ST JOSEPH'S UNIVERSITY

BENGALURU-27

DEPARTMENT OF COMPUTER SCIENCE AND COMPUTER APPLICATIONS

SYLLABUS FOR UNDERGRADUATE PROGRAMME BCA

For Batch 2024-2027

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 Instructions	Number of Hours of teaching per week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CA1124	Fundamentals of Computers	42	03	03	40	60	100
Theory	CA1224	Programming in C	42	03	03	40	60	100
Theory	CA1324	Mathematical Foundations	42	03	03	40	60	100
Theory	CA1424	Unix Programming	42	03	03	40	60	100
Practical	CA1P1	Information Technology Lab	28	02	1	25	25	50
Practical	CA1P2	C Programming Lab	28	02	1	25	25	50

Total Number of credits:22(Core papers (14), Languages(6), Foundation Course(2))

Semester 2	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	CA2124	Data Structures using C	42	03	03	40	60	100
Theory	CA2224	Object Oriented Concepts using JAVA	42	03	03	40	60	100
Theory	CA2324	Discrete Mathematical Structures	42	03	03	40	60	100
Theory	CA2424	Probability and statistics using R	42	03	03	40	60	100
Practical	CA2P1	Data Structures Lab	28	02	1	25	25	50
Practical	CA2P2	JAVA Lab	28	02	1	25	25	50
Total Number of credits:22(Core papers (14), Languages(6), Foundation Course(2))								

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	CA3125	Database Management Systems	42	03	03	40	60	100
Theory	CA3225	Web Development Technologies	42	03	03	40	60	100
Theory	CA3325	Artificial Intelligence	42	03	03	40	60	100
Theory	CA3425	Computer Organization and Architecture	42	03	03	40	60	100
Practical	CA3P1	DBMS Lab	28	02	1	25	25	50
Practical	CA3P2	Web Development Technologies Lab	28	02	1	25	25	50
Total Number of credits:22(Core papers (14), Languages(6), Open Elective(2))								

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	CA4125	Python Programming	42	03	03	40	60	100
Theory	CA4225	Computer Communication and Networks	42	03	03	40	60	100
Theory	CA4325	Operating System Concepts	42	03	03	40	60	100
Theory	CA4425	Software Engineering	42	03	03	40	60	100
Practical	CA4P1	Python programming Lab	28	02	1	25	25	50
Practical	CA4P2	Computer Networks lab	28	02	1	25	25	50
Total Number of credits:22(Core papers (14), Languages(6), Open Elective(2))								

EXAMINATION AND ASSESSMENTS

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: CA1124	**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", Addision-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: CA1P1	Course Title: Information technology and Unix 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 -

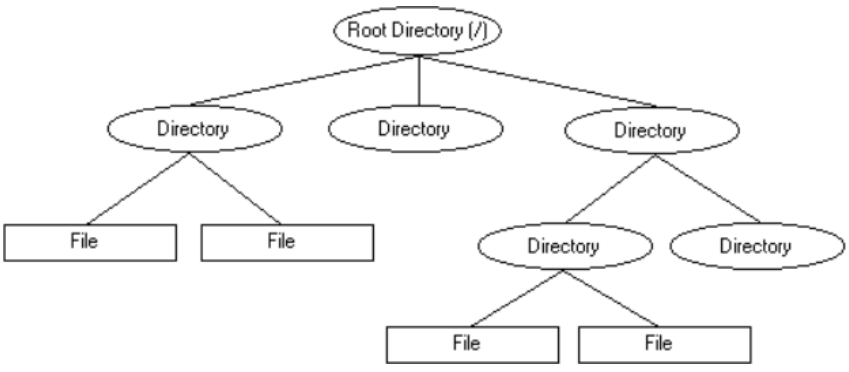
- To List various hardware components and its specifications.
- To recognize the hardware components of computers and describe its functions & to differentiate the role of hardware and software of the computer.
- To practice the setups of software and its installation process & to practice and compare all the tools of MS-Word, MS-Excel and MS-Powerpoint.
- To be familiar with Unix Operating System and its commands

Part A: Information Technology

1. Identification of the peripherals of a computer, components in a CPU and their functions.
2. Activities using Word Processor Software
3. Activities using Spreadsheets Software
4. Activities using Presentation Software
5. Activities involving Multimedia Editing (Images, Video, Audio …)
6. Flow charts: Installation and use of flow software for different arithmetic tasks like sum, average, product, difference, quotient and remainder of given numbers, calculate area of Shapes (Square, Rectangle, Circle and Triangle), arrays and recursion.

Part B: Unix Prgms

1. Implement Unix internal and external command list: man, who, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown.
2. Implement file and directory related command list: mkdir, rmdir cd, cp , more , cat, mv, rm ,diff,wc, etc command for following structure



3. Write a shell script program to check whether given input is even or odd.
4. Write a shell script program to find the largest of three numbers.
5. Write a script to compute gross salary of an employee according to the rule given below. If basic salary is < 10000 then HRA=15% of basic & DA=45% of basic If basic salary is >=10000 then HRA=20% of basic & DA=50% of basic.
6. Write a shell script that accept one or more filenames as argument and convert all of them to uppercase.
7. Implement regular expression commands for following basic regex: . , ^ , $, * , \, (), ?
8. Implement process based and filter based command in Unix

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: CA1224	**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	
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. **C Programming Basic Concepts:** C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants. **Input and output with C:** Formatted I/O functions - *printf* and *scanf,* control stings and escape sequences, output specifications with *printf* functions; Unformatted I/O functions to read and display single character and a string - *getchar, putchar, gets* and *puts* functions.	10
Unit – 2	

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. **Control Structures:** Decision making Statements - S*imple if, if_else*, *nested if_else, else_if ladder, Switch Case*, *goto, break* & *continue* statements; Looping	12

