Cloud Practitioner Essentials	42	03	03	40	60	100
Practical	CA5P1	Internet Technologies Lab	52	04	2	25	25	50
Practical	CA5P2	Cybersecurity Lab	52	04	2	25	25	50

Semester 6	Code Number	Title	No. of Hours of Instructions	Number of teaching Hrs /week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CA6123	Software Engineering	42	03	03	40	60	100
Theory	CA6223	Artificial Intelligence and Applications	42	03	03	40	60	100
Theory	CA6323	Internet of Things	42	03	03	40	60	100
Theory	CADE6423	Mobile Application Development	42	03	03	40	60	100
Theory	DAVO2	Vocational 2 Power BI	42	03	03	40	60	100
Practical	CA6P1	Mobile App Lab	52	04	02	25	25	50
Practical	CA6P2	Major Project Lab	52	04	02	25	25	50

EXAMINATION AND ASSESMENTS

THEORY

1. IA Weightage 40 %
2. End Semester Examination Weightage 60%

PRACTICAL/PROJECTS

1. IA Weightage 50 %
2. End Semester Examination Weightage 50%

Title	Credits	CA Marks	SE Marks	Total Marks	Time Duration for ESE
Core Subjects	3 or 4	40	60	100	2 Hrs
Department Electives	3 or 4	40	60	100	2 Hrs
Open Electives	3	40	60	100	2 Hrs
Practicals	2	25	25	50	2 Hrs

QUESTION PAPER PATTERN CORE/DEPARTMENT ELECTIVE

The question papers of the theory examinations should follow the pattern specified below:

Section	Marks for each question	Number Of Questions		Total Marks
		Total	Should Answer	
A	2	5	5	10
B	4	7	5	20
C		4	3	30
(a+b)	10			

Total Marks : 60

OPEN ELECTIVE

60 MCQS – One mark each.

INTERNAL ASSESSMENT FORMAT

THEORY:

1. First Activity (should be multiple activities): 10 marks
2. Mid Semester Test: 20 marks (Test will be for 30 marks. It will be converted to 20)
3. Second Activity (should be multiple activities): 10 marks
4. Third Activity only for the first semester students who had scored less than 5 out of 10. Will be conducted after activity 2. Maximum marks shall be 5.

Total 40 marks

MID SEMESTER TEST QUESTION PAPER PATTERN

Section	Marks for each question	Number Of Questions		Total Marks
		Total	Should Answer	
A	2	6	5	10
B	5	5	4	20
				30

PRACTICALS:

Every practical class the student should be assessed.

1. Writing the observation book 10 marks
 2. Executing the programs and viva voce 5 marks
 3. Record writing 5 marks
- Total 25 marks**

PIA - 25 marks

PROJECT LAB

Presentation / demo must be carried out in all the lab sessions in the whole semester for internal assessment of the project. In each lab session a student is evaluated for 25 marks.

Presentation /Demo-15

Viva Voce-10

Semester: I

Course Code: CA 1124	Course Title: Fundamentals of Computers
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 100	Exam Duration: 3 Hrs.

Course Outcomes (COs):

- Introduction to computers, classification of computers, anatomy of computer, constituents and architecture, microcontrollers
- Operating systems, functions of operating systems, classification of operating systems, kernel, shell, basics of Unix, shell programming, booting
- Databases, why databases are used, users, SQL, data types in SQL, introduction of queries - select, alter, update, delete, truncate, using where, and or in not in
- Internet basics, features, applications, services, internet service providers, domain name system, browsing, email, searching
- Web Programming basics, introduction of HTML and CSS programming
- Introduction of computers, classification of computers, anatomy of computer, constituents and architecture, microcontrollers.
- Introduction to emerging technologies in the field of computer applications.

Content	Hours
Unit - 1	
Fundamentals of Computers: Introduction to Computers - Computer Definition, Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organization of a Digital Computer; Number Systems - different types, conversion from one number system to another; Computer Codes - BCD, Gray Code, ASCII and Unicode; Boolean Algebra - Boolean Operators with Truth Tables; Types of Software - System Software and Utility Software; Computer Languages - Machine Level, Assembly Level & High Level Languages, Translator Programs - Assembler, Interpreter and Compiler.	10
Unit-2	

Introduction to computers: Characteristics of computers, Classification of Digital Computer Systems: Microcomputers, Minicomputers, Mainframes, Super computers. Anatomy of Computer: Introduction, Functions & Components of a Computer, Central Processing Unit, Microprocessor, Storage units, Input and output Devices. How CPU and memory works. Program execution with illustrative examples. Introduction to microcontrollers.	10
Unit-3	
Operating System Fundamentals: Operating Systems: Introduction, Functions of an operating System, Classification of Operating Systems, System programs, Application programs, Utilities, The Unix Operating System, Basic Unix commands, Microkernel Based Operating System, Booting.	6
Unit-4	
Introduction to Database Management Systems: Database, DBMS, Why Database -File system vs DBMS, Database applications, Database users, Introduction to SQL, Data types, Classification of SQL-DDL with constraints, DML, DCL, TCL	6
Unit-5	
Internet and Web Basics: Introduction, Features of Internet, Internet application, Services of Internet, Logical and physical addresses, Internet Service Providers, Domain Name System. Introduction to web, web browsers, http/https, URL, Introduction to HTML5, Introduction to CSS. Introduction to Emerging technologies: Overview and features of Augmented Reality, Virtual Reality, Artificial Intelligence, Machine learning, Data Warehousing, Data Mining, Big data analytics, Internet of Things.	10

Text Books:

1. Pradeep K. Sinha and Priti Sinha: Computer Fundamentals (Sixth Edition), BPB Publication
2. David Riley and Kenny Hunt, Computational thinking for modern solver, Chapman & Hall/CRC,

Reference:

1. J. Glenn Brook shear," Computer Science: An Overview", Addison-Wesley, Twelfth Edition.
2. R.G. Dromey, "How to solve it by Computer", PHI

BLUE PRINT

Unit Nos.	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	10	9
Unit 2	10	10
Unit 3	6	14
Unit 4	6	20
Unit 5	10	23
TOTAL	42	76
Maximum marks for the paper (Excluding bonus questions) = 60		

Course Code: CA 1P124	Course Title: Information Technology Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 03

Course Outcomes (COs):

At the end of this practical classes the student will be able -

- To List various hardware components and its specifications.
- To recognize the hardware components of computers and describe its functions.
- To differentiate the role of hardware and software of the computer.
- To practice the setups of software and its installation process.
- To practice and compare all the tools of MS-Word, MS-Excel and MS-Powerpoint.

Part A: IT Lab

1. Operating System Installation – Windows OS, UNIX/LINUX, Dual Booting.
2. Installation and Uninstallation of Software – Office Tools, Utility Software (like Anti- Virus, System Maintenance tools); Application Software - Like Photo/Image Editors, Audio Recorders/Editors, Video Editors ...); Freeware, Shareware, Payware and Trial ware; Internet Browsers, Programming IDEs,
3. System Configuration – BIOS Settings, Registry Editor, MS Config, Task Manager, System Maintenance, Third-party System Maintenance Tools (Similar to Cleaner and Jv16 Power Tools ...)
4. Activities using Word Processor Software
5. Activities using Spreadsheets Software
6. Activities using Presentation Software
7. Activities involving Multimedia Editing (Images, Video, Audio ...)
8. Tasks involving Internet Browsing

PART B UniX Lab

1. Study of Unix Internal And external commands.
2. Study of File related commands.
3. Study of Directory related commands.
4. Study of Vi editor.
5. Study of connecting commands.
6. Study of redirection commands.
7. Study of Regular Expressions.
8. Study of Shell Programming.
9. Study of Filter Commands
10. Study of process commands

Reference:

1. Computational Thinking for the Modern Problem Solver, By Riley DD, Hunt K.A CRC press, 2014
2. Ferragina P, Luccio F. Computational Thinking: First Algorithms, Then Code. Springer

Web References:

<http://www.flowgorithm.org/documentation/>

Course Code: CA1221	Course Title: Programming in C
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 100	Exam Duration: 3 hrs

Course Outcomes (COs):

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

- Confidently operate Desktop Computers to carry out computational tasks
- Understand working of Hardware and Software and the importance of operating systems
- Understand programming languages, number systems, peripheral devices, networking, multimedia and internet concepts
- Read, understand and trace the execution of programs written in C language
- Write the C code for a given problem
- Perform input and output operations using programs in C
- Write programs that perform operations on arrays

Content	Hours
Unit - 1	
<p>Introduction to C Programming: Overview of C; History and Features of C; Planning a Computer Program-Algorithm, Flowchart and Pseudo code with Examples. Structure of a C Program with Examples; Creating and Executing a C Program; Compilation process in C.</p> <p>C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants.</p> <p>Input and output with C: Formatted I/O functions - <i>printf</i> and <i>scanf</i>, control stings and escape sequences, output specifications with <i>printf</i> functions; Unformatted I/O functions to read and display single character and a string - <i>getchar</i>, <i>putchar</i>, <i>gets</i> and <i>puts</i> functions.</p>	10
Unit - 2	
<p>C Operators & Expressions: Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associatively; Evaluation of arithmetic expressions; Type conversion.</p> <p>Control Structures: Decision making Statements - <i>Simple if</i>, <i>if_else</i>, <i>nested if_else</i>, <i>else_if ladder</i>, <i>Switch Case</i>, <i>goto</i>, <i>break</i> & <i>continue</i> statements; Looping</p>	12

Statements - Entry controlled and exit controlled statements, <i>while, do-while, for</i> loops, Nested loops.	
Unit - 3	
<p>Derived data types in C: Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays -Declaration, Initialization and Memory representation.</p> <p>Strings: Declaring & Initializing string variables; String handling functions - <i>strlen, strcmp, strcpy and strcat</i>; Character handling functions - <i>toascii, toupper, tolower, isalpha, isnumeric</i> etc.</p>	8
Unit - 4	
<p>User Defined Functions: Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type.</p> <p>User defined data types: Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.</p>	6
Unit - 5	
<p>Pointers in C: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Advantages and disadvantages of using pointers;</p> <p>File Operations : File modes, File functions, and File operations, Text and Binary files, Command Line arguments.</p>	6

Text Books:

1. C: The Complete Reference, By Herbert Schildt.
2. Kernighan & Ritchie: The C Programming Language (PHI)

Reference Books:

1. P. K. Sinha & Priti Sinha: Computer Fundamentals (BPB)
2. E. Balaguruswamy: Programming in ANSI C (TMH)
3. Kamthane: Programming with ANSI and TURBO C (Pearson Education)
4. V. Rajaraman: Programming in C (PHI - EEE)
5. S. Byron Gottfried: Programming with C (TMH)
6. Yashwant Kanitkar: Let us C
7. P.B. Kottur: Programming in C (Sapna Book House)

BLUE PRINT

Unit Nos.	No. of Hours	Total marks for which questions are to be asked(including bonus questions)
Unit 1	10	9
Unit 2	12	10
Unit 3	8	14
Unit 4	6	20
Unit 5	6	23
TOTAL	42	76
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code: CA1P2	Course Title: C Programming Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours:52	Formative Assessment Marks:25
Exam Marks: 25	Exam Duration: 03

Course Outcomes (COs):

At the end of this practical classes the student will be able -

- To relate the types of programming paradigm and syntax.
- To recognize the need of programs that can solve the real world problem.
- To differentiate the types of program statements with illustrations.
- To practice the code optimization effectively that can solve mathematical problems.
- To interpret the role of event driven programs, and modularity programs.

Programming Lab

Part A:

1. Program to read radius of a circle and to find area and circumference
2. Program to read three numbers and find the biggest of three
3. Program to demonstrate library functions in math.h
4. Program to check for prime
5. Program to generate n primes
6. Program to read a number, find the sum of the digits, reverse the number and check it for palindrome
7. Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers
8. Program to read percentage of marks and to display appropriate message (Demonstration of else-if ladder)
9. Program to find the roots of quadratic equation (demonstration of switchCase statement)
10. Program to read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
11. Program to remove Duplicate Element in a single dimensional Array
12. Program to perform addition and subtraction of Matrices

Part B:

1. Program to find the length of a string without using built in function
2. Program to demonstrate string functions.
3. Program to demonstrate pointers in C
4. Program to check a number for prime by defining isprime() function

5. Program to read, display and to find the trace of a square matrix
6. Program to read, display and add two $m \times n$ matrices using functions
7. Program to read, display and multiply two $m \times n$ matrices using functions
8. Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
9. Program to Reverse a String using Pointer
10. Program to Swap Two Numbers using Pointers
11. Program to demonstrate student structure to read & display records of n students.
12. Program to demonstrate the difference between structure & union.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course

Course Code: CA1321	Course Title: Mathematical Foundations
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 100	Exam Duration: 3

Course Outcomes (COs):

- Study and solve problems related to connectives, predicates and quantifiers under different situations.
- Develop basic knowledge of matrices and to solve equations using Cramer's rule.
- Know the concept of Eigen values.
- To develop the knowledge about derivatives and know various applications of differentiation.
- Understand the basic concepts of Mathematical reasoning, set and functions

Content	Hours
Unit - 1	
Basic concepts of logic: Mathematical logic introduction-statements Connectives-negation, conjunction, disjunction- statement formulas and truth tables- conditional and bi Conditional statements- tautology contradiction- equivalence of formulas-duality law-Predicates and Quantifiers, Arguments.	6
Unit - 2	
Operations on sets: power set- Venn diagram Cartesian product-relations - functions- types of functions - composition of functions.	10
Unit - 3	
Matrix algebra: Introduction-Types of matrices-matrix operations- transpose of a matrix -determinant of matrix - inverse of a matrix- Cramer's rule	10
Unit - 4	
Matrix: finding rank of a matrix - normal form-echelon form cayley Hamilton theorem-Eigen values	6
Unit - 5	
Differential calculus: Functions and limits - Simple Differentiation of Algebraic Functions – Evaluation of First and Second Order Derivatives – Maxima and Minima	10

Text Books:

P. R. Vittal-Business Mathematics and Statistics, Margham Publications, Chennai,

Reference Books:

1. B. S. Vatsa-Discrete Mathematics –New Age International Limited Publishers, New Delhi
2. Discrete Mathematics and Its Applications, Kenneth H. Rosen: Seventh Edition, 2012.

BLUE PRINT

Unit Nos.	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	10	10
Unit 2	6	9
Unit 3	10	20
Unit 4	6	14
Unit 5	10	20
TOTAL	42	76
Maximum marks for the paper (Excluding bonus questions) = 60		

TITLE : UNIX PROGRAMMING
CODE : CA 1424
Hrs / Week : 3Hrs
CREDITS : 3

Course Objectives:

This course will enable students to:

- Understand the UNIX Architecture, File systems and use of basic Commands.
- Use of editors commands.
- Understand Shell Programming and to write shell scripts.
- Understand and analyze UNIX System calls, Process Creation, Control & Relationship.

