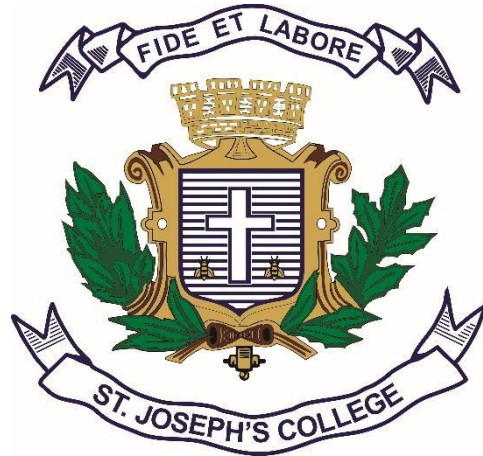


**ST. JOSEPH'S COLLEGE (AUTONOMOUS)**

**BENGALURU-27**

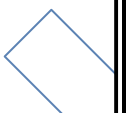


Re-accredited with 'A++' **GRADE with 3.79/4 CGPA** by  
NAAC Recognized by UGC as College of Excellence

**DEPARTMENT OF COMPUTER SCIENCE AND COMPUTER  
APPLICATIONS**

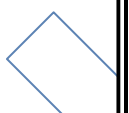
**SYLLABUS FOR UNDERGRADUATE PROGRAMME BCA**

For Batch 2021-2025



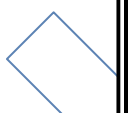
**The objectives of the Programme are:**

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 programming techniques 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**

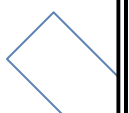
1. **Discipline knowledge:** Acquiring knowledge on basics of Computer Science and ability to apply to design principles in the development of solutions for problems of varying complexity
2. **Problem Solving:** Improved reasoning with strong mathematical ability to Identify, formulate and analyze problems 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 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 app for applicative problems.
6. **Modern Tool Usage:** Identify, select and use a modern scientific and IT tool or technique for modeling, prediction, data analysis and solving problems in the area of Computer Science and making them mobile based application software.
7. **Communication:** Must have a reasonably good communication knowledge both in oral and writing.
8. **Project Management:** Practicing of existing projects and becoming independent to launch own project by identifying a gap in solutions.
9. **Ethics on Profession, Environment and Society:** Exhibiting professional ethics to maintain the integrality 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. So, learn to learn ability.
11. **Motivation to take up Higher Studies:** Inspiration to continue educations towards advanced studies on 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 noted above 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)**  
**(2021-2024)**

<b>Semester 1</b>	<b>Code Number</b>	<b>Title</b>	<b>No. of Hours of Instructions</b>	<b>Number of Hours of teaching per week</b>	<b>Number of credits</b>	<b>Continuous Internal Assessment (CIA) Marks</b>	<b>End Semester Marks</b>	<b>Total marks</b>
Theory	CA1121	Fundamentals of Computers	42	03	03	40	60	100
Theory	CA1221	Programming in C	42	03	03	40	60	100
Theory	CA1321	Mathematical Foundations	42	03	03	40	60	100
Practical	CA1P1	Information Technology Lab	52	04	2	25	25	50
Practical	CA1P2	C Programming Lab	52	04	2	25	25	50

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

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

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

<b>Semester 3</b>	<b>Code Number</b>	<b>Title</b>	<b>No. of Hours of Instructions</b>	<b>Number of teaching hrs /week</b>	<b>Number of credits</b>	<b>Continuous Internal Assessment (CIA) Marks</b>	<b>End Semester Marks</b>	<b>Total marks</b>
<b>Theory</b>	<b>CA3121</b>	<b>Data Base Management Systems</b>	<b>42</b>	<b>03</b>	<b>03</b>	<b>40</b>	<b>60</b>	<b>100</b>
<b>Theory</b>	<b>CA3221</b>	<b>C# and DOT NET Framework</b>	<b>42</b>	<b>03</b>	<b>03</b>	<b>40</b>	<b>60</b>	<b>100</b>
<b>Theory</b>	<b>CA3321</b>	<b>Computer Communication and Networks</b>	<b>42</b>	<b>03</b>	<b>03</b>	<b>40</b>	<b>60</b>	<b>100</b>
<b>Practical</b>	<b>CA3P1</b>	<b>DBMS Lab</b>	<b>52</b>	<b>04</b>	<b>2</b>	<b>25</b>	<b>25</b>	<b>50</b>
<b>Practical</b>	<b>CA3P2</b>	<b>C# and DOT NET Framework Lab</b>	<b>52</b>	<b>04</b>	<b>2</b>	<b>25</b>	<b>25</b>	<b>50</b>
<b>Total Number of credits:13+13( Total Credit of OE, Languages, CAE, Voc, AECC, SEC)</b>								