Statements - Entry controlled and exit controlled statements, *while, do-while,for* loops, Nested loops.	
Unit – 3	
Derived data types in C: Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays -Declaration, Initialization and Memory representation. **Strings:** Declaring & Initializing string variables; String handling functions - *strlen, strcmp, strcpy and strcat;* Character handling functions - *toascii, toupper, tolower, isalpha, isnumeric* etc.	8
Unit – 4	
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. 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.	6
Unit – 5	
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; **File Operations :** File modes, File functions, and File operations, Text and Binary files, Command Line arguments.	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		

PROGRAMMING in C
List of problems for LAB exercises

1.	Write a program to convert the temperature in Celsius scale to Fahrenheit scale using the formula: **F = 9/5C + 32**
2.	Write a program in C to calculate the area of a triangle given the lengths of sides. **Area = $\sqrt{}$(S(S-a)(S-b)(S-c) where S = (a+b+c)/2**
3.	Given three numbers find the maximum number (without using an array)
4.	Given the radius of a circle calculate the a) Circumference and b) Area of the circle
Using Looping statements (ARRAYS)	
5.	Write a program in C to input some numbers into an array and find the average of some numbers.
6	Write a program in C using an array to input some numbers and find the STANDARD DEVIATION of those numbers. **Standard Deviation =$\sqrt{}$($\sum X_i^2 /N - (\sum X_i/N)^2$)**
7	Write a program in C using an array to INPUT some numbers into the array and find the **MAXIMUM NUMBER** and it's location.
8.	Write a program in C to input some numbers into an array and SORT them using **BUBBLE SORT** TECHNIQUE
9.	Write a program in C to input some numbers into an array and do **LINEAR SEARCH** to search for a given number. Let the program give the location of the number in the array if it exits or else indicate that the number is not present in the list
10.	Write a program in C to input some numbers into an array and do **BINARY SEARCH** to search for a given number. Let the program give the location of the number in the array if it exits or else indicate that the number is not present in the list
11.	Write a program in C to find the factorial of a given number.
12.	Write a programs in C to find the sum the sum of the following series. (USE **WHILE LOOP**) a) S = 1 + 2 + 3 +.....N terms b) S= 1 + 3 + 5 + … N terms c) S = 2 + 4 + 6 +…. N terms d) S = 1! + 2! + 3! + …N terms e) S = 1! -3! +5!-7!+… N terms f) Cos(x) = $1 - x^2/2! + x^4/4! - x^6/6! + \ldots$ as long as the value of the terms generated is more than equal to 0.0001. Where x = Theta*Pi/180. Theta is angle in degrees. g) Sin(x) = $x - x^3/3! + x^5/5! - x^7/7! + \ldots$ as long as the value of the terms generated is more than equal to 0.0001. Where **x = Theta*Pi/180**. Theta is angle in degrees
13.	Write a program in C to find the value of **^nC_r= n!/(r! * (n-r)!)** Using only one loop.
Conditional statements	
14.	Enter Employee ID, name and Basic. Calculate DA, HRA, IT, **Gross Pay and Net Pay**. Gross Pay = Basic + DA + HRA ,Net Pay = GrossPay- Tax **TAX** is calculated depending on different slabs of BASIC pay. (Rs 500 If Basic is below Rs 10,000, 8% if Basic is grater than equal to 10,000 but below 25,000; and 10% if it is 25,000 and above) **DA** is 20% of Basic and **HRA** is 10% of Basic.
15.	Write a program in C to input REgNo, Name and marks in three subjects. Print marks card and indicate whether the candidate has passed or failed and if passed indicate whether he/she gets **DISTINCTION, FIRST CLASS, SECOND CLASS or THIRD CLASS(** Criterion for passing: The candidate should have 35 and above in each subject and 40% average and should have 75% and above for DISTINCTION; 60 - 75% : FIRST CLASS: 50-60%: SECOND CLASS; 40-50%; THIRD CLASS)
MATRICES	
16.	Write program in C ti INPUT, PRINT and find the **SUM of TWO MATRICES.**
17.	Write program in C ti INPUT, PRINT and find the **PRODUCT of TWO MATRICES.**

18.	Write a program in C to find the maximum number in an array and it's location.
FUNCTION SUB PROGRAMS (use them in the main program)	
19.	.To convert temp in **Celsius scale to Fahrenheit** 2. To return the **area of a circle** given the Radius. 3. To return the area of a circle given the lengths of 3 sides. 3. To return the area of a rectangle. 4. To return the **average** of N numbers in an array 5. To return the **maximum** number in an array 6. to Input, Print and return the **average of N** numbers in an array (3 functions) 7. To return **the STANDARD DEVIATION** of N numbers in an array. 8. To input data into a MxN matrix. To print the content of a MxN matrix, to find the **sum of two matrices**. 9. To search for a given number using **LINEAR SEARCH** method. Return 1 if it is present or else return -1. 10. To Sort N numbers in an array using **BUBBLE SORT** technique. 11. To return the Cosine of an angle in degrees. 12. to return the length of a string **int StrLen(char S[])** 13. to make a duplicate of a string **void StrCpy(char T[], char S[])** 14. to join two strings. **void StrCat(char T[], char S[]**) 15. to return the cosine of an angle in degrees. **float CosX(float Theta)** 16. to return the sine of an angle in degrees. **float SinX(float Theta)**
FILE PROGRAMS:	
20.	Write a program in C to open a file and write some numbers into the file.
21.	Write a program in C to open a file and read the content of the file and display the content.
22.	Write a command line program to display the content of a text file.
23.	Write a command line program to make a duplicate of a file
POINTERS	
24	a) Write a function sub program in C with pointers to return the length of a string. (**int StrLen(char *S)**) b) Write the pointer version of the function to make a duplicate of a string (**void StrCpy(char *T, char *T)**) c) Write the pointer version of the function to join two strings. (**void StrCat(char *T, char *S**) d) Write a function sub program to **swap two numbers void Swap(int *A, int *B)**

Course Code: CA1324	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-Eigenvalues	6
Unit – 5	
Differential calculus: Functions and limits - Simple Differentiation ofAlgebraic 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, NewDelhi
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		

Course Code: CA1424	Course Title: Unix Programming
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 45	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 02Hrs

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:

- Learn about 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.