Course outcomes:

After studying this course, students will be able to:

- Explain multi user OS UNIX and its basic features
- Interpret UNIX Commands, Shell basics, and shell environments
- Design and develop shell programming, communication, System calls and terminology.
- Design and develop UNIX File I/O and UNIX Processes.

UNIT I

(9)

INTRODUCTION

Introduction- Unix Architecture. Features of Unix. The login prompt. General features of Unix commands. Understanding of some basic commands such as echo, printf, ls, who, date, passwd, cal. Meaning of Internal and external commands. The type command, the man command, more command, Knowing the user terminal. The root login. Becoming the super user: su command. Commands to add, modify and delete users.

UNIT II

(9)

UNIX FILES: Naming files. Basic file types/categories. Parent child relationship. The home directory and the HOME variable. Directory commands - pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations to represent present and parent directories and their usage in relative path names. File related commands - cat, mv, rm, cp, wc and od commands. File attributes and permissions. The ls command with options. Changing file permissions.

UNIT III

(9)

THE VI EDITOR: Basics- Different ways of invoking and quitting vi. Different modes of vi. Input mode commands. Command mode commands. The ex mode commands. Illustrative examples- Navigation commands. Simple examples using these commands. The shells interpretive cycle. Wild cards and file name generation. Three standard files and redirection. Connecting commands: Basic and Extended regular expressions. The grep.

UNIT IV

(9)

SHELL PROGRAMMING: Shell programming. Ordinary and environment variables. Read and read only commands. Command line arguments. exit and exit status of a command. Logical operators for conditional execution. The if, while, for and case control statements. Simple shell

program examples.

UNIT V

(9)

FILE INODES AND PROCESS: File inodes and the inode structure. File links – hard and soft links. Filters- Head and tail commands. Cut and paste commands. The sort command and its usage with different options. Meaning of a process. Mechanism of process creation. Parent and child process. The ps command with its options. Background processes. The bg and fg command.

Self Study: Comparison of different operating systems

Text Books:

- Sumitabha Das., Unix Concepts and Applications., 4th Edition., Tata McGraw Hill
- Behrouz A. Forouzan, Richard F. Gilberg : UNIX and Shell Programming- Cengage Learning – India Edition. 2009.

Reference Books:

- M.G. Venkatesh Murthy: UNIX & Shell Programming, Pearson Education.
- Richard Blum , Christine Bresnahan : Linux Command Line and Shell Scripting Bible, 2ndEdition , Wiley,2014.

BLUE PRINT FOR QUESTION PAPER

UNITS	2 MARKS (5 Questions)	4 MARKS (7 Questions)	10 MARKS (3 Questions)	TOTAL
UNIT 1	2	1	0.5	11
UNIT 2	2	2	1	20
UNIT 3	2	1	1	20
UNIT 4	2	1	0.5	11
UNIT 5	2	2	1	20
TOTAL MARKS	10	28	40	82

Semester: II

Course Code: CA2121	Course Title: Data Structures using C
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 100	Exam Duration: 3 Hrs.

Course Outcomes (COs):

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

- Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms
- Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs
- Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs
- Demonstrate different methods for traversing trees
- Compare alternative implementations of data structures with respect to performance
- Describe the concept of recursion, give examples of its use
- Discuss the computational efficiency of the principal algorithms for sorting, searching, and hashing

Content	Hours
Unit - 1	
<p>Introduction to data structures: Definition; Types of data structures - Primitive & Non-primitive, Linear and Non-linear; Operations on data structures.</p> <p>Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - <i>malloc</i>, <i>calloc</i>, <i>realloc</i> and <i>free</i>.</p> <p>Algorithm Specification, Performance Analysis, Performance Measurement-Asymptotic notations.</p>	6
Unit - 2	
<p>Arrays: Basic Concepts – Definition, Declaration, Initialization, Operations on arrays; Types of arrays; Arrays as abstract data types (ADT); Representation of Linear Arrays in memory; Traversing linear arrays; Inserting and deleting elements; Sorting – Selection sort, Bubble sort, Quick sort, Insertion sort; Searching - Sequential Search, Binary search; Iterative and Recursive searching; Multidimensional arrays;</p> <p>Representation of multidimensional arrays; Sparse matrices.</p>	10

Unit - 3	
Stacks: Basic Concepts – Definition and Representation of stacks; Operations on stacks; Applications of stacks; Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression using stack; Application of stack in function calls.	8
Unit - 4	
Queues: Basic Concepts – Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues; Linked list: Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly linked list, Header linked list, Circular linked list; Representation of Linked list in Memory; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation; Garbage collection.	10
Unit - 5	
Trees: Definition; Tree terminologies –node, root node, parent node, ancestors of a node, siblings, terminal & non-terminal nodes, degree of a node, level, edge, path, depth; Binary tree: Type of binary trees - strict binary tree, complete binary tree, binary search tree and heap tree; Array and pointer representation of binary tree. Traversal of binary tree; preorder, inorder and postorder traversal;	8

Text Books

1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures

References

1. Tanenbaum: Data structures using C (Pearson Education)
2. Kamathane: Introduction to Data structures (Pearson Education)
3. Y. Kanitkar: Data Structures Using C (BPB)
4. Kottur: Data Structure Using C
5. Padma Reddy: Data Structure Using C
6. Sudipa Mukherjee: Data Structures using C – 1000 Problems and Solutions (McGraw Hill Education, 2007)

BLUE PRINT

Unit Nos.	No. of hours	Total marks for which questions are to be asked (including bonus questions)
Unit 1	6	9
Unit 2	10	10
Unit 3	8	14
Unit 4	10	23
Unit 5	8	20
TOTAL	42	76
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code: CA2P1	Course Title: Data Structures Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks:25	Exam Duration: 03 Hours

Course Outcomes (COs):

At the end of this practical classes the will have an ability -

- To analyze algorithms with correctness, time and space complexity.
- To summarize and compare various searching and sorting techniques
- To describe the implementation of stack, queue and linked list operation.
- To interpret and organize the data using tree and graphs concepts.

Programming Lab

Part A:

1. Program to find GCD using recursive function
2. Program to display Pascal Triangle using binomial function
3. Program to generate n Fibonacci numbers using recursive function.
4. Program to implement Towers of Hanoi.
5. Program to implement dynamic array, find smallest and largest element of the array.
6. Program to create two files to store even and odd numbers.
7. Program to create a file to store student records.
8. Program to read the names of cities and arrange them alphabetically.
9. Program to sort the given list using selection sort technique.
10. Program to sort the given list using bubble sort technique.

Part B:

1. Program to sort the given list using insertion sort technique.
2. Program to sort the given list using quick sort technique.
3. Program to sort the given list using merge sort technique.
4. Program to search an element using linear search technique.
5. Program to search an element using recursive binary search technique.
6. Program to implement Stack.
7. Program to convert an infix expression to postfix.
8. Program to implement simple queue.
9. Program to implement linear linked list.
10. Program to display traversal of a tree.

Course Code: CA2221	Course Title: Object Oriented Programming with JAVA
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 100	Exam Duration: 03 hrs

Course Outcomes (COs):

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

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

Course Content

Content	Hours
Unit - 1	
Introduction to Java: Basics of Java programming, Data types, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, Math class, Arrays in java.	6
Unit - 2	
Objects and Classes: Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character, String Buffer, File, this reference.	6
Unit - 3	
Inheritance and Polymorphism: Inheritance in java, Super and sub class, Overriding, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java, Package in java, UTIL package.	8
Unit - 4	
Event and GUI programming: Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows, Menus, Dialog Box, Applet and its life cycle, Introduction to swing, Exceptional handling mechanism.	10
Unit - 5	

I/O programming: Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files.	6
Unit - 6	
Multithreading and Exception handling in java: Thread life cycle and methods, Runnable interface, Thread synchronization, Exception handling with try catch-finally, Collections in java, Introduction to JavaBeans and Network Programming.	6

Text Books

1. "Introduction to Java Programming" by Daniel Liang
2. Programming with Java, By E Balagurusamy – A Primer, Fourth Edition, Tata McGraw Hill Education Private Limited.
3. Core Java Volume I – Fundamentals, By Cay S. Horstmann, Prentice Hall
4. Object Oriented Programming with Java : Somashekara, M.T., Guru, D.S., Manjunatha, K.S

Reference Books

1. Java 2 - The Complete Reference – McGraw Hill publication.
2. Java - The Complete Reference, 7th Edition, By Herbert Schildt– McGraw Hill publication.

BLUE PRINT

Unit Nos.	No. of Hours	Total marks for which questions are to be asked (including bonus questions)
Unit 1	6	9
Unit 2	6	10
Unit 3	8	13
Unit 4	10	20
Unit 5	6	10
Unit 6	6	14
TOTAL	42	76
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code: CA2P2	Course Title: JAVA Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours:52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 03 Hours

Course Outcomes (COs):

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

- Implement Object Oriented programming concept using basic syntaxes of control Structures
- Identify classes, objects, members of a class and the relationships among them needed for a finding the solution to specific problem
- Demonstrates how to achieve reusability using inheritance
- Demonstrate understanding and use of interfaces, packages, different exception handling mechanisms and concept of multithreading for robust faster and efficient application development.
- Identify and describe common user interface components to design GUI in Java using Applet & AWT along with response to events

Practice Lab

1. Program to print the following triangle of numbers

```

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

```
2. Program to simple java application, to print the message, "Welcome to java"
3. Program to display the month of a year. Months of the year should be held in an array.
4. Program to find the area of rectangle.
5. program to demonstrate a division by zero exception
6. Program to create a user defined exception say Pay out of Bounds.

Programming Lab

PART A: Java Fundamentals OOPs in Java

1. Program to assign two integer values to X and Y. Using the 'if' statement the output of the program should display a message whether X is greater than Y.
2. Program to list the factorial of the numbers 1 to 10. To calculate the factorial value, use while loop. (Hint Fact of 4 = 4*3*2*1)
3. Program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use function overloading.

4. Program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the super class. MulDiv should have methods to multiply and divide. A main function should access the methods and perform the mathematical operations.
5. Program with class variable that is available for all instances of a class. Use static variable declaration. Observe the changes that occur in the object's member variable values.
6. Program
 - a. To find the area and circumference of the circle by accepting the radius from the user.
 - b. To accept a number and find whether the number is Prime or not
7. Program to create a student class with following attributes; Enrollment No: Name, Mark of sub1, Mark of sub2, mark of sub3, TotalMarks. The three marks must be calculated only when the student passes in all three subjects. The pass mark for each subject is 50. If a candidate fails in any one of the subjects, his total mark must be declared as zero. Using this condition write a constructor for this class. Write separate functions for accepting and displaying student details. In the main method create an array of three student objects and display the details.
8. In a college first year class are having the following attributes Name of the class (BCA, BCom, BSc), Name of the staff No of the students in the class, Array of students in the class
9. Define a class called first year with above attributes and define a suitable constructor. Also write a method called best Student () which process a first-year object and return the student with the highest total mark. In the main method define a first-year object and find the best student of this class
10. Program to define a class called employee with the name and date of appointment. Create ten employee objects as an array and sort them as per their date of appointment. ie, print them as per their seniority.
11. Create a package 'student. Fulltime. BCA 'in your current working directory
 - a. Create a default class student in the above package with the following attributes: Name, age, sex.
 - b. Have methods for storing as well as displaying

PART B: Exception Handling & GUI Programming

1. Program to catch Negative Array Size Exception. This exception is caused when the array is initialized to negative values.
2. Program to handle Null Pointer Exception and use the "finally" method to display a message to the user.
3. Program which create and displays a message on the window
4. Program to draw several shapes in the created window

5. Program to create an applet and draw grid lines
6. Program which creates a frame with two buttons father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother similar details of mother also appear.
7. Create a frame which displays your personal details with respect to a button click
8. Create a simple applet which reveals the personal information of yours.
9. Program to move different shapes according to the arrow key pressed.
10. Program to create a window when we press M or m the window displays Good Morning, A or a the window displays Good After Noon E or e the window displays Good Evening, N or n the window displays Good Night
11. Demonstrate the various mouse handling events using suitable example.
12. Program to create menu bar and pull-down menus.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course

Course Code: CA2321	Course Title: Discrete Mathematical Structures
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 100	Exam Duration: 03 Hrs

Course Outcomes (COs):

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

- To understand the basic concepts of Mathematical reasoning, set and functions.
- To understand various counting techniques and principle of inclusion and exclusions.
- Understand the concepts of various types of relations, partial ordering and equivalence relations.
- Apply the concepts of generating functions to solve the recurrence relations.
- Familiarize the fundamental concepts of graph theory and shortest path algorithm

Content	Hours
Unit - 1	
The Foundations: Logic and proofs: Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy.	10
Unit - 2	
Counting: Basics of counting, Pigeonhole principle, Permutation and combination, Binomial Coefficient and Combination, Generating functions-Generating Permutation and Combination. Advanced Counting Techniques: Applications of Recurrence Relations, Solving Linear Recurrence, Relations, Divide and Conquer Algorithms and Recurrence Relations, Inclusion-Exclusion, Applications of Inclusion-exclusion.	12
Unit - 3	
Induction and Recursion: Mathematical Induction, Strong Induction and Well-	12

Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Corrections.	
Relation: Properties of relation, Composition of relation, Closer operation on relation, Equivalence relation and partition. Operation on relation, Representing relation.	
Unit - 4	
Graphs: Graphs and Graph models, Special Types of Graphs, Sub graphs, Walk, Path and circuit, Fundamental circuit, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs.	8

Text Book:

1. Discrete Mathematics and Its Applications, Kenneth H. Rosen: Seventh Edition,2012.

References:

1. Discrete Mathematical Structure, Bernard Kolman, Robert C, Busby, Sharon Ross,2003.
2. Graph Theory with Applications to Engg and Comp. Sci: Narsingh Deo-PHI 1986.
3. Discrete and Combinatorial Mathematics Ralph P. Grimaldi, B. V. Ramatta,Pearson, Education, 5 Edition.
4. Discrete Mathematical Structures, Trembley and Manobar.

BLUE PRINT

Unit Nos.	No. of Hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	10	10
Unit 2	12	10
Unit 3	12	33
Unit 4	8	23
TOTAL	42	76
Maximum marks for the paper (Excluding bonus questions) = 60		

Semester: III

Course Code:CA3122	Course Title: DATABASE MANAGEMENT SYSTEMS
Course Credits:03	Hours/Week:03
Total Contact Hours:42	Formative Assessment Marks:40
Exam Marks:60	Exam Duration: 2 Hrs.