<b>Semester 4</b>	<b>Code Number</b>	<b>Title</b>	<b>No. of Hours of Instructions</b>	<b>Number of teaching hrs /week</b>	<b>Number of credits</b>	<b>Continuous Internal Assessment (CIA) Marks</b>	<b>End Semester Marks</b>	<b>Total marks</b>
<b>Theory</b>	<b>CA4121</b>	<b>Python Programming</b>	<b>42</b>	<b>03</b>	<b>03</b>	<b>40</b>	<b>60</b>	<b>100</b>
<b>Theory</b>	<b>CA4221</b>	<b>Computer Multimedia and Animation</b>	<b>42</b>	<b>03</b>	<b>03</b>	<b>40</b>	<b>60</b>	<b>100</b>
<b>Theory</b>	<b>CA4321</b>	<b>Operating System Concepts</b>	<b>42</b>	<b>03</b>	<b>03</b>	<b>40</b>	<b>60</b>	<b>100</b>
<b>Practical</b>	<b>CA4P1</b>	<b>Python programming Lab</b>	<b>52</b>	<b>04</b>	<b>2</b>	<b>25</b>	<b>25</b>	<b>50</b>
<b>Practical</b>	<b>CA4P2</b>	<b>Multimedia and Animation Lab</b>	<b>52</b>	<b>04</b>	<b>2</b>	<b>25</b>	<b>25</b>	<b>50</b>
<b>Total Number of credits:13+13( Total Credit of OE, Languages, CAE, Voc, AECC, SEC)</b>								

## Model Course Content for BCA, Semesters I and II

### Semester: I

Course Code: CA1121	<b>Course Title:</b> 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.

### Course Content

Content	Hours
<b>Unit - 1</b>	
<b>Fundamentals of Computers:</b> 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



<b>Unit-2</b>	
<b>Introduction to computers:</b> 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
<b>Unit-3</b>	
<b>Operating System Fundamentals:</b> 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
<b>Unit-4</b>	
<b>Introduction to Database Management Systems:</b> 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
<b>Unit-5</b>	
<b>Internet and Web Basics:</b> 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.	10
<b>Introduction to Emerging technologies:</b> Overview and features of Augmented Reality, Virtual Reality, Artificial Intelligence, Machine learning, Data Warehousing, Data Mining, Big data analytics, Internet of Things.	

**Text Books:**

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

**Reference:**

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

## BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	10	29
Unit 2	10	30
Unit 3	6	26
Unit 4	6	26
Unit 5	10	27
<b>TOTAL</b>	42	138
<b>Maximum marks for the paper (Excluding bonus questions) = 100</b>		

Course Code: CA1P1	<b>Course Title: Information Technology Lab</b>
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 03

### **Part A: Hardware**

1. Identification of the peripherals of a computer, components in a CPU and their functions.
2. Assembling and disassembling the system hardware components of personal computer.
3. Basic Computer Hardware Trouble shooting.
4. LAN and WiFi Basics.
5. Operating System Installation – Windows OS, UNIX/LINUX, Dual Booting.
6. Installation and Uninstallation of Software – Office Tools, Utility Software (like Anti-Virus, System Maintenance tools); Application Software - Like Photo/Image Editors, Audio Recorders/Editors, Video Editors ...); Freeware, Shareware, Payware and Trialware; Internet Browsers, Programming IDEs,
7. System Configuration – BIOS Settings, Registry Editor, MS Config, Task Manager, System Maintenance, Third-party System Maintenance Tools (Similar to Cleaner and Jv16 PowerTools ...)

### **Part B: Software**

1. Activities using Word Processor Software
2. Activities using Spreadsheets Software
3. Activities using Presentation Software
4. Activities involving Multimedia Editing (Images, Video, Audio ...)
5. Tasks involving Internet Browsing
6. Flow charts: Installation and using of flowgarithms 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.