Content	
Unit – 1	
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.	**(9)**
UNIT II	
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 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.	**(9)**
UNIT III	
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.	**(8)**

UNIT IV	
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.	**(10)**
UNIT V	
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, system calls. **Self Study:** Comparison of different operating systems (UNIX ,ANDROID,IOS)	**(9)**

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: CA2124	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	
Introduction to data structures: Definition; Types of data structures - Primitive & Non-primitive, Linear and Non-linear; Operations on data structures. Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - *malloc*, *calloc*, *realloc* and *free.* Algorithm Specification, Performance Analysis, Performance Measurement-Asymptotic notations.	6
Unit – 2	

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; Representation of multidimensional arrays; Sparse matrices.	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 liked list, Header liked 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.

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 a simple queue.
9. Program to implement linear linked list.
10. Program to display traversal of a tree.

Course Code: CA2224	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	10

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.	
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 numbers1 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 inan 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 factorialvalue, 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 causedwhen 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: CA2324	**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 principles 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 ofInclusion-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		

Course Code: CA2424	**Course Title: Probability and Statistics using R**
Course Credits: 03	**Hours/Week: 03**
Total Contact Hours: 45	**Formative Assessment Marks: 40**
Exam Marks: 60	**Exam Duration: 02Hrs**

Course Objectives:

1. To introduce the fundamental concepts of statistics and probability.
2. To develop skills in data analysis and interpretation.
3. To apply statistical methods to real-world problems.
4. To understand the role of probability in decision making.
5. To provide an introduction to R programming for statistical analysis.

Course Outcomes (COs):

At the end of this course, the student will be able to:

CO1	Knowledge	Recall fundamental concepts of statistics and probability
CO2	Understand	Explain measures of central tendency and dispersion
CO3	Application	Use probability theory to solve real-world problems
CO4	Analysis	Analyze data using graphical representations and summary statistics
CO5	Synthesis	Synthesize data visualization techniques in R to create insightful and visually compelling representations of statistical data, enhancing the clarity and impact of the finding
CO6	Evaluation	Assess the results of hypothesis testing and correlation analysis

Content	**Hours**

Unit 1	
Introduction to Statistics Definition and scope of statistics,Types of data: Qualitative and Quantitative, Data collection methods, Classification and tabulation of data, Graphical representation of data: Bar charts, Histograms, Pie charts, Measures of central tendency: Mean, Median, Mode, Measures of dispersion: Range, Variance, Standard Deviation	**10**
Unit 2	
Sampling and Estimation Concepts of population and sample, Sampling methods: Random sampling, Stratified sampling, Cluster sampling, Sampling distributions, Central Limit Theorem, Point estimation and interval estimation, Confidence intervals for mean, proportion, and variance, Practical exercises on sampling methods, Calculation and interpretation of confidence intervals	**8**
Unit 3	
Probability Theory Introduction to probability and its importance, Sample space and events, Classical definition of probability, Axioms of probability, Conditional probability, Bayes' theorem, Independent and dependent events, Problem-solving sessions on probability calculations, Real-life applications of probability	**8**
Unit 4	
Random Variables and Probability Distributions Definition of random variables: Discrete and Continuous, Probability distribution functions, Expected value and variance of random variables, Common probability distributions:· Discrete distributions: Binomial, Poisson, · Continuous distributions: Normal, Exponential, Calculation of expected value and variance, Exploration of probability distributions with real-world examples	**8**
Unit 5	

Introduction to R Programming Introduction to R and RStudio, Basic R syntax and data structures, Importing and exporting data in R, Data manipulation using dplyr and tidyr, Basic statistical analysis in R: Descriptive statistics, Probability distributions, Data visualization using ggplot2, Introduction to statistical modeling in R, Hands-on sessions in RStudio, Practical exercises on data manipulation and analysis, Creating and interpreting visualizations using ggplot2	**8**

REFERENCES:

1. "Statistics for Business and Economics" by Paul Newbold, William L. Carlson, Betty Thorne
2. "Probability and Statistics for Engineers and Scientists" by Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying E. Ye
3. "Introduction to the Practice of Statistics" by David S. Moore, George P. McCabe, Bruce A. Craig
4. "Probability and Statistics" by Morris H. DeGroot, Mark J. Schervish
5. "R for Data Science" by Hadley Wickham and Garrett Grolemund
6. "The Art of R Programming" by Norman Matloff

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**

Course Code: CA3125	**Course Title: Database Management Systems**
Course Credits: 3	**Hours/Week: 3**
Total Contact Hours: 42	**Formative Assessment Marks: 40**
Exam Marks: 60	**Exam Duration: 02 Hrs**

Course Objectives:

1. To enable the students to learn the designing of database systems, foundation on the relational model of data and normal forms.
2. To understand the concepts of database management system, design simple Database model
3. To learn and understand to write queries using SQL, PL/SQL.
4. To enable the students to learn the designing of database systems, foundation on the relational model of data and normal forms.
5. To understand the concepts of database management system, design simple Database models