Course Outcomes (COs):

- Understanding the core terms, concepts, and tools of relational database management systems.
- Understanding database design and logic development for database programming.
- Demonstrate the concepts of Relational database model, ER model and Distributed databases.
- Implement database operations and transactions using SQL.
- Apply the concepts of Transaction processing, Concurrency control, Database Recovery and Back-up in applications.
- Databases, why databases are used, users, SQL, data types in SQL, introduction of queries - select, alter, update, delete, truncate, using where, and or in not in

Content	Hours
Unit -1	
Database and Database Users: Introduction, Characteristics of the Database Approach, Database Users, Advantages of Using DBMS Approach, When Not to Use a DBMS. Database System Concepts and Architecture: Data Models, Schemas, and Instances, Three-schema Architecture and Data Independence, Database Languages and Interfaces, The Database System Environment, Centralized and Client-Server Architectures, Classification of Database Management Systems.	8
Unit-2	

Data Modeling Using Entity-Relationship Model: Using High-Level Conceptual Data Models for Database Design, An Example Database Application, Entity Types, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, Refining the ER Design, Naming Conventions and Design. File organization and storage, secondary storage devices, type of single level ordered index, multi-level indexes.	10
Unit-3	
Relational Algebra: Relational Model Concepts, Relational Model Constraints, Unary Relational Operations: SELECT, PROJECT, RENAME. Relational Algebra Operations from SET Theory, Binary Relational Operations: JOIN and DIVISION, Examples of Queries in Relational Algebra. Relational Database Design: Anomalies in a database, functional dependency, normal forms – 1NF, 2NF, 3NF, BCNF, Introduction to higher order normal forms.	8
Unit-4	
Structured Query Language: SQL Data Definition and Data Types, Basic Queries in SQL- CREATE statement, Specifying Constraints in SQL, Basic Retrieval Queries in SQL – SELECT statement, INSERT, DELETE, and UPDATE Statements in SQL, GRANT and REVOKE statements, Complex SQL Retrieval Queries - Comparisons Involving NULL and Three-Valued Logic, Nested Queries, The GROUP BY and HAVING Clauses. Views (Virtual Tables) in SQL - CREATE VIEW, View Implementation, View Update, and Inline Views. Schema Change Statements in SQL – DROP and ALTER command.	8
Unit-5	
Introduction to Transactions and Database Recovery: Introduction to transaction processing, transaction and system concepts, desirable properties of transactions. Concurrency control techniques: types of lock, two-phase locking techniques. Recovery techniques: recovery concepts, database backup and recovery from catastrophic failures.	8

Text Books:

1. Elmasri and Navathe: Fundamentals of Database Systems, 7th Edition, Addison - Wesley, 2016.
2. Silberschatz, Korth and Sudharshan Data base System Concepts, 7th Edition, TataMcGraw Hill, 2019.

Reference:

1. C.J. Date, A. Kannan, S. Swamynatham: An Introduction to Database Systems, 8th Edition, Pearson education, 2009
2. Database Management Systems :Raghu Ramakrishnan and Johannes Gehrke: , 3rd Edition, McGrawHill, 2003

BLUE PRINT

Unit No.	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	8	9
Unit 2	10	10
Unit 3	8	19
Unit 4	8	19
Unit 5	8	19
TOTAL	42	76
Maximum marks for the paper (Excluding bonus questions) = 60		

Course Code: CA3P1	Course Title: DATABASE LAB
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks:25	Exam Duration: 03 Hours

Course Outcomes (COs):

This course will enable students to

- To understand basic database concepts, applications, data models, schemas and instances.
- Describe the basics of SQL and construct queries using SQL. Emphasize the importance of normalization in databases.

SQL Programming

- Design, develop, and implement the specified queries for the following problems using Oracle, MySQL, MS SQL Server, or any other DBMS under LINUX/Windows environment.
 - Create Schema and insert at least 5 records for each table. Add appropriate database constraints.
1. Perform the following:
 - i) Creating a Database,
 - ii) Viewing all databases,
 - iii) Viewing all Tables in a Database,
 - iv) Creating Tables (With and Without Constraints),
 - v) Delete Table
 - vi) Rename Table.
 2. Write SQL Queries involving:
 1. Date Functions,
 2. String Functions and
 3. Math Functions.
 3. Create a table STATION to store information about weather observation stations with fields: ID(Primary Key), CITY, STATE, LAT, LONG and populate the table STATION with a few rows:
 - i) Write a SQL query to look at table STATION in undefined order
 - ii) Write a SQL query to select Northern stations (latitude > 39.7)
 - iii) Write a SQL query to select only ID, CITY, and STATE columns
 - iv) Write a SQL query to select only ID, CITY, and STATE columns where Longitude >45.
 4. Create a table STATION to store information about weather observation stations with fields: ID (Primary Key), CITY, STATE, LAT, LONG. Duplicate ID fields are not allowed. Populate the table STATION with a few rows.

- Create another table called STATS to store normalized temperature and precipitation data:
- ID field must match some STATION table ID (so that name and location will be known).
 - Allowable ranges will be enforced for other values.
 - No duplicate ID and MONTH combinations.
 - Temperature is in degrees Fahrenheit.
 - Rainfall is in inches.
- Populate the table STATS with some statistics for January and July.
 - Write a SQL query to look at table STATS in undefined order.
 - Write a SQL query to look at table STATS, picking up location information by joining with table STATION on the ID column.
 - Write a SQL query to look at the table STATS, ordered by month and greatest rainfall, with columns rearranged.
5. Create a table STATION to store information about weather observation stations with fields: ID(Primary Key), CITY, STATE, LAT, LONG. No duplicate ID fields allowed. Populate the table STATION with a few rows. Create another table called STATS to store normalized temperature and precipitation data.
- ID field must match some STATION table ID (so name and location will be known).
 - Allowable ranges will be enforced for other values.
 - No duplicate ID and MONTH combinations.
 - Temperature is in degrees Fahrenheit.
 - Rainfall is in inches.
- Write a SQL query to look at temperatures for July from table STATS, lowest temperatures first, picking up city name and latitude by joining with table STATION on the ID column
 - Write a SQL query to show MAX and MIN temperatures as well as average rainfall for each station
 - Write a SQL query (with sub query) to show stations with year-round average temperature above 50 degrees.
 - Rows are selected from the STATION table based on related values in the STATS table.
6. Create table called STATS to store normalized temperature and precipitation data.
- Allowable ranges will be enforced for other values.
 - No duplicate ID and MONTH combinations.
 - Temperature is in degrees Fahrenheit.
 - Rainfall is in inches.
- Create a view (derived table or persistent query) to convert Fahrenheit to Celsius and inches to centimeters
 - Add new column rainfall_centimeter to the table STATS.
 - Insert values into rainfall_centimeter from the view
 - Delete Column rainfall from table STATS
7. Create table called STATS to store normalized temperature and precipitation data:
- Allowable ranges will be enforced for other values.
 - No duplicate ID and MONTH combinations.
 - Temperature is in degrees Fahrenheit.
 - Rainfall is in inches.
- Write a SQL query to look at table STATS in a metric light (through the new view).

- ii) Write a SQL metric query restricted to January below-freezing (0 Celsius) data, sorted on rainfall.
8. Create table called STATS to store normalized temperature and precipitation data.
- Update all rows of table STATS to compensate for faulty rain gauges known to read 0.01 inches low
 - Update one row, ID 44's July temperature reading, to correct a data entry error
 - Make the above changes permanent
 - Undo that update
9. Create a table STATION to store information about weather observation stations with fields: ID(Primary Key), CITY, STATE, LAT, LONG. No duplicate ID fields allowed. Populate the table STATION with a few rows.
- Create another table called STATS to store normalized temperature and precipitation data.
- ID field must match some STATION table ID (so name and location will be known).
 - Allowable ranges will be enforced for other values.
 - No duplicate ID and MONTH combinations.
 - Temperature is in degrees Fahrenheit.
 - Rainfall is in inches.
- Delete data from STATION table where longitude is >90
 - Delete July data from STATS table where longitude is >90
 - Increase the size for the column CITY with the following information:-

COLUMNNAME	DATATYPE(SIZE)
-----	-----
CITY	VARCHAR (25)
 - Modify the column name of LONG to LONGITUDE present in the STATION table and verify the result.
10. Create Table INSTRUCTOR with the following fields: InstuctID, InstructName, Department, Salary.
- Find instructors whose salary is more than the salary of any employee from department 'Physics'.
 - Find the instructor name and department name of all instructors working in a department with any instructor whose name contains the letter "S".
 - Find the name and department name, of instructors whose salary is more than all their colleagues salaries in the same department.
 - Create table INSTRUCTOR with the following fields: InstuctID, InstructName, Department, Salary.
- Create another table called STUDENT with the following fields: StudentID, StudentName, Department, InstuctID
- Find the names of all instructors whose salary is greater than at least one instructor in the Finance department.
 - Find all instructors whose salary is less than the salary of all instructors in the Computer Science department and whose department name is not Computer Science.
 - Find the student name and department name of all student who study in a department with any student whose name contains the letter "S".

11. Consider the following schema for a Library Database:

BOOK (Book_id, Title, Publisher_Name, Pub_Year)

BOOK_AUTHORS (Book_id, Author_Name)

PUBLISHER (Publisher_Name , Address, Phone)

BOOK_COPIES (Book_id, Branch_id, No-of_Copies)

BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out, Due_Date)

LIBRARY_BRANCH (Branch_id, Branch_Name, Address).

- i) Draw the E_R Diagram for the Library database.
 - a. Write SQL query to Retrieve details of all books in the library: id, title, name of publisher, authors, number of copies in each branch.
 - b. Write an SQL query to compute the total number of books based on publisher name order by year.

12. Consider the schema for Movie Database:

ACTOR(Act_id, Act_Name, Act_Gender)

DIRECTOR(Dir_id, Dir_Name, Dir_Phone)

MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)

MOVIE_CAST(Act_id, Mov_id, Role)

RATING(Mov_id, Rev_Stars)

- i. Draw the E_R Diagram for the Movie database.
- ii. Write an SQL query to find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title. (Use Inner Join)
- iii. Write an SQL query to retrieve all the actors and any movies they have acted in (Use left Join)
- iv. Write an SQL query to retrieve all the directors and any movies they have directed.(Use right outer join)
- v. Write an SQL query to retrieve all the directors name and all the movies (use Full join)

Course Code: CA3222	Course Title: C# AND .NET FRAMEWORK
Course Credits: 03	Hours/Week:03
Total Contact Hours:42	Formative Assessment Marks:40
Exam Marks:100	Exam Duration: 2 Hrs.

Course Outcomes (COs):

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

- To design and develop Windows-based business applications using Visual Basic .NET programs that meet commercial programming standards.
- Design and code with the satisfactory results.
- Work is equivalent to that expected from someone already working in the information technology field as a professional programmer.
- Grade you according to commercial standards.

CONTENT	Hours
UNIT 1	
Introduction: .NET Definition, Characteristics, Application, Difference between c and c#, The, the .NET framework, benefits of the .NET approach, C# and .NET, .NET Architecture-Common Language Runtime, MSIL, .NET Framework Classes. Advantages of Managed Code, Garbage Collection, Security, program structure , command line argument, Literals, variables and data types , constant variables, scope of variables, boxing and unboxing.	8
UNIT 2	
Features of C# and OOP: Operators in C#, expression, Decision making and looping statements in c# Methods in C# -declaring methods, nesting of methods, methods parameters, the output parameters, Static data members, Static member function, Arrays - variable size arrays, array list class, String handling . Overview of OOP , Defining a class, Adding Variables and Methods, Member Access Modifiers, Creating Objects, Accessing Class Members.	9
UNIT 3	

Object Oriented C#: Types of Constructors ,This Reference, , Inheritance and polymorphism : classical inheritance, containment inheritance, defining a subclass, visibility control, defining subclass constructors, multilevel inheritance, hierarchical inheritance, overriding methods, hiding methods, abstract classes, abstract methods, sealed classes : Preventing inheritance, sealed methods, polymorphism. Interfaces : Multiple Inheritance : defining an interface, extending an interface, implementing interface, interface & inheritance.	9
UNIT 4	
Introduction to Visual Studio and Exceptions- ASP.NET. Difference Between ASP and ASP.NET, Creating A web application using ASP.NET ,Components of an ASP.NET Custom Control, Deploying ASP.NET application .Errors and Exception Handling, Exceptions Classes User Defined Exceptions , Delegate- delegate declaration, delegate methods, delegates instantiation, delegate invocation, using delegates.	8
UNIT 5	
Introduction to Data Access- Comparison between ADO & ADO.NET—The difference between Connection Model & Disconnected Model – difference between the DataSet and RecordSet- The Dataset Model. Accessing Data using ADO.NET: dataset-DataAdapter DataRelation.	8

TEXT BOOKS:

1. Simon Robinson, Christian Nagel, Karli Watson, Jay Glynn, Morgan Skinner and Bill Evjen, Professional C#, Wiley – dreamtech India Pvt. Ltd., 3rd Edition.
2. C# 8.0 and .NET, Core 3.0 is a book written by Mark J 4th Edition.
3. .NET(Core Reference) Microsoft® Visual C#® 2005: The Language by Donis Marshall
4. Programming in C#, E. Balagurusamy, 2017, McGraw Hill Education
5. C# 4.0 complete reference by Herbert Schildt, from McGraw Hill Education
6. Beginning ASP.NET in C#, Matthew MacDonald.
7. HANDS ON WITH ASP.NET MVC - COVERING MVC 6, Rahul Sahay.
8. Designing Microsoft ASP.NET Application, Jonathan Goodyear, Brain Peek.

BLUE PRINT

UNIT Nos.	No. OF HOURS	Total Marks for which questions to be asked (Including Bonus Questions)
UNIT-1	9	9
UNIT-2	6	10
UNIT-3	9	19
UNIT-4	9	19
UNIT-5	9	19
TOTAL	42	76
Maximum marks for the paper (Excluding bonus question) = 60		

Course Code: CA3P2	Course Title: C# and DOTNET LAB
Course Credits: 02	Hours/Week:04
Total Contact Hours:52	Formative Assessment Marks: 25
Exam Marks:25	Exam Duration: 03 Hours

Course Outcomes (COs):

This course will enable students to

- To design and develop Windows-based business applications using Visual Basic.NET programs that meet commercial programming standards.