**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/>

**Evaluation Scheme for Lab Examination**

Assessment Criteria		Marks
Activity - 1 from Part A	Write up on the activity/ task	3
	Demonstration of the activity/ task	5
Activity-2 from Part B	Write up on the activity/ task	3
	Demonstration of the activity/ task	5
Viva Voice based on Lab Activities		5
Practical Records		4
Total		<b>25</b>

Course Code: CA1221	<b>Course Title: Programming in C</b>
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

### Course Content

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

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

**Text Books:**

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

**Reference Books:**

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

**BLUE PRINT**

Unit Nos	No.of Hours	Total marks for which questions are to be asked(including bonus questions)
Unit 1	10	29
Unit 2	12	31
Unit 3	8	29
Unit 4	6	31
Unit 5	6	18
TOTAL	42	138
Maximum marks for the paper (Excluding bonus questions) =100		

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

## **Programming Lab**

### **Part A:**

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

### **Part B:**

1. Program to find the length of a string without using built in function
2. Program to demonstrate string functions.
3. Program to demonstrate pointers in C
4. Program to check a number for prime by defining isprime( ) function
5. Program to read, display and to find the trace of a square matrix
6. Program to read, display and add two m x n matrices using functions
7. Program to read, display and multiply two m x n matrices using functions
8. Program to read a string and to find the number of alphabets, digits, vowels,



consonants, spaces and special characters.

9. Program to Reverse a String using Pointer

10. Program to Swap Two Numbers using Pointers

11. Program to demonstrate student structure to read & display records of n students.

12. Program to demonstrate the difference between structure & union.

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

### Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program - 1 from Part B	Flowchart / Algorithm	
	Writing the Program	3
	Execution and Formatting	5
Program -2 from Part B	Flowchart/Algorithm	
	Writing the Program	3
	Execution and Formatting	5
Viva Voice based on C Programming		5
Practical Record		4
Total		<b>25</b>

Course Code: CA1321	<b>Course Title:</b> 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
<b>Unit - 1</b>	
<b>Basic concepts of logic:</b> 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
<b>Unit - 2</b>	
<b>Operations on sets:</b> power set- Venn diagram Cartesian product-relations - functions- types of functions - composition of functions.	10
<b>Unit - 3</b>	
<b>Matrix algebra:</b> Introduction-Types of matrices-matrix operations-transpose of a matrix -determinant of matrix - inverse of a matrix- Cramer's rule	10
<b>Unit - 4</b>	
<b>Matrix:</b> finding rank of a matrix - normal form-echelon form cayley Hamilton theorem-Eigen values	6
<b>Unit - 5</b>	
<b>Differential calculus:</b> Functions and limits - Simple Differentiation of Algebraic Functions – Evaluation of First and Second Order Derivatives – Maxima and Minima	10

**Text Books:**

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

**Reference Books:**

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

## **BLUE PRINT**

<b>Chapter</b>	<b>No. of hours</b>	<b>Total marks for which questions are to be asked (Including bonus questions)</b>
Unit 1	10	30
Unit 2	6	26
Unit 3	10	30
Unit 4	6	22
Unit 5	10	30
<b>TOTAL</b>	<b>42</b>	<b>138</b>
<b>Maximum marks for the paper (Excluding bonus questions) = 100</b>		

## Semester: II

Course Code: CA2121	<b>Course Title: Data Structures using C</b>
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

### Course Content

<b>Unit - 1</b>	
<b>Introduction to data structures:</b> 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 - <i>malloc</i> , <i>calloc</i> , <i>realloc</i> and <i>free</i> . Algorithm Specification, Performance Analysis, Performance Measurement-Asymptotic notations.	6
<b>Unit - 2</b>	
<b>Arrays:</b> Basic Concepts – Definition, Declaration, Initialization, Operations on arrays; Types of arrays; Arrays as abstract data types (ADT); Representation of Linear Arrays in memory;	10

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.	
--	--

<b>Unit – 3</b>	
<b>Stacks:</b> 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
<b>Unit – 4</b>	
<b>Queues:</b> Basic Concepts – Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues;  <b>Linked list:</b> Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly linked list, Header linked list, Circular linked list; Representation of Linked list in Memory;  Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation; Garbage collection.	10
<b>Unit – 5</b>	
<b>Trees:</b> 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;  <b>Binary tree:</b> 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))

### **BLUEPRINT**

Unit Nos	No. of hours	Total marks for which questions are to be asked (including bonus questions)
Unit 1	6	18
Unit 2	10	31
Unit 3	8	29
Unit 4	10	31
Unit 5	8	29
TOTAL	42	138
Maximum marks for the paper (Excluding bonus questions) =100		

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

### **Programming Lab**

#### **Part A:**

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

#### **Part B:**

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

### **Evaluation Scheme for Lab Examination**

Assessment Criteria		Marks
Program – 1 from Part A	Flowchart / Algorithm	
	Writing the Program	3
	Execution and Formatting	5
Program -2 from Part B	Flowchart/Algorithm	
	Writing the Program	3
	Execution and Formatting	5
Viva Voice based on C Programming		5
Practical Record		4
<b>Total</b>		<b>25</b>