CO No.	Bloom's Taxonomy Level	Course Outcome
1	Knowledge	Define and describe the basic concepts of database systems, data models, and database architecture
2	Comprehension	Explain relational database design concepts, including normalization and integrity constraints.
3	Application	Apply SQL commands to create, manipulate, and query database systems.
4	Analysis	Analyze complex queries and database designs to improve efficiency and performance.
5	Evaluation	Compare and evaluate different database models and transaction management techniques
6	Synthesis	Design and construct efficient database systems using indexing, stored procedures, and triggers

CONTENT	Hours
UNIT-1	
Database Concepts: Database and Database Users: Introduction, Characteristics of the Database, Database Users, Advantages of Using DBMS, 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.	7
UNIT-2	
Data Models: 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. Vector DB: Introduction to Vector Databases,Fundamentals of Vector Search,Popular Vector Databases and Tools.	7
UNIT-3	
RDBMS AND Normalization: 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, trivial functional dependency, normal forms – 1NF, 2NF, 3NF, BCNF.	8
UNIT-4	

SQL and PL/SQL: **Structured Query Language**: SQL Data Definition and Data Types, Basic Queries in SQLCREATE 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. **Relational SET Operators**: UNION – UNION ALL – INTERSECTS - MINUS.SQL Join Operators: Cross Join – Natural Join – Join USING Clause – JOIN ON Clause – Outer Join. PL/SQL: introduction to Cursors and Exceptions	12
UNIT-5	
Transaction Processing: Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, Characterizing schedules based on recoverability, Characterizing schedules based on Serializability, Transaction support in SQL. **Concurrency Control in Databases:** Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Multiversion Concurrency control techniques, Validation Concurrency control techniques, Granularity of Data items and Multiple Granularity Locking.	**8**
Self Study: NoSQL Databases(MongoDB,Cassandra,CouchDB etc.	

Text Books

1. Abraham Silberschatz, Henry F.Korth and S.Sudarshan,"Database System Concepts", McGraw Hill International Publication ,VI Edition
2. Ramez Elmasri and Shamkant B. Navathe. "Fundamentals of Database Systems ", Pearson publication, VII Edition.
3. Raghu Ramakrishnan and Johannes Gehrke, "Database Management Systems", McGraw-Hill Education, 3rd Edition.

Reference book:

1. Jeffrey A. Hoffer, Ramesh Venkataraman, and Heikki Topi, “Modern Database Management”, Pearson Publication, 13th Edition.
2. Coronel, Morris, Rob, "Database Systems, Design, Implementation and Management", Ninth Edition
3. Nilesh Shah, "Database Systems Using Oracle", 2nd edition, Pearson Education India, 2016
4. Shio Kumar Singh , “Database Systems “,Pearson publications ,II Edition
5. P.Rizwan Ahmed, RDBMS, Margham Publications, 2016 Web Resources

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	7	15
Unit 2	7	18
Unit 3	8	15
Unit 4	12	15
Unit 5	8	15
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) = 60		

Course Code:CA3125P	**Course Title: DATABASE LAB**
Course Credits:02	**Hours/Week:04**
Total Contact Hours:52	**Formative Assessment Marks:25**
ExamMarks: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.

i) Populate the table STATS with some statistics for January and July.
ii) Write a SQL query to look at table STATS in undefined order.
iii) Write a SQL query to look at table STATS, picking up location information by joining with table STATION on the ID column.
iv) 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.

i) 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
ii) Write a SQL query to show MAX and MIN temperatures as well as average rainfall for each station

iii) 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 a 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.

i) Create a view (derived table or persistent query) to convert Fahrenheit toCelsius and inches to centimeters

ii) Add new column rainfall_centimeter to the table STATS.

iii) Insert values into rainfall_centimeter from the view

iv) 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.

i) 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.

i) Update all rows of table STATS to compensate for faulty rain gauges knownto read 0.01 inches low

ii) Update one row, ID 44‟s July temperature reading, to correct a data entryerror

iii) Make the above changes permanent

iv) 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.

i) Delete data from STATION table where longitude is >90

ii) Delete July data from STATS table where longitude is >90

iii) Increase the size for the column CITY with the following information:-

COLUMNNAME	DATATYPE(SIZE)
CITY	VARCHAR (25)

iv) 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.

i) Find instructors whose salary is more than the salary of any employee from department „Physics".

ii) Find the instructor name and department name of all instructors working in a department with any instructor whose name contains the letter "S".

iii) Find the name and department name, of instructors whose salary is more than all their colleagues' salaries in the same department.

11. Create table INSTRUCTOR with the following fields: InstuctID, InstructName, Department, Salary.

Create another table called STUDENT with the following fields:

StudentID, StudentName, Department, InstuctID

i) Find the names of all instructors whose salary is greater than at least one instructor in the Finance department.

ii) 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.

iii) Find the student name and department name of all student who study in a department with any student whose name contains the letter "S".

12. 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.

ii) Write SQL query to Retrieve details of all books in the library:
id, title, name of publisher, authors, number of copies in each branch.

iii) Write an SQL query to compute the total number of books based on publisher name order by year.

13. 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: CA3225	Course Title: Web Development 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:**	9

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	
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.	9
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, 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 **Working with Forms**	**9**

Controlled components-Uncontrolled Components **Database Connection:** Establishing connection, selecting database, Query operations Using MySQL. **Open APIs** Introduction to APIs, Understanding Open API Specification (OAS), Creating an OpenAPI Definition, Using Open APIs in Applications **Object-Relational Mapping** Introduction to Object-Relational Mapping (ORM) frameworks **Case study:** Connect front-end with any data bases.	