Part A

1. Create a project to demonstrate the working of command line arguments for constant and read only using command prompt.
2. Create a project to demonstrate the working of boxing and unboxing by convert integer to object and vice versa.
3. Create a project to demonstrate the working of scope of the variables for the following access modifier.
 - a) Public
 - b) Protected
 - c) Private
 - d) Internal
 - e) Protected internal.
4. Create a project to demonstrate the working of different control statement used in c# with support to a generate shopping bill will contain bill number, name of clerk preparing the bill, each item with quantity and price and total amount to be paid. Total items in the bill are varying can be accommodated in a single bill. There is an array describing each item with a price. The price is to be picked up from that.
5. Create a project to demonstrate the working of looping statement used in c# to select an option from the menu. Upon selecting a menu item, we will be taken to concerned options. Here, if we don't select any options, it just hangs out. So the condition(menu) must be displayed, before its usage.
6. Given below is a hypothetical table showing rates of Income Tax for male citizens below the age of 65 years:

Taxable Income (T1) in	Income Tax in
Does not exceed 1,60,000	Nil
Is greater than 1,60,000 and less than or equal to 5,00,000	$(T1 - 1,60,000) * 10\%$
Is greater than 5,00,000 and less than or equal to 8,00,000	$[(T1 - 5,00,000) * 20\%] + 34,000$
Is greater than 8,00,000	$[T1 - 8,00,000] * 30\% + 94,000$

Write a program to input the age, gender (male or female) and Taxable Income of a person. If the age is more than 65 years or the gender is female, display "wrong category". If the age is less than or equal to 65 years and the gender is male, compute and display the Income Tax payable as per the table given above.

7. Create a project to demonstrate the working of

a. Write a program to input the length and breadth of a rectangle and find its diagonal.

using method $diagonal = \sqrt{length^2 + breadth^2}$

b. Write a program to pass 2 integer numbers as parameters. If either of the two numbers is 0, display invalid entry and the program should end, if it is valid entry, divide the larger number with the smaller number and display the result.

8. Create a project to demonstrate the working of

a) Swapping of two values using ref keyword.

b) Method overloading for displaying int and character data.

9. Create a project to demonstrate the working of nesting of method to reverse a number using nested method for new modifier.

10. Write a program to read the data using one-dimensional array and determine the following.

a. Total marks obtained.

b. The highest marks and lowest marks obtained subject wise.

11. Create a project to demonstrate the working multi-dimensional array for matrix multiplication.

12. Create a project to demonstrate the working of jagged array to input and display elements in the calendar format.

13 a. Write a program in C# Sharp to read a string through the keyboard and sort it using bubble sort

b. Create a project to demonstrate the working of immutable string operation and find the time taken for the immutable string manipulation.

14. Project to implement the access details of the banking information system using the access modifier.
15. Project to design a calculator with all basic functionality using delegates.

Part B

1. Project to implement achieve following goals using single level inheritance.
 - a. Name, Price, Processor, Ram and Hard drive should be defined in base class as constant.
 - b. You need to inherit these functionalities in your program and Print Details.
 - c. All the laptop should have different name, price, processor, ram and hard drive.
2. Create project to implement the following using multi-level inheritance.
 1. Method perimeter () and Area() should be defined in base class Shape.
 2. Design two classes called Circle and Rectangle that implements shape
 3. To compute area of respective shapes.
3. Write a program for creating a Car in which Engine, Tyre, Sound System should be defined in a base class. Your program should inherit all the functionality and gives custom color to a car using interface.
4. Define a class Student as given below using abstract class:

Data members/instance variables:
name, age, m1, m2, m3 (marks in 3 subjects), maximum, average

Member methods:
A parameterized constructor to initialize the data members.
To accept the details of a student.
To compute the average and the maximum out of three marks.
To display the name, age, marks in three subjects, maximum and average.

Write a main method to create an object of a class and call the above member method.
5. Write a menu driven class to accept a number from the user and check whether it is a Palindrome or a Perfect number using multiple interfaces.
 - a. Palindrome number- (a number is a Palindrome which when read in reverse order is same as read in the right order) Example: 11, 101, 151, etc.
 - b. Perfect number- (a number is called Perfect if it is equal to the sum of its factors other than the number itself.) Example: $6=1+2+3$
6. Design a class to overload a function compare() as follows:

void compare(int, int) — to compare two integers values and print the greater of the two integers.

void compare(char, char) — to compare the numeric value of two characters and print with the higher numeric value.

void compare(String, String) — to compare the length of the two strings and print the

Course Code: CA3322	Course Title: DATA COMMUNICATION AND COMPUTER NETWORKS
Course Credits: 03	Hours / Week:03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 02Hrs

Course Outcomes (COs):

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

- Have a good understanding of the OSI Reference Model and in particular have a good knowledge of Layers 1-3.
- Analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies .

Content	Hours
Unit -1	
DATA COMMUNICATION: Components of Data Communication – Direction of Data flow, Networks – Components and Categories – Types of Connections –Protocols and Standards – ISO / OSI model,; Transmission modes, Transmission Media, Switching- Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks. Self-study : Topologies	9
Unit -2	
DATA TRANSMISSION SYSTEMS: properties of media and digital transmission Systems – Twisted Pair , Coaxial Cable, Optical Fibre, Radio Transmission Infrared Light , Multiplexing – frequency – Division , Time Division , SONET; Wavelength Division Multiplexing Circuit switches;	6
Unit -3	
DATA LINK LAYER: Introduction, Framing, and Error – Detection and Correction – Parity – LRC – CRC Hamming code, detection and correction – Error detection , Two dimensional parity checks , Internet checksum , Polynomial code; standardized Polynomial codes , Error detecting capability of a polynomial code Flow and Error Control, Noiseless Channels, Noisy Channels, FDDI, ALOHA, CSMA/CD, Random access, Controlled access, Channelization .	9

Unit - 4	
NETWORK LAYER -Logical Addressing, Internetworking, Tunneling, Address mapping, ICMP, IGMP, Forwarding, Uni-Cast Routing Protocols, Multicast Routing Protocols, Dynamic routing: RIP,OSPF,BGP,CIDR.	9
Unit -5	
TRANSPORT LAYER AND APPLIATION LAYER: Process to Process Delivery, UDP and TCP protocols, IP, TCP, UDP headers Data Traffic, Congestion, Congestion Control, QoS, Integrated Services, Differentiated Services, and QoS in Switched Networks. : Domain name space, DNS in internet, electronic mail, SMTP, MIME,FTP,WWW, HTTP, SNMP,NFS protocols. Self Study: DNS, FTP,HTTP,SNMP	9

Text Books:

1. Data Communications and Networking, Behrouz A. Forouzan , Fifth Edition.
2. Data and Computer Communication, William Stallings, Eighth Edition, Pearson Education,
3. Computer Networks, Andrew S Tanenbaum, 5th Edition. Pearson Education, PHI.
4. An Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson Education
5. Understanding communications and Networks, 3rd Edition, W.A. Shay, Cengage Learning.

BLUE PRINT

UNIT Nos.	No. Of Hours	Total Marks for which questions to be asked (Including Bonus Questions)
UNIT-1	9	10
UNIT-2	6	9
UNIT-3	9	19
UNIT-4	9	19
UNIT-5	9	19
TOTAL	42	76
Maximum marks for the paper (Excluding bonus question) = 60		

SEMESTER IV

Course Code: 4122	Course Title: Python Programming
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 02Hrs

Course Outcomes (COs):

This course focuses on structure and components of Python program. It emphasis on designing object-oriented program with Python classes. At the end of this course the student will be able to:

- Understand the need of Python in various fields.
- Get the knowledge in solving the complex problems with optimized solutions.
- Integrate and connect various real time projects in achieving multi-disciplinary fields.
- Understand the concept of object-oriented programming paradigm and its applications.
- Understand the data manipulations and built-in functions.

Content	Hours
Unit - 1	
Basics of Python program Printing text, printing the result of a calculation, Strings in Python, The Escape Character, Variables and Types, Numeric Data Types in Python, Numeric Operators, Expressions, Integer division, Operator Precedence. Programming examples for each concept. String and Slice: The String Data Type, Negative Indexing in Strings, Slicing, Slicing with Negative Numbers, Using a Step in a Slice, Slicing Backwards, String Operators, String Replacement Fields, String Formatting. Programming examples for each concepts.	8
Unit - 2	
Program Flow Control & Functions Conditional blocks using if, else and elif, for loops in, for loop using ranges, use of while loops in python, Loop manipulation using pass, continue, break and else. Organizing python codes using functions, organizing python projects into modules, importing own module as well as external modules, Understanding Packages.	8
Unit - 3	
String, List, Dictionary & File handling: Understanding in-built string methods, List manipulation and Dictionary manipulation. Programming using string, list and dictionary in-built functions. Reading files in python, writing files in python, Understanding read and write functions, manipulating file pointer using seek, Programming using file operations.	9
Unit - 4	
OO Programming & Regular Expressions: Concept of class, object and instances, constructor, class attributes and destructors,	9

Inheritance, overlapping and overloading operators, Adding and retrieving dynamic attributes of classes. Pattern matching and searching, Power of pattern searching using regex in python, Pattern finding programs using regular expression.	
Unit – 5	
Exception Handling & Database Interaction: Avoiding code break using exception handling, Safe guarding file operation using exception handling, Handling and helping developer with error code. SQL Database connection using python, Creating and searching tables, Reading, storing and updating information on database.	8

REFERENCES:

1. “Python Programming: A Complete Guide for Beginners to Master, Python Programming Language” by Brian Draper
2. “Python Programming for Beginners: Python Programming Language Tutorial” by Joseph Joyner

BLUE PRINT

Unit Nos.	Number of Hours	Total marks for which the question are to be asked (including bonus questions)
UNIT I	8	9
UNIT II	8	10
UNIT III	9	19
UNIT IV	9	19
UNIT V	8	19
TOTAL	42	76
Maximum marks for the paper (Excluding bonus questions) = 60		

Course Code:CA4P1	Course Title: Python Programming Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours:52	Formative Assessment Marks: 25
Exam Marks:25	Exam Duration: 03Hours

Course Outcomes (COs):

This course will enable students to

- Understand the menu driven program and its approach in making the program more interactive.
- Implement the Object Oriented Programming Paradigm in solving real world problems.
- Analyze the various features like built-in library and data structure to operate on the various types of data.

Python Program List

(Do not use inbuilt functions wherever possible)

1. Create a Menu Driven Program using Python that takes a number as input and prints the following:
 - a. Reverse of the given number
 - b. Sum of Digits
 - c. Palindrome or Not
 - d. Series "1 + 2 +.... +n="
 - e. Smallest Divisor of the given number

2. Create a Menu Driven Program using Python that takes a string as input and prints the Following:
 - a. Reverse the string
 - b. Length of the string
 - c. Number of Vowels in a String
 - d. Number of Uppercase Letters and Lower Case Letters in a String
 - e. New string that does not have any duplicate characters

3. Create a Menu Driven Program using Python that takes a List of Numbers from the user and does the following:
 - a. Print the Second Largest Number in the List
 - b. Swap the first and last items of that list and print
 - c. Remove the duplicate members in that list and print.
 - d. Print the number Occurring Odd Number of Times in a List
 - e. Print the list which has the square of the numbers in the given List.

4. Create a Menu Driven Program using Python that takes two lists from the user and does the following:
 - a. Print the Union of the Lists
 - b. Print the Intersection of the Lists
 - c. Print the Merged and sorted List
 - d. Print the sum of all the numbers in both the lists
 - e. Print the average of all numbers in both the lists

5. Create a Menu Driven Program using Python that does the following:
 - a. Creates and Prints a Dictionary with numbers and their squared values till 10
 - b. Print the sum of all values in the Dictionary
 - c. Ask a number from the user and Remove it from the Dictionary and Print

- d. Print all items in the dictionary whose values are above 50
6. Create a Menu Driven Program using Python that takes a number from the user does the following Using Recursion:
 - a. Print Fibonacci Series
 - b. Print Factorial of that number
 - c. Print the Sum of the digits in that number
 - d. Print if the given number is Prime Number or Not
7. Create a Menu Driven Program using Python that does the following file operations:
 - a. Create a File named 'zen_of_python.txt and write the data that comes from import this into the created file (Use Exclusive Creation).
 - b. Print all the data in that file
 - c. Print total number of words in that file.
 - d. Capitalize First Letter of Every word in that file
 - e. Print the contents of the file in reverse order
8. Create a Menu Driven Program using Python that does the following:
 - a. Print the current Date and Time (DD-MM-YYYY HH:MM: SS AM/PM)
 - b. Print all the Sundays of a specified year
 - c. Print new Date after adding 28 days to the current Date
 - d. Print the Day of the given Date
 - e. Calculate Age by taking the Date of Birth
9. Create a Menu Driven Program in Python that takes a string and checks if it follows the regular expressions given below:
 - a. Matches a string that has an 'a' followed by anything, ending in 'b'
 - b. Print only a 4-digit number from a message as string
 - c. Convert a date of yyyy-mm-dd format to dd-mm-yyyy format
 - d. Find all five characters long word in a string
 - e. Split a string with multiple delimiters
10. Create a User Defined Exception for Age, that is raised if the given number is not above 18 and below 100. Use this exception in a program where it is raised programmatically after accepting a Number.
11. Create a Menu Driven Program in Python that prints the following pattern for the given number: (Refer Pattern List in Google Classroom)
 - a. Pattern #2: Inverted Pyramid of Numbers
 - b. Pattern #6: Reverse Pyramid of Numbers
 - c. Pattern #8: Unique Pyramid Pattern of Digits.
 - d. Pattern #11: Pyramid of Horizontal Tables,
 - e. Pattern #15: Downward Triangle Pattern of Stars
12. Create a Menu Driven Program in Python that does the following:
 - a. Accept a sentence and reverse each word in it

- b. Take a string input and sort them based on the number of characters in each word and mention the count of letters in a word at the end.
- c. Take a number as string type and convert it into an integer without using inbuilt functions
- d. Take a String and append a digit after every character with total count of that character in that string (ignore case)

13. You are given input file input_digits_P1.txt and input_words_P1.txt which contains Comma separated random integers. E.g. 2, 3, 10, 8, 7, 20, 11, 25, 30, 17, 12.

- Read the integers from the input_digits_P1.txt file and calculate the Sum of the Odd and Even numbers.
- Create a file with the name sum_of_odd_numbers.txt and store the sum of odd Numbers in that file. E.g. 63 Create a file with the name sum_of_even_numbers.txt and store the sum of even numbers in that file.e.g. 82
- In the same program read the words from input_words_P1.txt (E.g.: Apple, Ball, Cat, Dog, Egg) separated by comma and generate the below output using the digits read in Output:

2 Apple, 10 Ball, 8 Cat, 20 Dog, 30 Egg, 12 Apple Save the final output in a file final_output.txt.

14. Create a Menu Driven Program using Python that performs the following using the Database Structure mentioned below: Table Name: Student Fields: student_id, student_name, student_course, student_email, student_dob

- a) Add a new Student and Display the Details
- b) Edit a Particular Student Detail and display the details.
- c) Show All Student Details
- d) Show One Student Detail e) Delete a Student Record.