Course Code: CA2221	<b>Course Title: Object Oriented Programming with JAVA</b>
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
<b>Unit - 1</b>	
<b>Introduction to Java:</b> Basics of Java programming, Data types, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, Math class, Arrays in java.	6
<b>Unit - 2</b>	
<b>Objects and Classes:</b> 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
<b>Unit - 3</b>	
<b>Inheritance and Polymorphism:</b> 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
<b>Unit - 4</b>	
<b>Event and GUI programming:</b> 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.	
<b>Unit – 5</b>	
<b>I/O programming:</b> Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files.	6
<b>Unit – 6</b>	
<b>Multithreading and Exception handling in java:</b> 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.

### BLUEPRINT

Unit Nos	No. of hours	Total marks for which questions are to be asked (including bonus questions)
Unit 1	6	14
Unit 2	6	26
Unit 3	8	31
Unit 4	10	33
Unit 5	6	14
Unit 6	6	20
TOTAL	42	138
Maximum marks for the paper (Excluding bonus questions) =100		

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

### Course Outcomes (COs):

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

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

### Practice Lab

1. Program to print the following triangle of numbers  
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5
2. Program to simple java application, to print the message, "Welcome to java"
3. Program to display the month of a year. Months of the year should be held in an array.
4. Program to find the area of rectangle.
5. program to demonstrate a division by zero exception
6. Program to create a user defined exception say Pay out of Bounds.

### Programming Lab

#### PART A: Java Fundamentals OOPs in Java

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

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

#### **PART B: Exception Handling & GUI Programming**

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

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

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

#### **Evaluation Scheme for Lab Examination**

Assessment Criteria		Marks
Program – 1 from Part A	Flowchart / Algorithm	
	Writing the Program	3
	Execution and Formatting	5
Program -2 from Part B	Flowchart/Algorithm	
	Writing the Program	3
	Execution and Formatting	5
Viva Voice based on Programming		5
Practical Record		5
<b>Total</b>		<b>25</b>

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

### Course Outcomes (COs):

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

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

### Course Content

Content	Hours
<b>Unit - 1</b>	
<b>The Foundations:</b> 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
<b>Unit - 2</b>	
<b>Counting:</b> Basics of counting, Pigeonhole principle, Permutation and combination, Binomial Coefficient and Combination, Generating functions-Generating Permutation and Combination.  <b>Advanced Counting Techniques:</b> Applications of Recurrence Relations, Solving Linear Recurrence, Relations, Divide and Conquer Algorithms and Recurrence Relations, Inclusion-Exclusion, Applications of Inclusion-exclusion.	12
<b>Unit - 3</b>	
<b>Induction and Recursion:</b> Mathematical Induction, Strong Induction and Well-	12

Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Corrections. <b>Relation:</b> Properties of relation, Composition of relation, Closer operation on relation, Equivalence relation and partition. Operation on relation, Representing relation.	
<b>Unit - 4</b>	
<b>Graphs:</b> 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:**

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

**BLUE PRINT**

<b>Chapter</b>	<b>No. of hours</b>	<b>Total marks for which questions are to be asked (Including bonus questions)</b>
Unit 1	10	30
Unit 2	12	26
Unit 3	12	30
Unit 4	8	22
<b>TOTAL</b>	<b>42</b>	<b>138</b>
<b>Maximum marks for the paper (Excluding bonus questions) = 100</b>		

**Semester: III**

<b>Course Code:CA3121</b>	<b>Course Title: DATABASE MANAGEMENT SYSTEMS</b>
Course Credits:03	Hours/Week:03
Total Contact Hours:42	Formative Assessment Marks:40
Exam Marks:60	Exam Duration: 2 Hrs.

**Course Outcomes(COs):**

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

**Course Content**

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

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

**Text Books:**

1. Elmasri and Navathe: Fundamentals of Database Systems, 7th Edition, Addison -Wesley, 2016.

2. Silberschatz, Korth and Sudharshan Data base System Concepts, 7th Edition, Tata McGraw Hill, 2019.

Reference:

2. C.J. Date, A. Kannan, S. Swamynatham: An Introduction to Database Systems, 8th Edition, Pearson education, 2009

3. 2. Database Management Systems :Raghu Ramakrishnan and Johannes Gehrke: , 3rd Edition, McGrawHill, 2003



**Blueprint**

<b>Chapter</b>	<b>No. of hours</b>	<b>Total marks for which questions are to be asked (Including bonus questions)</b>
Unit 1	8	6
Unit 2	10	10
Unit 3	8	25
Unit 4	8	21
Unit 5	8	21
TOTAL	42	83
Maximum marks for the paper (Excluding bonus questions) = 60		