Text Books:

1. HTML, CSS, and JavaScript All in One, Sams Teach Yourself, 3/e by Julie C. Meloni and Jennifer Kyrnin | 8 May 2020

2. Learning React: Modern Patterns for Developing React Apps, Second Edition,2020

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: CA3P1	Course Title: Web Development 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 be 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 color 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: **CA3324**	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 concept of NLP 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	10

mini-max algorithm, Alpha-Beta Pruning.	
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.	10
Unit-4	
Learning Systems Forms of Learning Types – Supervised, Unsupervised, Reinforcement Learning, Learning Decision Trees.	4
Unit-5	
Introduction to Natural Language Processing and Expert Systems Natural Language Processing (NLP),Introduction to NLP and Its Importance, Text Preprocessing: Tokenization, Stemming, Lemmatization, Stopword Removal. Applications: Sentiment Analysis, Chatbots, Translation Systems. **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.	10

Text Books:

1. Poole, D. and Mackworth, A. T Artificial Intelligence: Foundations of Computational Agents,Third Edition, 2023,Cambridge University

2. Russell, S. and Norvig, Artificial Intelligence: A Modern Approach fourth edition 2020

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	10	22
Unit 2	10	22
Unit 3	8	12
Unit 4	8	12
Unit 5	8	10
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) = 60		

Course Code: CA3425	**Course Title: Computer organization and Architecture**
Course Credits: 03	**Hours/Week: 03**
Total Contact Hours: 42	**Formative Assessment Marks: 40**
Exam Marks: 60	**Exam Duration: 2Hrs**

Course Objectives:

1. Understand the fundamental principles of digital logic design
2. Learn to design and analyze combinational and sequential circuits
3. Gain knowledge of computer organization and architecture
4. Understand central processor organization and its components
5. Develop an understanding of input-output systems and memory organization

Course Outcomes:

Here the students will be able to:

1. Design and optimize combinational and sequential circuits
2. Apply Boolean algebra techniques and Karnaugh maps for circuit simplification
3. Understand and explain the architecture of a computer system
4. Analyze and design central processing units (CPUs)
5. Design and evaluate input-output systems and memory architectures

CONTENT	**Hours**
UNIT-1	

INTRODUCTION TO DIGITAL SYSTEMS AND COMPUTER ARCHITECTURE Functional Units of a Computer, bus structure, Von Neumann Architecture,Logic Gates- AND, OR, NOT,NAND, NOR, EXOR, truth tables, Boolean expressions and their simplifications, SOP & POS-min-term, max-term & Karnaugh map . Self-Study: Practical applications of logic circuits in modern computing systems.	**7**

UNIT-2	
COMBINATIONAL AND SEQUENTIAL CIRCUITS : Half Adder, Full Adder, Multiplexers, De-multiplexers, Encoders, Decoders Flip flops- RS, JK, D, T, binary Counters, shift registers, memory unit-random-access memories (RAM), read-only memories (ROM). **Self-Study**: Implementation in real-world digital systems and Timing dependencies and clock signals in sequential circuits.	**7**
UNIT-3	
BASIC COMPUTER ORGANISATION AND DESIGN: Instruction codes, computer register, computer instruction, timing and control, instruction cycle, memory reference instructions, input output and interrupt, design of basic computer. **Self-Study:** Examples of instruction flow in simple operations and Concepts of memory addressing and data retrieval methods.	**10**
UNIT-4	
CENTRAL PROCESSOR ORGANIZATION: Introduction, general register organisation, stack organisation, instruction formats, addressing modes, data transfer and manipulation instructions, shift instructions, program control ,RISC and CISC. **Self-Study:** Use of Microinstructions in Modern Processors and Hazards (Data, Control, Structural) and Their Resolution	**10**
UNIT-5	

INPUT-OUTPUT ORGANIZATION AND MEMORY ORGANIZATION: Peripheral devices, I/O interface, asynchronous data transfer, direct memory access (DMA). Interrupts and its types, input-output processor (IOP). Auxiliary memory, microcomputer, memory hierarchy, associative memory, virtual memory, cache memory, memory management hardware. **Self-Study:** DMA in graphics processing units (GPUs) for high-speed data transfers in gaming and video rendering.	**8**

Textbook:

"Computer Systems Architecture" by M. Morris Mano and Charles Kime
Publisher: Pearson Edition: 3rd and 5th Revised Edition

Reference Books:

1. "Computer Organization" by Carl Hamacher, Zvonko Vranesic, and Safwat Zaky
   Publisher: McGraw-Hill Edition: 5th Edition (2002)
2. "Structured Computer Organization" by Andrew S. Tanenbaum
   Publisher: Pearson Edition: 6th Edition (2012)
3. "Modern Computer Architecture and Organization" by Jim Ledin
   Publisher: Wiley Edition: 1st Edition (2020)
4. "Computer Organization and Architecture: Designing for Performance" by William Stallings
   Publisher: Pearson Edition: 10th Edition (2016)
5. "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy Publisher: Morgan Kaufmann Edition: 5th Edition (2013)
6. Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson
   Publisher: Morgan Kaufmann Edition: 6th Edition (2019)

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	7	15
Unit 2	7	18
Unit 3	10	15
Unit 4	10	15
Unit 5	8	15
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) = 60		

SEMESTER IV

Course Code: CA4125	**Course Title:** Python Programming
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2Hrs

Course Objectives:

1. Gain knowledge of slicing , positive and negative indexing
2. Understanding Collection and Implementation of control structures
3. Learn to organizing code using module and package for better readability and maintenance.
4. Understand core concepts of OOPS and regex
5. Handle errors and exceptions gracefully to build robust programs and integrate with real world scenario.