15.a. Write a Python program that matches a string that has an *a* followed by zero or more b's.

b. Write a Python program to find the sequences of one upper case letter followed by lower case letters.

c. Write a Python program to find the substrings within a string.

d. Write a Python program to convert a date of yyyy-mm-dd format to dd-mm-yyyy format.

Course Code: 4222	Course Title: Computer Multimedia and Animation
Course Credits: 03	Hours/Week: 03
Total Contact Hours:42	Formative Assessment Marks: 40
Exam Marks:100	Exam Duration: 3 Hrs.

Course Outcomes (COs):

This Course emphasis on Multimedia Basic Tools and various techniques. At the end of the course the student will be able

- Will improve the skills in Animation and Videos.
- Understand the requirement of multimedia in making the presentations more live and attractive.
- Identity the high scope and requirements in all the disciplines, in developing teaching aids and other presentations.
- Improve the creative skill in making movies and animations by using various tools and techniques.

Course Content

Content	Hours
Unit - 1	
Introduction to Multimedia: History of Multimedia, its market – Content copyright – Resources for multimedia developers – Types of produces – Evaluation – Hardware Architecture – OS and Software – Multimedia Architecture – Software library – Drivers.	6
Unit - 2	
Computer Fonts and Hypertext: Usage of text in Multimedia, Families and faces of fonts, outline fonts, bitmap font International character sets and hypertext, Digital fonts techniques. Text file formats and Compression techniques.	6
Unit - 3	
Image fundamentals and representations Colour Science , Colour, Colour Models, Colour palettes, Dithering, 2D Graphics, Image Compression and File Formats :GIF, JPEG, JPEG 2000, PNG, TIFF, EXIF, PS, PDF, Basic Image Processing [Can Use Photoshop], Use of image editing software, White balance correction, Dynamic range correction, Gamma correction, Photo Retouching.	10
Unit - 4	
Audio fundamentals and representations Digital Audio : Audio software and hardware, Characteristics of sound and Digital audio Digitization of sound, frequency and bandwidth, decibel system, data rate, audio file format, Sound synthesis, MIDI, wavetable, Digital Audio systems, Audio file formats – Using Audio in Multimedia Applications – Audio for content. Compression and transmission of audio on Internet, Adding sound to your multimedia project.	10
Unit 5	
Video and Animation Video Basics , Video software and hardware, How Video Works, Broadcast Video	10

Standards, Analog video, Digital video, Video Recording and Tape formats, Shooting and Editing Video (Use Adobe Premier for editing), Video Compression and File Formats. Video compression based on motion compensation, MPEG-1, MPEG-2, MPEG-4, MPEG-7, MPEG-21, Animation: Cell Animation, Computer Animation, Morphing. Video capture and playback systems.	
---	--

Text Books:

1. Tay Vaughan, "Multimedia making it work", Tata McGraw-Hill, 2008.
2. Rajneesh Aggarwal & B. B Tiwari, "Multimedia Systems", Excel Publication, New Delhi, 2007.
3. Li & Drew, " Fundamentals of Multimedia" , Pearson Education, 2009.
4. Parekh Ranjan, "Principles of Multimedia", Tata McGraw-Hill, 2007 2. Anirban Mukhopadhyay and Arup Chattopadhyay, "Introduction to Computer Graphics and Multimedia", Second Edition, Vikas Publishing House
5. The book of Audacity : record, edit, mix, and master with the free audio editor
Schroder, Carla. San Francisco : No Starch Press; c2011
6. Creating DSLR video : from snapshots to great shots
Harrington, Richard, 1972-Berkeley, CA : Peachpit Press; 2012.

BLUE PRINT

Unit Nos.	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	6	9
Unit II	6	10
Unit III	10	19
Unit IV	10	19
Unit V	10	19
TOTAL	42	76
Maximum marks for the paper (Excluding bonus question)= 60		

Course Code:CA4P2	Course Title: Computer Multimedia and Animation Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours:52	Formative Assessment Marks: 25
Exam Marks:25	Exam Duration: 03Hours

Course Outcomes (COs):

This course will enable students to:

- Understand the different open source for the presentation and animations.
- Understand the need of Multimedia in making audio and videos.
- Analysis the various tools and techniques and develop various project that can used in the real world concepts.

Practical Program list

1. Creating presentation with animation include text, images and sounds. Any advanced opensources. (3 different Slide Presentation)
2. Creating Brochure with images and text using any open source tools. (any open sources likecanva etc.(3 Different Events Brochure example conference)
3. With the help of Quiz maker use any 2 open source to create a quiz with 10 questions.
4. Using Flash perform different operations (rotation, scaling move etc.) on objects
5. Demonstrate a step for live stream any video. (Use OBS studio, zoom / google meet).
6. Demonstrate various screen recording tools to use it as a teaching aid.
7. Creating Audio and video, rendering, Editing, Trimming, Video files.
8. Create a video and demonstrate merging and any two transitions. Compress with variousonline open sources
9. Creating a moving object in flash include 10 key frames.
10. Creating one sample home page and links by using flash or Dreamweaver.
11. Create a storyboard presentation consisting of 10 slides using any storyboard open sources.
12. Using movie maker open source create 5 minutes documentary.
13. Create a Various object and use different effects using tools in Flash.
14. Create Scenery using Flash tools and implement five animations.

Course Code: CA4322	Course Title: Operating System Concepts
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 3 Hrs.

Course Outcomes (COs):

- Understand the different services provided by Operating System at different level.
- They learn real life applications of Operating System in every field.
- To acquire the fundamental knowledge of the operating system architecture and components.
- Understands the use of different process scheduling algorithm and synchronization techniques to avoid deadlock.
- They will learn different memory management techniques like paging, segmentation and demand paging, Device and file management techniques etc.

CONTENT	HOURS
UNIT 1:	
INTRODUCTION: Definition, functions, operating system structure, types, Buffering & spooling, multiprogramming, time - sharing, Real time system, distributed systems, parallel systems protection, system components, system calls.	8
UNIT 2:	
PROCESS MANAGEMENT: Process concept (program, process and thread), states of a process, process control block, functions, Cooperating process, Inter-process communication, Scheduling Criteria, schedulers-types, (pre-emptive and non-preemptive) FCFS, SJF, Round Robin, priority scheduling, multilevel, multilevel feedback algorithms	10
UNIT 3:	
PROCESS SYNCHRONIZATION AND DEADLOCK: process synchronization – semaphores, critical section problems, classic problems of synchronization, Deadlock ,deadlock characterization, methods for handling deadlock, deadlock prevention, avoidance and detection using resource allocation graph and wait for graph, Recovery from deadlock.	8
UNIT 4:	
MEMORY MANAGEMENT: Functions, Different Schemes-Single Continuous-Partitioned -Multiple Relocatable-Paging-Demand Paging-Segmentation-Paged segmentation. Virtual Memory management: Demand paging, Page replacement and page replacement algorithm (FIFO, LRU), Segmentation and thrashing.	8
UNIT 5:	
DEVICE AND FILE SYSTEM: Disk structure, allocation methods, free space management, need for disk scheduling, scheduling algorithm-FCFS (First Come, First Served), SSTF (Shortest Seek Time First), SCAN, C-SCAN (circular scan), file system – definition, access methods, directory structure, organization, file protections.	8

Text Books:

1. Operating System by Milan Milenkovic, McGraw Hill.

References:

1. Operating System by Madnick and Donoval, McGraw Hill.
2. Operating Systems: Internals and Design Principles, by William Stallings, seventh edition
3. Operating System Concepts by James L Peterson. (2nd Edition)
4. Operating System Design and Implementation by Andrew S Tenenbaum. (3rd Edition).

BLUE PRINT

UNITS	NO OF HOURS	Total Marks for which questions to be asked (Including Bonus Questions)
UNIT-1	9	10
UNIT-2	6	9
UNIT-3	9	19
UNIT-4	9	19
UNIT-5	9	19
TOTAL	42	76
Maximum marks for the paper (Excluding bonus question) = 60		

Course Content for V Semester BCA

Course Code: CA 5123	Course Title: E Commerce
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives

1. To provide students with an overview and understanding of e-commerce with a specific emphasis on Internet Marketing.
2. Explain the process that should be followed in building an E-commerce presence.
3. To explore the major issues associated with e-commerce-security, privacy, intellectual property rights, authentication, encryption, acceptable use policies, and legal liabilities

Course Outcomes (COs):

1. Define the major types and themes of e-commerce and to understand the evolution of e-commerce with implication in academic disciplines.
2. Identify the key components of e-commerce business models, to describe major B2C and B2B Models and to understand business key concepts and strategies.
3. Discover E commerce presence and relate SWOT analysis to analyze security threats and management policies for online payment systems.
4. Understand the key features of the Internet audience and the basic concepts of consumer behaviour and to Identify and describe the basic digital commerce marketing and advertising strategies and tools, understand the costs and benefits of online marketing communications.
5. Infer and articulate the Supply chain Management and collaborative commerce.

Content	Hours
Unit - 1	
Introduction to E Commerce Unique features of E Commerce technology, Types of E Commerce, Historical Evolution of E Commerce, Understanding E Commerce: Organizing themes. Academic disciplines concerned with E Commerce. Case study: Pinterest: A Picture is worth of Thousand words.	8
Unit-2	
E Commerce Business Models Introduction, eight key elements of a Business Model, Crowdfunding, Major Business-to-Consumer(B2C) Business Models, Major Business-to-Business(B2B) Business Models, Impact of E Commerce in Business: Strategy, Structure and Process. The Internet Technology Background, The Internet Today, The future Internet Infrastructure, The World Wide Web, The Internet, and the Web: Features. Case study: Freemium: Takes Pandora Public.	8

Unit-3	
E Commerce Presence: E-Commerce Presence Map, Timeline, SWOT Analysis, Building an E Commerce Presence: A Systematic Approach, The E-Commerce security environment, Security threats in E Commerce, Technology Solutions, Management Policies, Business procedures and Public Laws, Payment System, E Commerce Payment System, Electronic presentment, and payment. Case study: BITCOIN	10
Unit-4	
E Commerce Marketing and Advertising Concepts Consumer Online: Internet Audience and Consumer Behaviour, Digital commerce marketing and advertising: Traditional Online Marketing and Advertising Tools. Social, Mobile and Local Marketing and Advertising. Internet Marketing Technologies: Web Transaction logs, Supplementing the logs, Databases: Data Warehouses, Data Mining and Big Data. Marketing automation and customer relationship management (CRM) systems. Understanding the costs and benefits of online marketing communications.	10
Unit-5	
Supply Chain Management and Collaborative Commerce An Overview of B2B E-Commerce, Procurement Process and Supply Chain, Trends in Supply chain Management and Collaborative Commerce.	6

Text Book:

Kenneth C. Laudon and Carol Guercio Traver:” E-Commerce: Business, Technology, and Society”, Pearson,2017.

Reference Books

1. Daniel Amor: “E Business R(Evolution)” Pearson Edude,2020
2. Krishnamurthy: “E-Commerce Management” Vikas Publishing House, Vol 7 , 9 July 2002.
3. David Whitely: “E-Commerce: Strategy, Technologies and Applications” Tata McGraw Hill, July 2010
4. P. T. Joseph: “E-Commerce: A managerial Perspectives” Tata McGraw Hill,2004.

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	8	10
Unit 2	8	10
Unit 3	10	22
Unit 4	10	22
Unit 5	6	14
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code: CA 5223	Course Title: Internet Technologies.
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives:

1. Students will be able to design an interactive web page using Scripting languages.
2. Students will be able to distinguish Single Page React application and traditional web development frameworks.
3. Students will be able to differentiate between static web pages and dynamic webpages.
4. Students will be able develop an application from scratch using React JS.
5. Students will be able to create, build, and deploy react applications with server-side programming.

Course Outcomes:

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

1. Design simple web pages using markup languages like HTML.
2. Create dynamic web pages using Java script, React JS which are easy to navigate and use.
3. Build complex user interfaces having a unidirectional data flow, with React.js.
4. Program server-side web pages that have to process request from client- side web pages.

Content	Hours
Unit - 1	
HTML: Introduction to HTML, HTML Syntax, Semantic Markup, Structure of HTML Documents, Quick Tour of HTML Elements, HTML5 Semantic Structure Elements, formatting tags, Header elements, font tag, List tags, Image & anchor tags. HTML Tables and Forms: Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, Table and Form Accessibility CSS: Introduction to CSS, CSS Syntax, Location of Styles, Selectors, The Cascade: How Styles Interact, The Box Model, CSS Text Styling.	9
Unit-2	
JavaScript: Client-Side Scripting, Introduction to JavaScript- Design Principles, Syntax, JavaScript Objects, Syntax-Variables and Data Types-Statements-Operators-Literals-Functions-Objects-Arrays-Built-in Objects -The Document Object Model (DOM), JavaScript Events, Forms.	8

Unit-3

ReactJS :

ES6 Overview

Introduction to ES6, ES6 lable, Block scope, let & const, Template literals, Arrow functions, Spread and Rest operators, Destructuring

Classes - Inheritance, Static properties and methods, Inheritance. Promises, Modules

Introduction to React:

Introduction to React-React version history-React 16 vs React 15-Just React - Hello World- Using create-react-app-Anatomy of react project- Running the app- Debugging first react app.

8

Unit-4

Templating using JSX:

Working with React- create Element- Expressions- Using logical operators- Specifying attributes.

Specifying children components- Significance of component architecture.

Types of components - Functional, Class based, Pure.

Working with state and props:

Introduction to state and its significance- Read state and set state- Passing data to component using props-Validating props using prop Types.

Rendering lists:

Using React key prop-Using map function to iterate on arrays to generate elements.

8

Unit-5

Event handling in React:

Understanding React event system, Understanding Synthetic event- Passing arguments to event handlers.

Working with Forms

Controlled components-Uncontrolled Components-Understand the significance to default Value prop- Using react ref prop to get access to DOM element.

Data Base Connection:

Establishing connection, selecting database, Query operations Using MySQL.

Case study:

Connect front-end with any data bases.

9

Text Books:

1. HTML, CSS, and JavaScript All in One: Covering HTML5, CSS3, and ES6, Sams Teach Yourself Paperback – Illustrated, 9 January 2019.
2. Learning React: Functional Web Development with React and Redux 1st Edition,2017.

References:

1. MASTERING HTML, CSS & Java Script Web Publishing Paperback – 15 July 2016.
2. Web Design with HTML, CSS, JavaScript and JQuery Set Paperback – 15 August 2014.
3. HTML To React: The Ultimate Guide.