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

### **Course Outcomes (COs):**

This course will enable students to

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

### **SQL Programming**

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

Create another table called STATS to store normalized temperature and precipitation data:

- ID field must match some STATION table ID (so that name and location will be known).
  - Allowable ranges will be enforced for other values.
  - No duplicate ID and MONTH combinations.
  - Temperature is in degrees Fahrenheit.
  - Rainfall is in inches.
- 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 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 to Celsius 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 known to read 0.01 inches low
  - ii) Update one row, ID 44's July temperature reading, to correct a data entry error
  - 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)

## Evaluation Scheme for Lab Examination

<b>Assessment Criteria</b>	<b>Marks</b>
Creation of Table	5
Insertion of Tuples	5
SQL Queries	10
Viva Voice	5
Total	25

<b>Course Code: CA3221</b>	<b>Course Title: .NET Technologies</b>
Course Credits: 03	Hours/Week:03
Total Contact Hours:42	Formative Assessment Marks:40
Exam Marks:100	Exam Duration: 2 Hrs.

### Course Outcomes (COs):

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

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

<b>CONTENT</b>	<b>Hours</b>
<b>UNIT 1</b>	
<b>Introduction:</b> .NET Definition, Characteristics, Application, Difference between c and c#, The, the .NET framework, benefits of the .NET approach, C# and .NET, .NET Architecture-Common Language Runtime, MSIL, .NET Framework Classes. Advantages of Managed Code ,Garbage Collection ,Security. program structure , command line argument, Literals, variables and data types , constant variables, scope of variables, boxing and unboxing.	8
<b>UNIT 2</b>	
<b>Features of C# and OOP:</b> Operators in C#, expression, Decision making and looping statements in c# Methods in C# -declaring methods, nesting of methods, methods parameters, the output parameters, Static data members, Static member function, Arrays - variable size arrays, array list class, String handling . Overview of OOP , Defining a class, Adding Variables and Methods, Member Access Modifiers, Creating Objects, Accessing Class Members,	9
<b>UNIT 3</b>	
<b>Object Oriented C#:</b> Types of Constructors ,This Reference, , Inheritance and polymorphism : classical inheritance, containment inheritance, defining a subclass, visibility control, defining subclass constructors, multilevel inheritance, hierarchical inheritance, overriding methods, hiding methods, abstract classes, abstract methods, sealed classes : Preventing inheritance, sealed methods, polymorphism. Interfaces : Multiple Inheritance : defining an interface, extending an interface, implementing interface, interface & inheritance,	9
<b>UNIT 4</b>	

<b>Introduction to Visual Studio and Exceptions-ASP.NET.</b> Difference Between ASP and ASP.NET, Creating A web application using ASP.NET ,Components of an ASP.NET Custom Control, Deploying ASP.NET application .Errors and Exception Handling, Exceptions Classes User Defined Exceptions , Delegate- delegate declaration, delegate methods, delegates instantiation, delegate invocation, using delegates.	8
<b>UNIT 5</b>	
<b>Introduction to Data Access-</b> Comparison between ADO & ADO.NET—The difference between Connection Model & Disconnected Model – difference between the DataSet and RecordSet- The Dataset Model. Accessing Data using ADO.NET: dataset-DataAdapter DataRelation.	8

**TEXT BOOKS:**

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

**BLUEPRINT**

<b>UNITS</b>	<b>No. OF HOURS</b>	<b>Total Marks for which questions to be asked (Including Bonus Questions)</b>
UNIT-1	9	6
UNIT-2	6	2
UNIT-3	9	25
UNIT-4	9	25
UNIT-5	9	25
TOTAL	42	83
Maximum marks for the paper (Excluding bonus question) = 60		



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

**Course Outcomes (COs):**

This course will enable students to

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

Part A

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

6. Given below is a hypothetical table showing rates of Income Tax for male citizens below the age of 65 years:

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

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

7. Create a project to demonstrate the working of

a. Write a program to input the length and breadth of a rectangle and find its diagonal. using method  $diagonal = \sqrt{length^2 + breadth^2}$

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

8. Create a project to demonstrate the working of

a) Swapping of two values using ref keyword.

b) Method overloading for displaying int and character data.

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

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

a. Total marks obtained.

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

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

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

- 13 a. Write a program in C# Sharp to read a string through the keyboard and sort it using bubble sort
- b. Create a project to demonstrate the working of immutable string operation and find the time taken for the immutable string manipulation.
14. Project to implement the access details of the banking information system using the access modifier.
15. Project to design a calculator with all basic functionality using delegates.