Course Outcomes:

Here the students will be able to:

1. Get the knowledge in solving complex problems with optimized solutions.
2. Understand Python data structures - lists, tuples, dictionaries to represent complex data.
3. Understand the concept of object-oriented programming paradigm and its applications.
4. Integrate and connect various real time projects in achieving multi-disciplinary fields.
5. Understand possible error-handling constructs for unforeseen inputs

CONTENT	**Hours**
UNIT-1	

Basics of Python: 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. Understanding in-built string methods programming examples for each concepts.	8

UNIT-2	
Collections and Control statements **Lists** : Create, Access, slicing Negative indexing ,list methods, **Tuple :** Create, Access, slicing Negative indexing ,tuple methods, **Set** : creation and operations , **Dictionary :** Create add and replace values, operation on dictionaries Programming using list and dictionary in-built functions. Conditional blocks & Flow Control 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.	7
UNIT-3	
Functions, Packages and File handling : **Functions:** parameters and arguments, positional arguments, keyword arguments and variable length arguments, local and global scope of variables, recursive functions lambda function **Packages :** Organizing python codes using functions, organizing python projects into modules, importing own module as well as external modules, Understanding Packages , importing packages . **File handling :** Reading files in python, writing files in python, Understanding read and write functions, manipulating file pointer using seek, Programming using file operations.	10
UNIT-4	

Object Oriented Programming and Regular Expression **Object Oriented Programming :** Concept of class, object and instances, constructor, class attributes and destructors, Inheritance, overlapping and overloading operators, Adding and retrieving dynamic attributes of classes. **Regular Expressions :** Pattern matching and searching, Power of pattern searching using regex in python, Pattern finding programs using regular expression.	8
UNIT-5	
Exception Handling and Python for Data Analysis: Avoiding code break using exception handling, Safe guarding file operation using exception handling, Handling and helping developer with error code. Introduction to NumPy,NumPy Arrays and Operations, Mathematical and Statistical Functions in NumPy, Introduction to Pandas, Data Handling with Pandas Data Manipulation in Pandas Data Analysis and Visualization with Pandas	8

TEXT BOOKS:

1. Allen B. Downey, "Think Python : How to Think like a Computer Scientist", 3rd Edition, O'Reilly Publishers, 2024.
2. Karl Beecher, "Computational Thinking: A Beginners Guide to Problem Solving and programming", 1st Edition, BCS Learning & Development Limited, 2017.
3. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython by Wes McKinney , Edition: 2nd Edition (October 24, 2017) Publisher: O'Reilly Media
4.Python Data Analytics: With Pandas, NumPy, and Matplotlib by Fabio Nelli 2nd Edition (2023) publisher Apress

REFERENCES:

1. "Python Programming: A Complete Guide for Beginners to Master, Python Programming Language" by Brian Draper 2016
2. "Python Crash Course" by Eric Matthes is a comprehensive introduction to Python programming, 3rd **Edition,** published **2022** by No Starch Press.
3. "Python Programming for Beginners: Python Programming Language Tutorial" by Joseph Joyner 2014

BLUEPRINT

Chapter	**No. of hours**	**Total marks for which questions are to be asked (Including bonus questions)**
Unit 1	8	15
Unit 2	7	18
Unit 3	10	15
Unit 4	9	15
Unit 5	8	15
TOTAL	42	78
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: 2Hrs

Objectives:

The student should be made

1. Engage in practical coding exercises to emphasize theoretical knowledge.
2. Enhance problem-solving skills by integrating multiple concepts
3. Implement classes and objects in lab exercises to solidify the concepts of OOP.

Course Outcomes:

Here the students will be able to

1. To Learn the ability to use Python's built-in data structures
2. To develop modular code by ensuring readability and reusability in their programs.
3. To design and implement classes and objects by applying basic OOPs concepts
4. To Learn debugging techniques to identify and resolve errors.
5. Ability to integrate multiple concepts.

LIST OF EXPERIMENTS

Basics of Python

Lists

Program 1: Write a Python program to create a list of n numbers entered by the user and find the largest and smallest numbers in the list.

Program 2 : Write a Python program to count the number of occurrences of an element in a list.

Tuples and Dictionaries

Program 3 : Write a Python program to create a tuple with at least 5 elements of your choice. Implement the following functionalities:

a) Print all the elements of the tuple using a loop.

b) Allow the user to input a value and check if it exists in the tuple. If it exists, display the index of the value; otherwise, print a message saying the value is not found.

c) Display the total number of elements in the tuple.

Program 4 : Write a Python program to create a Dictionary-based Phonebook to store names as keys and phone numbers as values and Implement functionalities to add, search, and delete contacts from the phonebook using dictionary operations.

Strings

Program 5: Write a Python program to count the frequency of each character in a string.
Program 6: Write a Python program to remove duplicate elements from a list of strings without using built-in methods.
Program 7: Write a Python program to reverse a string using slicing and loop methods.

Data Structure

Program8: Write a Python program to implement a stack.
Program 9 : Write a Python program to implement a queue.

Functions

Program 10: Write a Python module named currency_converter that includes functions to convert amounts between USD, EUR, and INR using predefined exchange rates.
Program 11: Write a Python program to demonstrate the use of default and keyword arguments.

Modules and Packages

Program 13 : Write a Python program to create a module with mathematical operations . Import and use the module in a separate script.
Program 14 : Create a package with two modules:

a) To convert celsius_to_fahrenheit(celsius)

b) To convert fahrenheit_to_celsius(fahrenheit)

File Handling

Program 15: Write a Python program to create a text file, write data into it, use the seek and append methods .

Program 16: Write a Python program to write a list of strings into a file and then read it back.

Object-Oriented Programming (OOP)

Class and Objects

Program 17: Write a Python program to create a Book class, which will represent a book in the library , create objects of that class to store information about different books.

Program 18 : Write a program to define a class Car with a constructor that initializes make, model, year, and

price. Add a method to display car details. Then, write a program to create a list of cars and print details of all cars whose price is greater than a given value.