BLUEPRINT

Chapter	No. of Hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	9	10
Unit 2	8	10
Unit 3	8	22
Unit 4	8	22
Unit 5	9	14
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code: CA5P1	Course Title: Internet Technologies Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hrs

Course Objectives:

1. Students will be able to design an interactive web page using HTML5, Java script.
2. Students will be able to distinguish Single Page React application and traditional web development frameworks.
3. Students will be able develop an application using React 16.
4. Students will be able to create react applications with server-side programming.

Course Outcomes:

Students can acquire the following skills after completing the course:

1. To able to develop interactive web pages.
2. To Demonstrate various open sources in managing data.
3. Importing and exporting data across the web pages.
4. Perform thorough Form Validation.
5. Able to connect the Front-end with back-end.

PROGRAM LIST

1. Create your profile page i.e., educational details, Hobbies, Achievement, My Ideals etc. using HTML lists.
2. Use table tag to format web page. Also create the Time Table of your class using table tag.
3. Write HTML code to build a Reservation form.
4. Write HTML code to build an application form.
5. Create a home page using CSS concepts to add a menu and various text, image and colour properties.
6. Create Style sheet to set formatting for text tags and embed that style sheet on web pages created for your site.
7. Develop and demonstrate the usage of inline, internal and external style sheet using CSS.
8. Write a JavaScript program to determine whether a given year is a leap year in the Gregorian calendar.
9. Write a JavaScript program to convert temperatures to and from Celsius, Fahrenheit.
10. Write *JavaScript* to validate the following fields of the Registration page.
 1. First Name (Name should contains alphabets and the length should not be less than 6 characters).
 2. Password (Password should not be less than 6 characters length).

Project Development Using React JS.

11. Prepare Project for Weather Finder App using React Js
12. Prepare Project for Bill and Discount Calculator using React JS.

Evaluation Scheme for Lab Examination

Assessment Criteria	Marks
Writing	8
Execution	10
Viva Voice	7
Total	25

Course Code: CA5323	Course Title: Cyber Law & Cyber Security
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives:

1. To provide an understanding of the main issues related to security in modern networked computer systems.
2. To understand concepts and foundations of computer security.
3. Basic knowledge about security-relevant decisions in designing IT infrastructures like personal laptop to large-scale infrastructures.

Course Outcomes:

Here the students will be able to understand

1. An extensive, detailed, and critical understanding of the concepts, issues, principles and theories of computer network security.
2. Theoretical and detailed practical knowledge of a range of computer.
3. Network security technologies as well as network security tools and services.
4. Gain experience of analysing, designing, implementing and validating solutions to computer network security challenges using common network security tools and formal methods.
5. Demonstrate knowledge of reasoning and knowledge representation for solving cybercrime problems.

CONTENT	Hours
UNIT-1	
Cyber Laws Internet Governance – Challenges and Constraints, Cyber Threats: - Cyber Warfare-Cyber Crime-Cyber Terrorism-Cyber Espionage, State and Private Sector in Cyberspace, Cyber Security Standards. The INDIAN Cyberspace, The Indian Penal code, National Cyber Security Policy 2017.	7
UNIT-2	
Cyber Security Introduction to Cyber Security. Confidentiality, Integrity and Availability – Triad. Attacks: Threats, Risk Assessment and Analysis, Hackers and types, Information Classification, Policies, Procedure and Guidelines, Vulnerabilities and Risk, Layers of cyber security. Self-study: Best cyber security practices.	8
UNIT-3	

<p>Basics of Cryptography</p> <p>Symmetric and Asymmetric Cryptosystems, Classical Encryption Techniques – Substitution Techniques, Transposition Techniques. Data Encryption Standard (DES), Principles of public key cryptosystems-The RSA Algorithm-Key management - Diffie Hellman Key exchange, Hashing, MD5 and SHA-1 Algorithms.</p> <p>Self-study: SHA 512 Algorithm</p>	10
UNIT-4	
<p>Network and Wireless Attacks</p> <p>Network Sniffing, packet analysis, display and capture filters, Ettercap, DNS Poisoning, ARP Poisoning, Denial of services, Vulnerability scanning, Setup network, IDS/IPS, Router attacks, Man-in-the-middle Attack, MAC Filtering, Packet Encryption, Packet Sniffing, Types of authentication, Attacks on WEP, WPA Encryption.</p> <p>Case study: Fake hotspots.</p>	7
UNIT-5	
<p>Network Security</p> <p>IP security architecture, Security protocols, IP Sec, Firewalls, IDPS – Types and technologies. Trusted systems – Electronic payment protocols, SET</p> <p>Network Security Applications, Authentication, Mechanisms: Passwords, Cryptographic authentication protocol, Kerberos, X.509 certificate, Digital Signatures. Web Security: SSL Encryption, Securing online payments (OTP).</p> <p>Self-study: Transport layer secure.</p>	10

Text Books:

1. Sunit Belapure and Nina Godbole, "Cyber Security : Understanding Cyber-crimes and Legal perspectives", Wiley India Pvt Ltd,2020.
2. Atul Kahate, "Cryptography and Network Security," Tata McGraw-Hills (Reprinted 2021).

References:

1. William Stallings; "Cryptography and Network Security: Principles and Practices", Fifth Edition, Prentice Hall Publication Inc., 2007.
2. Nina Godbole and Sunit Belapure; "Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives", Wiley Publications, 2011.
3. Matt Bishop, "Computer Security Art and Science", Pearson/PHI, 2002.
4. Michael E Whiteman and Herbert J Mattord; "Principles of Information Security", Vikas Publishing House, New Delhi, 2003.
5. Alfred J. Menezes, Paul. C. Van Oorschot, and Scott A. Vanstone "Handbook of Applied Cryptography", CRC press, Lib of Congress -2006

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	7	10
Unit 2	8	10
Unit 3	10	22
Unit 4	7	22
Unit 5	10	14
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code: CA5P2	Course Title: Cyber Security Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hrs

Course Objectives:

The student should be made

1. Exposed to different Cipher techniques
2. Learn to implement the algorithms DES, RSA, MD5,SHA-1
3. Learn to use security tools Cryptool, GnuPG, KF sensor, VI Strumbler.

Course Outcomes:

Here the students will be able to understand

1. An extensive, detailed and critical understanding of the concepts, issues, principles and theories of computer network security.
2. Theoretical and detailed practical knowledge of a range of computer.
3. Network security technologies as well as network security tools and services.
4. Gain experience of analysing, designing, implementing and validating solutions to computer network security challenges using common network security tools and formal methods.
5. Demonstrate knowledge of reasoning and knowledge representation for solving cybercrime problems.

LIST OF EXPERIMENTS

1. Write a program to encrypt and decrypt a Password.
2. Implement the substitution mono alphabetic technique by using Caesar Cipher algorithm.
3. Write a Program to implement Play fair Cipher algorithm.
4. Write a program to demonstrate polyalphabetic cipher by using Hill Cipher algorithm.
5. Implement Substitution and transposition technique by using Vigenère Cipher algorithm.
6. Write a program to implement Rail fence – row & Column Transformation technique.
7. Demonstrate Data Encryption Standard Algorithm using Cryptool.
8. Implement the RSA Algorithm using Cryptool.
9. Write a program to demonstrate Diffie-Hellman's key exchange.
10. Demonstrate Message digest (MD5) Algorithm.
11. Implement the SHA-1 hashing Algorithm.
12. Demonstrate how to provide secure data storage, secure data transmission and for creating digital signatures (GnuPG).

13. Setup a honey pot and monitor the honeypot on network (KF Sensor).
14. Perform wireless audit on an access point or a router and decrypt WEP and WPA.(VI Stumbler).

Evaluation Scheme for Lab Examination

Assessment Criteria	Marks
Writing	8
Execution	10
Viva Voice	7
Total	25

Course Code: CADE5423	Course Title: Cloud Computing
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives

1. To recognise how the cloud has developed from the earlier technologies.
2. To comprehend the idea of cloud computing.
3. Becoming aware of the numerous problems with cloud computing.
4. To recognise the rise of the cloud as the newest paradigm for computing.

Course Outcomes (COs):

1. Articulate the main concepts, key technologies, strengths and limitations of cloud computing.
2. Learn the key and enabling technologies that help in the development of cloud.
3. Develop the ability to understand and use the architecture of compute and storage cloud, service and delivery models.
4. Explain the core issues of cloud computing such as resource management and security.
5. Be able to install and use current cloud technologies.

Content	Hours
Unit - 1	
<p>Introduction to Cloud Computing</p> <p>Definition and history of cloud computing, Characteristics and benefits of cloud computing. Cloud Deployment models: Public, Private, Hybrid, and Community Clouds.</p> <p>Cloud service models</p> <p>SaaS, PaaS, IaaS, IDaaS.</p>	10
Unit-2	
<p>Data Storage in the cloud</p> <p>Examining the Evolution of Network Storage, Understanding Cloud-Based Data Storage, Advantages and Disadvantages of Cloud-Based Data Storage, Cloud-Based Backup Systems, Understanding File Systems, Industry-Specific Cloud-Based Data Storage, Cloud-Based Database Solutions, Cloud-Based Block Storage.</p>	6
Unit-3	
<p>Virtualization</p> <p>Understanding Virtualization, The History of Virtualization, Leveraging Blade Servers, Server Virtualization, Desktop Virtualization, Desktop Solutions on Demand, Virtual Networks, Data Storage Virtualization, Why Virtualize?</p> <p>Securing the Cloud</p> <p>General Security Advantages of Cloud-Based Solutions, Introducing Business Continuity and Disaster Recovery</p> <p>Disaster Recovery and Business Continuity and the Cloud</p> <p>Understanding the Threats, Understanding Service-Level Agreements, Measuring Business Impact: The Essence of Risk Mitigation, Disaster Recovery Plan Template</p>	12
Unit-4	
<p>Service Oriented Architecture</p> <p>Understanding Service-Oriented Architecture, Understanding Web Service Performance, Web Service and Reuse, Scaling Web Services, Web Services and Loose Coupling, Treating a Web Service as a Black Box, Web Service Interoperability, Governing Web Services.</p>	6

<p>Managing the Cloud</p> <p>Know Your Service-Level Agreement, Ensure and Audit System Backups, Know Your System's Data Flow, Beware of Vendor Lock-In, Determine Technical Support and Help Desk Procedures, Determine Training Procedures, Know the Provider's Security Policies and Procedures, Define the Data Privacy Requirements, Know Specifics About the Economics of the Cloud and Return on Investment, Monitor Capacity Planning and Scaling Capabilities, Monitor Audit-Log, Use Solution Testing and Validation.</p>	
<p>Unit-5</p>	
<p>Migrating to the Cloud</p> <p>Define the System Goals and Requirements, Protect Your Existing Data, Use an Experienced Cloud Consultant, Know Your Application's Current Characteristics, Remember Vendor Lock-In, Define Your Training Requirements, Establish a Realistic Deployment Schedule, Review the Budget Factors, Identify IT Governance Issues, Understanding Cloud Bursting.</p> <p>Mobile Cloud Computing</p> <p>The Evolution of Mobile Computing, The Mobile Cloud Ecosystem, Introducing the Mobile Players, Mobile Development Considerations.</p> <p>Self-Study: The future of the Cloud & Cloud Service Providers.</p>	<p>8</p>

Text Book

Cloud Computing, Kris Jamsa, Second Edition, 2022, Jones & Bartlett Publishers.

Reference Books

1. Cloud Computing: A Practical Approach, Anthony T Velte, Toby J Velte, Robert Elsenpeter, 2017, McGraw Hill Education.
2. Mastering Cloud Computing, Foundations and Application Programming, Raj Kumar Buyya, Christen vecctiola, S Tammaraiselvi, First Edition, McGraw Hill Education
3. George Reese, "Cloud Application Architectures: Building Applications and Infrastructure in the Cloud: Transactional Systems for EC2 and Beyond (Theory in Practice)", O'Reilly, 2009.
4. Rittinghouse, John W., and James F. Ransome, "Cloud Computing: Implementation, Management and Security", CRC Press, 2017.
5. Distributed and Cloud Computing, Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra MK , 2012, Elsevier.
6. Cloud Computing, Theory and Practice, Dan C Marinescu, 2013 Elsevier Inc.

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	10	12
Unit 2	6	10
Unit 3	12	22
Unit 4	6	14
Unit 5	8	20
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code: CAVO1	Course Title: AWS - Cloud Practitioner Essentials
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives

1. To describe what a cloud service provider (CSP) is and the value they bring to computing.
2. To identify basic security and compliance aspects of the AWS platform and the shared security model.
3. To understand the billing, account management, and pricing models.
4. Evaluate basic or core characteristics of deploying and operating in the AWS Cloud
5. Analyse and differentiate between on-premises and cloud infrastructure

Course Outcomes (COs):

1. Define the AWS Cloud
2. Explain the AWS pricing philosophy
3. Identify the global infrastructure components of AWS
4. Describe the security and compliance measures of the AWS Cloud, including AWS Identity and Access Management (IAM)
5. Create a virtual private cloud (VPC) by using Amazon Virtual Private Cloud (Amazon VPC)
6. Demonstrate when to use Amazon Elastic Compute Cloud (Amazon EC2), AWS Lambda, and AWS Elastic Beanstalk
7. Differentiate between Amazon Simple Storage Service (Amazon S3), Amazon Elastic Block Store (Amazon EBS), Amazon Elastic File System (Amazon EFS), and Amazon Simple Storage Service Glacier (Amazon S3 Glacier)
8. Demonstrate when to use AWS database services, including Amazon Relational Database Service (Amazon RDS), Amazon DynamoDB, Amazon Redshift, and Amazon Aurora
9. Explain the architectural principles of the AWS Cloud
10. Explore key concepts related to Elastic Load Balancing, Amazon CloudWatch, and Amazon EC2 Auto Scaling.