## **Part B**

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

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

Member methods:  
A parameterized constructor to initialize the data members.  
To accept the details of a student.  
To compute the average and the maximum out of three marks.  
To display the name, age, marks in three subjects, maximum and average.  
Write a main method to create an object of a class and call the above member method.
5. Write a menu driven class to accept a number from the user and check whether it is a Palindrome or a Perfect number using multiple interfaces.
  - a. Palindrome number- (a number is a Palindrome which when read in reverse order is same as

read in the right order) Example: 11, 101, 151, etc.

b. Perfect number- (a number is called Perfect if it is equal to the sum of its factors other than the number itself.) Example:  $6=1+2+3$

6. Design a class to overload a function `compare()` as follows:

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

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

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

7. Project to implement run time polymorphism by creating a base class with two methods. In the first method, call the second method. Inherit a class and override the second method. create an object of the derived class.

8. Create a project to demonstrate the working of sealed class and sealed method.

For displaying by Defining a class Employee having the following description:

Instance variables:

`int pan` to store personal account number

`String name` to store name

`double tax_income` to store annual taxable income

`double tax` to store tax that is calculated

Member functions:

`input ()` Store the pan number, name, taxable income

`calc ()` Calculate tax for an employee

`display ()` Output details of an employee.

9. Write a program to input a number and check whether it is a Smith number or not. Smith number is such a number, the sum of whose digits equals the sum of the digits of its prime factors using multi cast delegate.

Smith number is a composite number in which the sum of its digits is equal to the sum of the digits of all its prime factors.

For Example 378 is a Smith Number as the sum of the digits of 378 are:  $3+7+8 = 18$ . The prime

Factors of 378 are: 2, 3, 3, 3, 7 (sum =  $2+3+3+3+7 = 18$ ).

Similarly 22 is a Smith Number as the sum of the digits are:  $2+2=4$ . The prime factors of 22 are:

2 and 11 (Sum =  $2+(1+1) = 4$ )

Other Examples include 27, 58, 85, 94, 121, 166, 202, 265, etc

10. Project to throw exception for the following

- *Array Index Out Of Bound*
- *Arraytypemismatchexception,*
- *Stackoverflowexception*
- *Division by zero.*

11. Project to implement multithreading with the use of multiplication tables. Three threads must be defined. Each one must create one multiplication table they are 5 table, 7 table, 13 table.
12. Write ASP.NET code to send data entered in two textboxes from one web page to another web page. Display the data on two separate labels.
13. Project to displays students' information suchas firstname, middle name, last name, Father Name, Mother name, DOB, Email, phone number, Gender, Address etc. With respect to a button click using HTML and ASP control.
14. Create a web-based application to check the entered user name and password are valid or not, Check the entered user name and password of a web form with a database connection using ADO.Net.

### **Evaluation Scheme for Lab Examination**

<b>Assessment Criteria</b>	<b>Marks</b>
Writing	8
Execution	10
Viva Voice	7
<b>Total</b>	<b>25</b>

<b>Course Code: CA3321</b>	<b>Course Title: DATA COMMUNICATION AND COMPUTER NETWORKS</b>
Course Credits: 03	Hours / Week:03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 02Hrs

### Course Outcomes (COs):

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

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

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

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

**Text Books:**

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

**BLUEPRINT**

<b>UNITS</b>	<b>No. OF HOURS</b>	<b>Total Marks for which questions to be asked (Including Bonus Questions)</b>
UNIT-1	9	6
UNIT-2	6	2
UNIT-3	9	25
UNIT-4	9	25
UNIT-5	9	25
<b>TOTAL</b>	<b>42</b>	<b>83</b>
Maximum marks for the paper (Excluding bonus question) = 60		

## SEMESTER IV

<b>Course Code: 4121</b>	<b>Course Title: Python Programming</b>
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 02Hrs

### Course Outcomes (COs):

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

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

### Course Content

Content	Hours
<b>Unit – 1</b>	
<b>Basics of Python program</b>  Printing text, printing the result of a calculation, Strings in Python, The Escape Character, Variables and Types, Numeric Data Types in Python, Numeric Operators, Expressions, Integer division, Operator Precedence. Programming examples for each concept. String and Slice:  The String Data Type, Negative Indexing in Strings, Slicing, Slicing with Negative Numbers, Using a Step in a Slice, Slicing Backwards, String Operators, String Replacement Fields, String Formatting. Programming examples for each concepts.	8
<b>Unit – 2</b>	
<b>Program Flow Control &amp; Functions</b>  Conditional blocks using if, else and elif, for loops in, for loop using ranges, use of while loops in python, Loop manipulation using pass, continue, break and else. Organizing python codes using functions, organizing python projects into modules, importing own module as well as external modules, Understanding Packages.	8
<b>Unit – 3</b>	