Inheritance

Program 19 : Design a simple E-commerce system using Python. You will create a base class called Product and a derived class called ElectronicsProduct that inherits from Product. The base class will store general information about a product, while the derived class will add extra details specific to electronic products.

Program 20 : Create a Python program to simulate a vehicle management system with a base class Vehicle. Then create derived classes Car, Motorcycle, and Truck that inherit from Vehicle. The Vehicle class should have basic details like speed and fuel, while each derived class should have its own unique methods such as cargo_capacity for Truck and wheel_type for Motorcycle.

Method overloading and Operator overloading

Program 21: Create a Python program to overload a method that can perform mathematical operations on varying numbers of arguments.

Program 22 : Create a Complex class that supports basic complex operations. Overload the + and - operators to allow vector objects to be added or subtracted

Regular expression

Program 23 : Write a Python program to validate a password entered by the user. The program should ensure the password meets the following criteria:

- The password must be at least **8 characters long**.
- The password must contain at least **one uppercase letter**.
- The password must include at least **one digit**.
- The password must have at least **one special character**

Program 24 : Validate an email address format and handle invalid input.

Exception Handling

Program 25 : Design and implement a Python program that simulates a simple banking system. The program should allow a user to check their balance, deposit money, and withdraw money.

Use exception handling to manage the following scenarios:

1. Insufficient Funds: Raise a custom exception InsufficientFundsError if the user tries to withdraw an amount greater than their current balance.

2. Invalid Input: Handle errors when the user enters invalid data (e.g., entering text instead of numbers for deposit or withdrawal amounts).

Program 26 : Write a Python program that prompt the user to enter the name of a file. The program should try to open and read the file. If the file does not exist, it should catch the FileNotFoundError and display a message: "File not found"

Python for Data Analysis

Program 27 : Write a Python program to perform Basic Arithmetic Operations on two NumPy arrays and Applying Universal Functions in NumPy

Program 28 : Write a Python program to **plot line graphs**, **histograms**, **bar charts**, and **scatter plots**

using the plot() function in Pandas.

Evaluation Scheme for Lab Examination

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

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: CA4225	Course Title: Computer Communication and Networks
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2Hrs

Course Objectives:

1. To build an understanding of the fundamental concepts of computer networking.
2. Familiarize the student with the basic taxonomy and terminology of the computer networking area.
3. Introduce the student to network security concepts
4. Preparing the student for entry in advanced courses in computer networking.

Course Outcomes:

Here the students will be able to

1. Understand the concepts of Data Communication.
2. Study the functions of OSI Layers.
3. Familiarize with the Transmission Media, Flow Control and Error Detection and Correction.
4. Understand fundamental concepts in Routing, Addressing & working of Transport Protocols.
5. Gain familiarity with common networking & Application Protocols
6. Understand the formats of Wired and Wireless LANs.

CONTENT	Hours
UNIT-1	

INTRODUCTION Communication Network and services, Approaches to Network Design, Network Functions, Key factors in Communication Network, Network Topology, Applications of networks, Line configuration, Layered Architecture and Applications – Examples of Layering, OSI Reference Model, TCP/IP , Telnet, FTP and IP Utilities Self-study: Digital representation of data	7

UNIT-2	
TRANSMISSION SYSTEMS: Digital transmission Systems – Twisted Pair, Coaxial Cable, Optical Fiber, Multiplexing – frequency – Division , Time – Division , Wavelength Division Multiplexing, Switching- Circuit , Packet and Message. Hamming distance and coding, Checksum, Error Correction and detection, Cyclic redundancy check. **Self-study**: Unguided media - Radio wave, Microwave, Infrared Transmission systems.	8
UNIT-3	
PEER –TO-PEER PROTOCOLS: Peer-to peer Protocols and service models, Data Link Control, ARQ Protocols stop and wait, Go–back-N, Selective Repeat, Transmission efficiency of ARQ Protocols, Other adaptation functions, - Sliding window flow control Timing Recovery in Synchronous Services, Reliable Stream Service.	10
UNIT-4	
LOCAL AREA NETWORKS AND MEDIUM ACCESS CONTROL PROTOCOLS: Multiple access communications: Local Area network – LAN Structure, MAC Sub layer, Logical link control layer, Random Access protocols ALOHA, Slotted ALOHA, CSMA, CSMA/CD, Scheduling approaches to medium access control – Reservation Systems, polling , Token passing rings, Channelization – FDMA, TDMA, CDMA. LAN Standard – Ethernet and IEEE 802.3. Self-study: IEEE 802.4 standard	7

UNIT-5	
ROUTING AND SECURITY: Routing algorithms classification, Routing tables, shortest path routing algorithms, Flooding, Hierarchical routing, Distance vector routing Link state routing, congestion control algorithms. Transport layer: TCP and UDP, IP Adress, IPV4, IPV6. Network security, ITU-T security Architecture, Three aspects of security, Security attacks, Security services, Security Mechanisms. Cryptography, Symmetric and asymmetric encryption, Caesar Cipher, Play fair Cipher.	10

Text Books:

1. Stallings, "Data and Computer Communications", 7th Edition, Pearson Education, 2012 Reference

References:

1. Andrew S Tanenbaum, "Computer Networks", 5th Edition, Pearson Education.
2. Behrouz Forouzan, Introduction to Data Communication & Networking TMH, 2019.
3. Larry &Peterson & Bruce S Davis; Computer networks Second Edition, Morgan Kaufman, 2000.
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 -2018.

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	7	15
Unit 2	8	18
Unit 3	10	15
Unit 4	7	15
Unit 5	10	15
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) = 60		

Course Code: CA4P2	Course Title: Computer Networks Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours:52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2Hrs

Objectives:

The student should be made

1. Exposed to learn the concepts using the network simulator
2. Learn to implement the algorithms on Error correction and detection.
3. Learn to use network security Cryptool.