Content	Hours
Unit - 1	
Cloud Concepts Overview Introduction to cloud computing, Advantages of the cloud, Introduction to AWS, Moving to the AWS Cloud.	8
Cloud Economics and Billing Fundamentals of pricing, Total cost of ownership, Activity: Simple Monthly Calculator, Delaware North case study, AWS Organizations, AWS billing and cost management, Billing dashboards, Technical support models, Activity: Support Plan Scavenger Hunt.	
Unit-2	

<p>AWS Global Infrastructure Overview</p> <p>AWS global infrastructure, Demo: AWS global infrastructure, AWS services and service categories, Activity: AWS Management Console Clickthrough</p> <p>Cloud Security</p> <p>AWS shared responsibility model, Activity: AWS Shared Responsibility Model, AWS IAM, Demo: AWS IAM Console, Securing a new AWS account, Lab: Introduction to AWS IAM, Securing accounts, Securing data, Working to ensure compliance.</p>	10
<p>Unit-3</p>	8
<p>Networking and Content Delivery</p> <p>Networking basics, Amazon VPC, VPC networking, Activity: Label This diagram, Demo: Amazon VPC Console, VPC security, Activity: Design a VPC, Lab: Build a VPC and Launch a Web Server, Route 53, CloudFront</p> <p>Compute</p> <p>Compute services overview, Amazon EC2 part 1, Amazon EC2 part 2, Amazon EC2 part 3, Demo: Amazon EC2, Lab: Introduction to Amazon EC2, Activity: Amazon EC2 Versus Managed Services, Demo: Amazon EC2 Part Console, Amazon EC2 cost optimization, Container services, Introduction to AWS Lambda, Activity: AWS Lambda, Introduction to AWS Elastic Beanstalk, Activity: AWS Elastic Beanstalk</p>	8
<p>Unit-4</p>	8
<p>Storage</p> <p>AWS EBS, Demo: Amazon Elastic Block Store Console, Lab: Working with EBS, AWS S3, Demo: AWS S3 Console, AWS EFS, Demo: AWS EFS Console, AWS S3 Glacier, Demo: AWS S3 Glacier Console, Activity: Storage Technology Selection</p> <p>Databases</p> <p>Amazon RDS, Demo: Amazon RDS Console, Lab: Build a Database Server, Amazon DynamoDB, Demo: Amazon DynamoDB, Amazon Redshift, Amazon Aurora, Activity: Database case study.</p>	8
<p>UNIT 5</p>	8
<p>Cloud Architecture</p> <p>AWS Well-Architected Framework design principles, Activity: AWS Well-Architected Framework Design Principles, Operational excellence, Security, Reliability, Performance efficiency, Cost optimization, Reliability & high availability, AWS Trusted Advisor, Activity: Interpret AWS Trusted Advisor Recommendations</p> <p>Automatic Scaling and Monitoring</p> <p>Elastic Load Balancing, Activity: Elastic Load Balancing, Amazon CloudWatch, Activity: Amazon CloudWatch, Amazon EC2 auto scaling, Lab: Scale & Load Balance your Architecture</p>	8

Reference Books

1. AWS Certified Cloud Practitioner Study Guide: CLF-C01 Exam, Ben Piper, David Clinton, Sybex Publishers
2. AWS Basics: Beginner's Guides, Gordon Wong, CreateSpace Independent Publishing
3. AWS: The Beginners Guide to Amazon Web Services, Dennis Hutten, CreateSpace Independent Publishing
4. Amazon Web Services for Dummies, Bernard Golden, John Wiley & Sons
5. Learning Amazon Web Services (AWS): A Hands-On Guide to the Fundamentals of AWS Cloud, Mark Wilkins, Pearson Education
6. AWS Certified Cloud Practitioner (CLF-C01) Cert Guide, Anthony Sequeira, Pearson Education

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	8	8
Unit 2	10	10
Unit 3	8	20
Unit 4	8	20
Unit 5	8	20
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) =60		

Course Content for VI Sem BCA

Course Code:CA6123	Course Title : SOFTWARE ENGINEERING
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration:2 Hrs.

Course Objectives:

1. To inculcate in student's different concepts of software engineering principles
2. To analyse the different categories of requirements in a software project, develop the skills necessary to design, develop and execute software projects.
3. To comprehend the aspects of software Quality, software maintenance and various testing strategies
4. To get an overview of the total management spectrum from the perspective of a software project.

Course outcomes:

On completion of the course the student will be able to:

1. Understand the importance of the stages in the software life cycle and various process models.
2. Get an in depth knowledge of the Agile framework with an emphasis on Scrum and XP.
3. Implement the design and developmental aspects using software engineering principles.
4. Will comprehend the important principles of quality and various aspects of quality assurance and different maintenance techniques.
5. Realise the importance of software project management concepts

CONTENT	HOURS
UNIT 1:	
SOFTWARE PROCESS AND SOFTWARE PROCESS MODELS Introduction: Software process, concepts of project and product, process, Life cycle models: Waterfall model, spiral model, Prototyping Model, Agile framework: values and principles of Agile, Agile manifesto, Extreme programming, Scrum Framework.	08
UNIT 2:	

<p>SOFTWARE REQUIREMENTS AND DESIGN</p> <p>Software requirements: Functional- non-functional requirements, User requirement, Software Requirement engineering, Feasibility studies, Requirements elicitation and analysis, requirement validation, software requirement specification (SRS document).</p> <p>Software Design: Design Process, Design strategies, Design characteristics, architectural design: System Structuring, control modelling, Modular decomposition,</p> <p>Software cost estimation techniques: top down, bottom-up approach, estimation by analogy, COCOMO model.</p> <p>Self-Study: Data Flow Diagrams, introduction to function oriented and object-oriented design.</p>	10
UNIT 3:	
<p>SOFTWARE CODING AND SOFTWARE TESTING</p> <p>Software coding: Introduction, coding techniques, coding styles, Coding verification techniques, Coding tools, code documentation,</p> <p>Software Testing: Introduction, verification and validation, A strategic approach to software testing, Unit testing, Black-box and white box and integration testing, system testing.</p>	08
UNIT 4:	
<p>SOFTWARE QUALITY AND SOFTWARE MAINTENANCE</p> <p>Software quality: Introduction, quality metrics, quality assurance: tasks, goals, quality planning, quality control, software reliability and software reuse.</p> <p>Software maintenance: Introduction, corrective maintenance, adaptive maintenance, perfective maintenance.</p>	08
UNIT 5:	
<p>SOFTWARE PROJECT MANAGEMENT</p> <p>The management spectrum: The people- stake holders, team leaders, software teams, agile teams, the product- software scope, problem decomposition, the process- melding the product and process, process decomposition, The project, W⁵HH principle, critical practices.</p>	08

Text books:

1. Roger S. Pressman, "Software Engineering: A Practitioner's Approach", Alternate Edition, 7th Edition, McGraw Hill.
2. Ian Sommerville, "Software Engineering", 8th Edition, Pearson Publication Ltd 2012.

Reference books:

1. Software Engineering, an Engineering approach- James F. Peters.

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	8	12
Unit 2	10	12
Unit 3	8	22
Unit 4	8	22
Unit 5	8	10
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code: CA6223	Course Title: Artificial Intelligence and Applications.
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives

1. To impart artificial intelligence principles, techniques and its history.
2. To assess the applicability, strengths, and weaknesses of the basic knowledge representation, problem solving, and learning methods in solving real world problems.
3. To develop intelligent systems by assembling solutions to concrete computational problems.

Course Outcomes (COs):

1. Evaluate Artificial Intelligence (AI) methods and describe their foundations.
2. Analyze and illustrate how search algorithms play vital role in problem solving.
3. Demonstrate knowledge of reasoning and knowledge representation for solving real world problems.
4. Having analytical ability to apply learning techniques in solving real world problems.
5. Illustrate the construction of learning and expert system.

Content	Hours
Unit - 1	
Artificial Intelligence and its Issues Introduction: What is AI? , The foundations of AI, The history of AI , Applications of AI, Intelligent Agents: Agents and Environments, Good Behaviour ,The nature of environments, The structure of agents.	8
Unit-2	
Problem Solving Problem solving by Search, Well defined problems and solutions, Real world problems, Problem space - State space, Uninformed Search Strategies-BFS, DFS, Informed (Heuristic) Search Strategies-A* Search, Heuristic Functions, Local Search algorithms and Optimization Problems-Hill climbing Search, Constraint Satisfaction Problems, Means -ends Analysis, Game playing -The mini-max algorithm, Alpha-Beta Pruning.	10
Unit-3	
Knowledge Representation and Reasoning Logical systems, Knowledge Based systems, Propositional Logic-Resolution, Forward and Backward Chaining, Predicate Logic, First Order Logic-Syntax and Semantics of First Order Logic, Inference in First Order Logic-Unification, Forward Chaining, Backward Chaining, Resolution. Semantic Nets, Frames, Conceptual Dependency, Scripts. Ontological Representations and applications.	10
Unit-4	
Learning Systems Forms of Learning Types - Supervised, Unsupervised, Reinforcement Learning, Learning Decision Trees.	6
Unit-5	
Expert Systems Stages in the development of an Expert System - Probability based Expert Systems - Expert System Tools - Difficulties in Developing Expert Systems – Applications of Expert Systems.	8

Text Books:

1. Russell, S. and Norvig, P. 2015. Artificial Intelligence - A Modern Approach, 3rd edition, Prentice Hall.
2. Poole, D. and Mackworth, A. 2010. Artificial Intelligence: Foundations of Computational Agents, Cambridge University Press.

Reference Books

1. Rich ,E., Knight K and Shankar, B. Artificial Intelligence, 3rd edition, Tata Mc Graw Hill.
2. Luger, G.F. Artificial Intelligence -Structures and Strategies for Complex Problem Solving, 6th edition, Pearson.
3. Brachman, R. and Levesque, H. Knowledge Representation and Reasoning, Morgan Kaufmann.
4. Alpaydin, E. Introduction to Machine Learning. 2nd edition, MIT Press.
5. Sutton R.S. and Barto, A.G. Reinforcement Learning: An Introduction, MIT Press.
6. Padhy, N.P.. Artificial Intelligence and Intelligent Systems, Oxford University Press.

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	8	8
Unit 2	10	10
Unit 3	10	22
Unit 4	6	18
Unit 5	8	20
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) = 60		

Course Code: CA6323	Course Title: Internet Of Things
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs

Course Objectives

This course will enable students to

1. Assess the genesis and impact of IoT applications, architectures in real world.
2. Illustrate diverse methods of deploying smart objects and connect them to network and compare different protocols for IoT.
3. Implement the tools, Arduino and Raspberry Pi in the IoT environment.
4. Infer the role of Data Analytics in IoT

Course Outcomes (COs):

After studying this course, students will be able to

1. Interpret the impact and challenges posed by IoT networks leading to new architectural models.
2. Compare and contrast the deployment of smart objects and the sensor-based technologies to connect them to network and appraise the role of IoT protocols for efficient network communication.
3. Solve real time IoT problems using Arduino.
4. Articulate IoT solutions using raspberry pi
5. Understand the need for Data Analytics in IoT.

Content	Hours
Unit - 1	
Introduction and Concepts of IoT Introduction: What is IoT? Genesis of IoT – IoT WF Architecture – Simplified IoT Reference Architecture, IoT Challenges, Core IoT Functional Stack, IoT Data Management and Compute Stack IoT Applications : Home Automation – Smart Cities – Retail – Logistics – Agriculture – Industry.	8
Unit-2	
Smart Objects and Protocols in IoT Smart Objects: The “Things” in IoT, Sensors, Actuators, and Smart Objects, Sensor Networks, Connecting Smart Objects, Communications Criteria, IoT Access Technologies: IEEE 802.15.4 - LORAWAN - IP as Network Layer: 6LoWPAN Application Protocols for IoT : CoAP – MQTT – XMPP.	9
Unit-3	
IoT Platform Design Methodology and Arduino Programming using Simulator Introduction to IoT Platform Design Methodology Steps – Introduction to Arduino, Exploring Arduino Board , Arduino Family, Installing Arduino IDE- Arduino Shields - Fundamentals of Arduino Programming, Basics Structure of Sketch including setup() and Loop(), Accessing Pins from sketch for Input and Output Functions, Use of Serial Library to communicate with Arduino ,Examples Modules for Arduino – Programs to interface LED switch and Potentiometer-Interact with serial monitor of our computer screen- Interfacing Sensors to the Arduino. Self-Study: Implementing IoT Programs using TinkerCAD.	10
Unit-4	
Programming with Raspberry Pi and IoT Simulation Tools Introduction to Raspberry Pi , Exploring Raspberry Pi Learning Board , Raspberry Pi Operating Systems, Operating System Setups, Raspberry Pi, Formatting SD card, OS installation, First Boot, Raspberry Commands, Programming Raspberry Pi with python: GPIO ,IoT Tools: Chef ,Raspberry Pi Interfacing and Programming. Self Study: Implementing IoT Programs using PROTEUS	9
Unit-5	
Building Blocks for IoT Analytics and Applications of IoT Introduction IoT Analytics: IoT Data and Big Data - Challenges of IoT Analytics Applications – IoT Analytics Lifecycle and Techniques, IoT, Cloud and Big Data Integration for IoT Analytics: Cloud-based IoT Platform, Data Analytics for IoT , Development Tools for IoT Analytics Applications: VITAL Architecture for IoT Analytics Applications	6

Text Books:

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", 1stEdition, Pearson Education (Cisco Press Indian Reprint).2017 **(Unit 1 and Unit 2)**
2. Srinivasa K G, "Internet of Things", CENGAGE Learning India, 2017 **(Unit 3 and Unit 4)**
3. John Soldatos "Building Blocks for IoT Analytics Internet-of-Things Analytics", River Publishers, 2017.**(Unit 5)**

Reference Books

1. Vijay Madiseti and ArshdeepBahga, "Internet of Things (A Hands-on-Approach)", 1stEdition, VPT, 2014.
2. Simon Monk, "Programming the Raspberry Pi: Getting Started with Python", McGraw Hill Professional, 2012\
3. The official raspberry Pi Projects Book
https://www.raspberrypi.org/magpiissues/Projects_Book_v1.pdf
4. Eben Upton and Gareth Halfacree, "Raspberry Pi User Guide", August 2016, 4th edition, John Wiley & Sons \
5. Alex Bradbury and Ben Everard, "Learning Python with Raspberry Pi", Feb 2014, JohnWiley & Sons
6. Michael Margolis, "Arduino Cookbook", First Edition, March 2011, O'Reilly Media, Inc.

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	8	14
Unit 2	9	10
Unit 3	10	22
Unit 4	9	22
Unit 5	6	10
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) = 60		

Course Code: CADE6423	Course Title: Mobile Application
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 3 Hrs

Course Objectives:

1. To create robust mobile applications and learn how to integrate them with other services.
2. To design intuitive, reliable mobile apps using the android services and components.
3. Develop user interface and multimedia that works with different mobile screens.
4. Implementing SQLite for the back end data storage.

Course Outcomes:

After completing this course, student will be able to:

1. Build basic mobile applications with Android.
2. Understand both the basic and advanced concepts of Android.
3. Create java code and XML for making the app interactive and dynamic.
4. Implement and Deploy the application on Google Play
5. Analyze various services and database by creating simple projects.