<p><b>String, List, Dictionary &amp; File handling:</b></p> <p>Understanding in-built string methods, List manipulation and Dictionary manipulation. Programming using string, list and dictionary in-built functions. Reading files in python, writing files in python, Understanding read and write functions, manipulating file pointer using seek, Programming using file operations.</p>	9
<b>Unit – 4</b>	
<p><b>OO Programming &amp; Regular Expressions:</b></p> <p>Concept of class, object and instances, constructor, class attributes and destructors, Inheritance, overlapping and overloading operators, Adding and retrieving dynamic attributes of classes. Pattern matching and searching, Power of pattern searching using regex in python, Pattern finding programs using regular expression.</p>	9
<b>Unit – 5</b>	
<p><b>Exception Handling &amp; Database Interaction:</b></p> <p>Avoiding code break using exception handling, Safe guarding file operation using exception handling, Handling and helping developer with errorcode. SQL Database connection using python, Creating and searching tables, Reading, storing and updating information on database.</p>	8

**REFERENCES:**

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

**BLUEPRINT**

**Title of the paper: Programming in Python**

Chapter	Number of Hours	Total marks for which the question are to be asked (including bonus questions)
UNIT I	8	6
UNIT II	8	6
UNIT III	9	25
UNIT IV	9	25
UNIT V	8	21
TOTAL	42	83
<b>Maximum marks for the paper (Excluding bonus questions) = 60</b>		

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

### Course Outcomes (COs):

This course will enable students to

- Understand the menu driven program and its approach in making the program more interactive.
- Understand the need of Object Oriented Programming Paradigm.
- Analysis the various features like built library and data structure to manage the various types of data.

### Python Program List

(Do not use inbuilt functions wherever possible)

1. Create a Menu Driven Program using Python that takes a number as input and prints the following:
  - a. Reverse of the given number
  - b. Sum of Digits
  - c. Palindrome or Not
  - d. Series "1 + 2 +.... +n="
  - e. Smallest Divisor of the given number
2. Create a Menu Driven Program using Python that takes a string as input and prints the Following:
  - a. Reverse the string
  - b. Length of the string
  - c. Number of Vowels in a String
  - d. Number of Uppercase Letters and Lower Case Letters in a String
  - e. New string that does not have any duplicate characters
3. Create a Menu Driven Program using Python that takes a List of Numbers from the user and does the following:
  - a. Print the Second Largest Number in the List
  - b. Swap the first and last items of that list and print
  - c. Remove the duplicate members in that list and print.
  - d. Print the number Occurring Odd Number of Times in a List
  - e. Print the list which has the square of the numbers in the given List.
4. Create a Menu Driven Program using Python that takes two lists from the user and does the following:
  - a. Print the Union of the Lists
  - b. Print the Intersection of the Lists
  - c. Print the Merged and sorted List
  - d. Print the sum of all the numbers in both the lists
  - e. Print the average of all numbers in both the lists
5. Create a Menu Driven Program using Python that does the following:
  - a. Creates and Prints a Dictionary with numbers and their squared values till 10
  - b. Print the sum of all values in the Dictionary

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

12. Create a Menu Driven Program in Python that does the following:
- Accept a sentence and reverse each word in it
  - Take a string input and sort them based on the number of characters in each word and mention the count of letters in a word at the end.
  - Take a number as string type and convert it into an integer without using inbuilt functions
  - Take a String and append a digit after every character with total count of that character in that string (ignore case)

13. You are given input file input\_digits\_P1.txt and input\_words\_P1.txt which contains Comma separated random integers. E.g. 2, 3, 10, 8, 7, 20, 11, 25, 30, 17, 12.

- Read the integers from the input\_digits\_P1.txt file and calculate the Sum of the Odd and Even numbers.
- Create a file with the name sum\_of\_odd\_numbers.txt and store the sum of odd Numbers in that file. E.g. 63 Create a file with the name sum\_of\_even\_numbers.txt and store the sum of even numbers in that file.e.g. 82
- In the same program read the words from input\_words\_P1.txt (E.g.: Apple, Ball, Cat, Dog, Egg) separated by comma and generate the below output using the digits read in Output:  
2 Apple, 10 Ball, 8 Cat, 20 Dog, 30 Egg, 12 Apple Save the final output in a file final\_output.txt.