Course Outcomes:

Here the students will be able to

1. Understand the concepts of Computer networks.
2. Study the basic network commands and network configuration commands.
3. Familiarize with the Transmission Media, Flow Control and Error Detection & Correction.
4. Understand fundamental concepts in Routing, Addressing & working of Transport Protocols.
5. Gain familiarity with network security.

LIST OF EXPERIMENTS

1. Implement the functionality of various network devices such as Repeater, Hub, Switch, Router, Digitizer, Bridge, and Gateway.
2. Implement IP Addressing, Subnetting, and Super netting to configure IP addressing schemes
3. Design and Connect Computers in a Local Area Network (LAN)
4. Execute basic commands like `ping`, `tracert`, `ipconfig`, and `nslookup` to analyze and test network connectivity.
5. Write a socket program for the implementation of echo.
6. Write a program for error detecting code using CRC-CCITT (16- bits).
7. Write a program for error detecting code using 2D Parity check.
8. Write a program to find the shortest path between vertices using Dijikstra's Algorithm.

9. Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.
10. Write a program for congestion control using leaky bucket algorithm.
11. Write a program for congestion control using Token bucket algorithm.
12. Write a program on datagram socket for client/server to display the messages on client side, typed at the server side.
13. Write a program to implement the Token Passing algorithm.
14. Write a program to encrypt and decrypt a Password

15. Implement the substitution mono alphabetic technique by using Caesar Cipher algorithm.
16. Write a program to implement Play fair Cipher algorithm.

Evaluation Scheme for Lab Examination

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

Course Code: CA4325	Course Title: Operating System Concepts
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2 Hrs.

Course Objectives:

1. Understand the basic concepts and functions of an operating system.
2. Learn the different types of operating systems and their structures.
3. Study process management techniques and process scheduling algorithms.
4. Explore memory management methods and virtual memory systems.
5. Understand disk and file system management, including scheduling and protection mechanisms.

Course Outcomes:

1. Define and describe the core functions of an operating system.
2. Analyze and apply process scheduling algorithms in different scenarios.
3. Implement process synchronization techniques and solve synchronization problems.
4. Apply memory management strategies and optimize system performance.
5. Manage disk scheduling and file systems efficiently with appropriate algorithms.

CONTENT	HOURS
UNIT 1:	
INTRODUCTION: Definition, functions, operating system structure, Types of operating system: single user, simple batch –spooling , multiprogramming, time – shared , Real time system, distributed system , parallel operating systems, system calls, Types of system calls.	**8**
UNIT 2:	
PROCESS MANAGEMENT: Process, Process life cycle, Process control block, operations on a process , Inter-process communication, Process scheduling, Scheduling Criteria and Algorithms: FCFS- SJF- Round Robin- priority scheduling, -multilevel queue, multilevel feedback algorithms,	**8**

UNIT 3:	
PROCESS SYNCHRONIZATION AND DEADLOCK: Process synchronization –critical section problems, semaphore, Peterson's Solution, classic problems of synchronization- Reader-Writer Problem, Dining Philosophers Problem **DEADLOCK**: Deadlock Characterization, Methods for Handling Deadlocks, wait for graph and resource allocation graph, Deadlock Prevention, Avoidance and Detection - bankers Algorithm; Recovery from Deadlock.	**10**
UNIT 4:	
MEMORY MANAGEMENT: Contiguous Memory Allocation, Swapping, Paging, Segmentation, Paged-segmentation, Virtual Memory management: Demand paging, Page replacement algorithms:-FIFO-LRU-optimal. Allocation of frames, page faults, Thrashing.	**8**
UNIT 5 :	
DEVICE AND FILE SYSTEM: Disk Structure, need for disk scheduling, Scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN,. **FILE SYSTEM**: -File concept; File Access methods; Directory structure; File system mounting; File Protection.	**8**

Text Book:

Operating System Concepts (10th Edition) by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, published by Wiley (2018).

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 Code:CA4425	**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, Incremental model, RAD model, Agile framework: values and principles of Agile, Agile manifesto, Extreme programming, Scrum Framework	**10**
UNIT 2:	

SOFTWARE REQUIREMENTS **Software requirements:** Functional- non-functional requirements, User requirement, Software Requirement engineering, Feasibility studies, Requirements elicitation and analysis, requirement validation, software requirement specification (SRS document).	**06**
UNIT 3:	
SOFTWARE DESIGN AND PROJECT MANAGEMENT **Software Design:** Design Process, Design strategies, Design characteristics, architectural design: System Structuring, control modelling, Modular decomposition, Data Flow Diagrams. **Software Project Management & Cost estimation Technique** Software Project Management: Software cost estimation techniques: top down, bottom-up approach, estimation by analogy, COCOMO model, LOC.	**08**
UNIT 4:	
SOFTWARE CODING AND SOFTWARE TESTING **Software Coding:** Introduction, coding techniques, styles, Coding verification techniques, code documentation. **Software Testing:** Introduction, verification and validation, A strategic approach to software testing, Unit testing, Black-box and white box and integration testing, system testing	**08**
UNIT 5:	
SOFTWARE QUALITY AND SOFTWARE MAINTENANCE **Software Quality:** Introduction, quality metrics, quality assurance: tasks, goals, quality planning, quality control, software reliability and software reuse. **Software Maintenance:** Introduction, corrective maintenance, adaptive maintenance, perfective maintenance, Preventive maintenance . **Introduction to DevOps**, Advantages of adopting DevOps practices	**10**

Text books:

1. Roger S. Pressman, "Software Engineering: A Practitioner's Approach", Alternate Edition, 7th Edition, McGraw Hill.
2. Ian Somerville, "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		