Content	Hours
Unit - 1	
<p>Introduction to Android Introduction to Android, Application in Android, Android Operating System, Android versions, Features of Android, Android Architecture, Android Devices and History of Android.</p> <p>Android Apps, In-Built Applications, Prerequisites, Android Environment Setup, Downloading JDK, Setting Path of JDK, Installation of Android Studio in Mac, Linux and Windows, setting up ADT Plug In, Creating First Android Application, Introduction Android Studio Vs Eclipse.</p>	6
Unit-2	
<p>Fundamentals of Android Studio and Layouts Android Studio Fundamentals and Android Studio IDE, Viewing and adding Java Compilation and Runtime, Creating basic first Project, Android Components, Android Intent, Types of Intent, Android Services, Types of Android Services, Broadcast Receivers, Android Content Provider, Notification, Types of Views, Overview of Android Widgets. Layout, Understanding of Constraint and Linear Layout Understanding of Frame and Relative Layout, Image View, Demonstration of Animation (any 5)</p>	10
Unit-3	

<p>Classifying the Activities and UI Elements</p> <p>Starting and setting Activity, Android Lifecycle, Intent and its types, implementing the Intent.</p> <p>Passing data to Multiple Activity, Introduction to Themes and Colors, Working with Themes and Colors - Creating Project, Changing values of defined colors, Theme Colors, Styles, Menu, Fragments, Working with Fragments Activity, Introduction to Classes, Long Running Task, the Long Running Tasks.</p>	6
<p>Unit-4</p>	
<p>Shared Preference and Internal Storage</p> <p>Introduction to Shared Preference, Overview of Operating Mode, Introduction to Editor, Methods, Storing Data, Retrieving Data, Clear and Delete Working with Shared Preference - xml File, Main Activity</p> <p>Internal Storage, Difference between Shared Preference and Internal Storage, Read and Write data to Internal Storage, Syntax to Write a text file and Reading a text file, example of Internal Storage</p>	10
<p>Unit-5</p>	
<p>Advanced Android Features and Media Functions</p> <p>Controlling Audio, Audio Volume Seeking, Custom Action Bar, Customizing Action Bar, Alert Dialog, Adding Alert Dialog, List View, Timer, Showing and Hiding UI Element, Try and Catch Block.</p> <p>Implementing Map Activity, SQLite Database, Implementing and working with SQLite Database</p> <p>SQLite Advance, Basics of Kotlin, Difference between Kotlin and Java, Exploring Kotlin Language.</p>	10

Text Books:

1. Learn Java for Android Development (2nd edition), Author: Jeff Friesen, Publisher: Apress, Published Date: February 20th 2013.
2. Android Programming: The Big Nerd Ranch Guide, **Author:** Bill Phillips, Brian Hardy, Publisher: Big Nerd Ranch Guides, Published Date: April 07th 2013
3. Programming Android: Java Programming for the New Generation of Mobile , Devices (2nd Edition), Author: Zigurd Mednieks, Laird Dornin, G. Blake Meike, Masumi Nakamura, Publisher: O'Reilly Media, Published Date: October 19th 2012.

References:

1. Android Programming THE BIG NERD RANCH GUIDE BILL PHILLIPS & BRIAN HARDY
<https://ptgmedia.pearsoncmg.com/images/9780321804334/samplepages/0321804333.pdf>
2. SamsTeachYourself Android™ Application Development 800 East 96th Street, Indianapolis, Indiana, 46240 USA Lauren Darcey Shane Conder Second Edition.
<http://www.luciopanasci.it/Ebooks/Sams%20Teach%20Yourself%20Android%20Application%20Development%20in%2024%20Hours,%202nd%20Edition.pdf>

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	6	10
Unit 2	10	12
Unit 3	6	12
Unit 4	10	22
Unit 5	10	22
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) = 60		

Course Code: CA6P1	Course Title: Mobile Applications Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hrs

Course Objectives:

1. To develop interactive Mobile apps using Java code and XML.
2. To implement Layouts, views and media player in designing user friendly screen.
3. To Analyze the Project structure in Android Studio by adding various resources.
4. To apply animation and Graphic objects within the application.

Course Outcomes:

Students can acquire the following skills after completing the course:

1. Able to develop interactive and user friendly mobile apps.
2. Apply various resources in managing data and design.
3. Managing a SQLite database as back end and making application more advanced.
4. Able to develop multimedia components such as audio, video and animations.
5. Able to connect API for the external service as GPS.

Program List:

1. Create an android app to display various android life cycle phases' use both log cat and toast to display the state of an Activity. Let the Activity contain Text views and image view.
2. Create a login application to verify user name and password. On successful login, redirect to another activity that has a textview to display "welcome user" with logout button. On click of logout button, a dialog should appear with ok and cancel buttons. On click of oK button, navigate back to the login activity and on click of cancel button, stay on the same activity.
3. Create an android app with first activity having edit-text and send button. On click of send button, use explicit intent to send the text within edit-text to a second activity and displayed within text-view. Also include a validation where the edit-text can accept only the text.
4. Create a calculator app that performs addition, subtraction, division and multiplication operation on numbers. Use suitable background or images in an activity. Use validation for the edit text that can take only numerical values.

5. Create a spinner application with strings taken from resource directory res/values/strings.xml and on changing the spinner value, image has to change. Image to be saved in the drawable directory.
6. Create an app that uses radio button group which calculates discount on shopping bill amount. Use editText to enter bill amount and select one of three radio buttons to determine a discount for 20, 25, or 30 percent. The discount is calculated upon selection of one of the buttons and displayed in a textview control.
7. Use linear layout to create a simple application that will take the contents of a predefined textview and use a button to cause the application to take that text, convert it to uppercase, colour, font family and size and display it in an edittext field.
8. When working with edittext controls on the screen, create an application to respond to a particular keystroke rather than requiring the user to touch a button using key event. Use the Keystroke that can print a profile.
9. Create an application that uses table layout with textview, edittext and buttons. Also implement customized styles.xml file within res/values directory, to style text of textview control.
10. Create an application to perform the operations of create, insert, delete, view and update using SQLite database. The front end should include registration details.
11. Create an app to display 3 button controls vertically aligned. On selecting a button, the color of the screen will change.
12. Implement 3 buttons to start, pause and stop the audio and Video player. Use the raw folder to include the sources.
13. Develop an app that can show the connectivity status and boot Status as soon the app starts. Implement broad cast receiver and Set the manifest file.
14. Implement a current location map using external service through API and also show the requirement of manifest file with the permission.

Evaluation Scheme for Lab Examination

Assessment Criteria	Marks
Writing	8
Execution	10
Viva Voice	7
Total	25

Course Code: CA6P2	Course Title: Major Project Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hrs

Course Objective:

1. Identify real world problems and challenges that need IT based solutions.
2. Demonstrate an ability to work in teams.
3. Improve the team building, communication and management skills of the students.
4. Use all concepts of IT in creating a solution for a problem.
5. Report and present the findings of the study conducted in the preferred domain.

Course Outcomes:

Outcomes of the Major Project:

Project Analysis and Design will cover following

1. Problem definition
2. Analyze Systems requirements.
3. Plan the systems design phase of the SDLC.
4. Decide the s/w requirement specifications and h/w requirement specifications.
5. Perform and evaluate feasibility studies like cost-benefit analysis, technical feasibility, time feasibility and operational feasibility for the project.
6. Work on data collection methods for fact finding.
7. Construct and evaluate data dictionaries/ decision trees/ decision table.
8. Create and evaluate graphical tools as systems flow charts, entity-relationship(ER) diagrams and state transition diagrams.
9. Construct and evaluate data flow diagrams.

Project Development and Implementation will cover following

1. Perform coding for the project.
2. Design and evaluate system outputs.
3. Design and evaluate systems inputs.
4. Design and evaluate validity checks for input data.
5. Design and evaluate user interfaces.
6. Documentation of requirements
7. Prepare documentation of project
8. Perform various testing techniques/strategies.
9. Generate various reports in project.
10. Deploy the project on machine/lab/real time environment
11. Define the future scope and further enhancement of the system.

Project Guidelines

1) Understanding the significance of Project

Most of the students are under an impression that if good layout is proposed then it will help them in scoring good marks but the quality of project is analyzed by proper write-ups. It provides an opportunity for learners to demonstrate originality and to plan and organize the project work and put the practical approach of all the topics studied in the entire curriculum.

2) Meaning of Project

A project is a study of factual information for comprehending and applying the various concepts of the course into practice. Its main purpose is not to generalize but to study the situation with a practical orientation.

3) No Objection Certificate

If the project is carried out in a company or organization, then a certificate for no objection of same needs to be presented. It should mention that the organization has no objection in publishing the findings of the project study. The certificate should contain the name of authority with signature and company stamp and should be given on company's letterhead and duly signed by authorized signatory.

GUIDELINES FOR BCA PROJECT

- The project work should normally include software development.
- The students are expected to work on real-life project. However, it is not mandatory for a student to work on a real-life project.
- Not more than four students are permitted to work on a project.
- The project may be done in the university campus/concern study institute or in an approved sponsoring organization (industry/ research and development laboratories / educational institution / Software Company related to the proposed topic.
- The student can formulate a project problem with the help of her/his lab supervisors / teachers and if approved, the students commence working on it.
- A candidate is required to present the progress of the project work during the semester as per the schedule.

4) PROJECT SYNOPSIS FORMAT

The project proposal should be prepared and approved in consultation with supervisor. The project proposal should clearly state the project objectives and the environment of the proposed project to be undertaken. The project proposal should contain complete details in the following form:

1. Title of the project
 2. Introduction and objectives of the project
 3. Analysis (DFD, ER diagrams, class diagrams, time line etc. As per the project requirements).
 4. A complete structure which includes:
 - Name of modules and their description
 - Database / data structures description
 - Process logic of each module (flow chart)
 - Reports generation. (Report format)
 5. Tools / platform, hardware and software requirement specifications
 6. Organization/ Company Certificate
- 5) Technical Specifications of Project Report

Length:

The length of the report should be between 80- 100 pages including the cover page, summary, table of contents, list of figures, list of tables, and acknowledgement.

Script and Page Format:

The report should be typed using a Word Processor on standard A4 (210 mm x 297 mm) paper size. A conventional font, size 12-point for Normal text, 14 for headings, 16 for Chapter heading, and line spacing of 1.5 mm should be used.

Margins:

Left-hand margins should have a width of not less than 38 mm to facilitate binding. The right hand, the top, and the bottom should be 25 mm. Each page must be typed in one side, leaving a wide margin.

Paper and Print Quality Paper and print quality:

Paper and Print Quality Paper and print quality are important for successful legibility. The report can be printed on a standard quality paper, (e.g., photocopy paper)

Pagination:

Positioning of page numbers should be at the bottom of the pages. Pages starting from the summary until the last list of tables should be numbered using Latin numbers (I, II, III, IV, ...).

Pages starting from the Introduction until the appendices should be numbered using numbers (1,2, 3,...). Pages with figures and tables or illustrations must be also numbered.

Binding:

The report should be hard bound.

Number of copies: Two copies should be submitted – self copy, College copy (Group).

Project Analysis and Design will cover following

1. Problem definition
2. Analyze Systems requirements.
3. Plan the systems design phase of the SDLC.
4. Decide the s/w requirement specifications and h/w requirement specifications.
5. Perform and evaluate feasibility studies like cost-benefit analysis, technical, feasibility, time feasibility and operational feasibility for the project.
6. Work on data collection methods for fact finding.
7. Construct and evaluate data dictionaries/ decision trees/ decision table.
8. Create and evaluate graphical tools as systems flow charts, entity-relationship (ER) diagrams and state transition diagrams.
9. Construct and evaluate data flow diagrams.

Project Development and Implementation will cover following

1. Perform coding for the project.
2. Design and evaluate system outputs.
3. Design and evaluate systems inputs.
4. Design and evaluate validity checks for input data.
5. Design and evaluate user interfaces.
6. Documentation of requirements
7. Prepare documentation of project
8. Perform various testing techniques/strategies.
9. Generate various reports in project.
10. Deploy the project on machine/lab/real time environment
11. Define the future scope and further enhancement of the system.
12. Add Appendices (if any) to be placed in support with the project report documentation.

Project Evaluation

Presentation / demo must be carried out in all the lab sessions in the whole semester for internal assessment of the project. In each lab session a student is evaluated for 25 marks.

Presentation /Demo-15

Viva Voce-10

End Semester assessment must be carried out based on documentation, demo and Viva

Assessment Criteria	Marks
Presentation/Demo	10
Documentation	5
Viva Voice	10
Total	25

Lab Manual for 10 weeks

Weeks Descriptions

Week 1: Detail briefing of project, Group formation

Week 2: No objection certificate, Synopsis and timeline /Gantt Chart Submission

Week 3: Requirement Analysis

Week 4: System Design

Week 5: UI design for minimum 4 pages.

Week 6: Database Specification

Week 7: UI design and Coding

Week 8: Implementation, Testing and maintenance

Week 9: Demo of the Project

Week 10: Documentation for Supervisor Signature

Course Code: DAV01	Course Title: Power BI
Course Credits: 03	Hours/Week: 04
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 100	Exam Duration: 03 Hrs

Course Description

This course introduces Power BI, a business analytics service by Microsoft that enables users to visualize and analyse data from various sources. The course covers basic concepts of data analysis, data visualization, and data modeling in Power BI. Students will learn to create interactive reports, dashboards, and visualizations using Power BI Desktop.

Learning Outcomes

By the end of the course, students will be able to:

- Understand the basic concepts of data analysis, data visualization, and data modeling
- Navigate the Power BI Desktop interface.
- Import data from various sources into Power BI
- Create basic data models and relationships between tables.
- Use DAX (Data Analysis Expressions) to create calculated columns and measures.
- Create interactive reports and visualizations using various chart types, tables, and other visualization tools.
- Design and publish dashboards for stakeholders.
- Use Power BI service to share dashboards and reports with others.
- Understand the security and governance considerations in using Power BI.

Course Outcome

C01: Understanding of basic concepts of Power BI

C02: Creating and applying data models in Power BI

C03: Visualising datasets using Power BI

C04: Understanding and applying data modelling and visualisation techniques using the advanced features of Power BI

C05: Creating Dashboard using Power BI

C06: Learning security, Collaboration, and best practices of Power BI

Unit 1: Introduction to Power BI

- Introduction to Power BI and its features
- Power BI Desktop and Power BI Service
- Data sources and connectors

Unit 2: Data Modeling in Power BI

- Creating data models and relationships
- Introduction to DAX (Data Analysis Expressions)

Unit 3: Data Visualization in Power BI

- Introduction to data visualization
- Creating charts, tables, and other visualization tools
- Using filters and slicers

Unit 4: Advanced Data Modeling and Visualization

- Advanced data modeling techniques
- Advanced visualization techniques
- Custom visuals and importing from AppSource.

Unit 5: Creating Dashboards

- Introduction to dashboards
- Designing and creating dashboards in Power BI
- Sharing and collaborating on dashboards

Unit 6: Security and Governance in Power BI

- Security considerations in Power BI
- Sharing and collaboration in Power BI
- Best practices for governance and administration

REFERENCE BOOKS:

1. Mastering Microsoft Power BI By **Brett Powell**
2. Beginning Power BI by Dan Clark