14. Create a Menu Driven Program using Python that performs the following using the Database Structure mentioned below: Table Name: Student Fields: student\_id, student\_name, student\_course, student\_email, student\_dob

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

- Write a Python program that matches a string that has an *a* followed by zero or more b's.
- Write a Python program to find the sequences of one upper case letter followed by lower case letters.
- Write a Python program to find the substrings within a string.
- Write a Python program to convert a date of yyyy-mm-dd format to dd-mm-yyyy format.

### Evaluation Scheme for Lab Examination

Assessment Criteria	Marks
Writing	8
Execution	10
Viva Voice	7
<b>Total</b>	<b>25</b>

<b>Course Code: 4221</b>	<b>Course Title: Computer Multimedia and Animation</b>
Course Credits: 03	Hours/Week: 03
Total Contact Hours:42	Formative Assessment Marks: 40
Exam Marks:100	Exam Duration: 3 Hrs.

### Course Outcomes (COs):

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

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

### Course Content

Content	Hours
<b>Unit - 1</b>	
<b>Introduction to Multimedia:</b> History of Multimedia, its market – Content copyright – Resources for multimedia developers – Types of produces – Evaluation – Hardware Architecture – OS and Software – Multimedia Architecture – Software library – Drivers.	6
<b>Unit - 2</b>	
<b>Computer Fonts and Hypertext:</b> Usage of text in Multimedia, Families and faces of fonts, outline fonts, bitmap font International character sets and hypertext, Digital fonts techniques. Text file formats and Compression techniques.	6
<b>Unit - 3</b>	
<b>Image fundamentals and representations</b> Colour Science , Colour, Colour Models, Colour palettes, Dithering, 2D Graphics, Image Compression and File Formats :GIF, JPEG, JPEG 2000, PNG, TIFF, EXIF, PS, PDF, Basic Image Processing [ Can Use Photoshop ], Use of image editing software, White balance correction, Dynamic range correction, Gamma correction, Photo Retouching.	10

<b>Unit - 4</b>	
<b>Audio fundamentals and representations</b> Digital Audio : Audio software and hardware, Characteristics of sound and Digital audio Digitization of sound, frequency and bandwidth, decibel system, data rate, audio file format, Sound synthesis, MIDI, wavetable, Digital Audio systems, Audio file formats – Using Audio in Multimedia Applications – Audio for content. Compression and transmission of audio on Internet, Adding sound to your multimedia project.	10
<b>Unit 5</b>	
<b>Video and Animation</b> Video Basics , Video software and hardware, How Video Works, Broadcast Video Standards, Analog video, Digital video, Video Recording and Tape formats, Shooting and Editing Video (Use Adobe Premier for editing), Video Compression and File Formats. Video compression based on motion compensation, MPEG-1, MPEG-2, MPEG-4, MPEG-7, MPEG-21, Animation: Cell Animation, Computer Animation, Morphing. Video capture and playback systems.	10

#### **Text Books:**

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

Chapter	Number of Hours	Total marks for which the questions are to be asked (including bonus questions)
Unit I	6	1
Unit II	6	5
Unit III	10	26
Unit IV	10	26
Unit V	10	25
<b>TOTAL</b>	<b>42</b>	<b>83</b>
<b>Maximum marks for the paper (Excluding bonus question)= 60</b>		

<b>Course Code: 4P2</b>	<b>Course Title: Computer Multimedia and Animation Lab</b>
Course Credits: 02	Hours/Week:04
Total Contact Hours:52	Formative Assessment Marks: 25
Exam Marks:25	Exam Duration: 03Hours

### Course Outcomes (COs):

This course will enable students to:

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

### Practical Program list

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

### Evaluation Scheme for Lab Examination

Assessment Criteria	Marks
Writing	8
Execution	10
Viva Voice	7
<b>Total</b>	<b>25</b>

<b>Course Code: CA4321</b>	<b>Course Title: Operating System Concepts</b>
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 3 Hrs.

### Course Outcomes (COs):

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

### COURSE CONTENT:

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



Text Books:

1. Operating System by Milan Milenkovic, McGraw Hill.

References:

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

BLUEPRINT

<b>UNITS</b>	<b>NO OF HOURS</b>	<b>Total Marks for which questions to be asked (Including Bonus Questions)</b>
UNIT-1	9	6
UNIT-2	6	6
UNIT-3	9	21
UNIT-4	9	25
UNIT-5	9	25
TOTAL	42	83
<b>Maximum marks for the paper (Excluding bonus question) =</b>		