Course Name : **Bachelor of Science**
Discipline : **Computer Science**
(Those who join in June 2024 and after)

COURSE OBJECTIVES:

The syllabus for B.Sc. Computer Science degree under semester system has been designed on the basis of Choice Based Credit System, (CBCS) which would focus on job oriented programmes and Value Added Education.

To promote the students to understand the basic knowledge in the field of Computer Science, to train them in writing Programs in different Computer Languages and prepare the students to fulfill the need of the IT industry. To encourage the aspiring students for higher education

ELIGIBILITY FOR ADMISSION:

Candidate should have passed the Higher Secondary Examination conducted by the Board of Higher Secondary Education, Govt. of Tamil Nadu or any other Examinations accepted by the syndicate as equivalent there to with Mathematics / Computer Science as one of the Subjects.

DURATION OF THE COURSE: Three Years

COURSE SCHEME:

I year B.Sc. COMPUTER SCIENCE

Semester	Part	Subject Name	Hours	Credit	Int + Ext =Total	Local	Regional	National	Global	Professional Ethics	Gender	Human Values	Environment & Sustainability	Employability	Entrepreneurship	Skill Development	Subject Code	Revised / New / No Change / Interchanged & Percentage of Revision
I	Part - I	Tamil	6	3	25+75=100												**U24PT11**	**New**
	Part - II	English	6	3	25+75=100												U23PE11	No Change
	Core	C Programming	4	4	25+75=100				✔					✔		✔	**U24CSC11**	**New**
	Core Lab	**LAB:** Programming in C	6	3	40+60=100				✔					✔		✔	**U24CSCP11**	**New**
	Core	Digital Principles and Applications	4	4	25+75=100				✔							✔	**U24CSC12**	**New**
	Allied	Mathematical Foundation I	4	4	25+75=100				✔							✔	**U24MAAC11**	**New**
	Total		**30**	**21**														
II	Part - I	Tamil	6	3	25+75=100												**U24PT21**	**New**
	Part - II	English	6	3	25+75=100												U23PE21	No Change
	Core	Advanced C Programming	4	4	25+75=100				✔					✔		✔	**U24CSC21**	**New**
	Core Lab	**LAB:** Programming in Advanced C	6	3	40+60=100				✔					✔		✔	**U24CSCP21**	**New**
	Core	Computer Organization	4	4	25+75=100				✔							✔	**U24CSC22**	**20% Revised**
	Allied	Mathematical Foundation II	4	4	25+75=100				✔							✔	**U24MAAC21**	**New**
	Total		**30**	**21**														

Year	Part	Subject	Credit	Int = Total	Code
I & II	Part V	NSS / NCC / Physical Education/ YRC / RRC	3	100 = 100	U22NS4 / U22NC4 / U22PS4 / U22YR4 / U22RR4

SEMESTER III							
PART	Sub. Code	Name of the Subject	Hours	Credits	INT+ EXT = TOT	(Revised / New / No Change / Interchanged (R/N/C/I))	Courses having focus on employability/ entrepreneurship/ skill development
I		Tamil	6	3	25+75=100		
II		English	6	3	25+75=100		
III		Java Programming	5	4	25+75=100		Employability / Skill Development
III		Data Structures	4	4	25+75=100		Skill Development
III		LAB: Programming in Java	5	3	40+60=100		Employability / Skill Development
III		Resource Management Techniques	4	4	25+75=100		Skill Development
		Value Education	-	3			
TOTAL			30	24			
SEMESTER IV							
I		Tamil	6	3	25+75=100		
II		English	6	3	25+75=100		
III		Advanced Java Programming	4	4	25+75=100		Employability / Skill Development
III		Operating Systems	4	4	25+75=100		Skill Development
III		LAB: Programming in Advanced Java	6	3	40+60=100		Employability / Skill Development
III		Numerical Methods	4	4	25+75=100		Skill Development
		Environmental Studies	-	2			
V		NSS/NCC/RRC/YRC/PE		1			
TOTAL				24			

Semester	Part	Name of the Subject	Hours	Credit	Int + Ext= Total	Subject Code	Revised/New/ No Change / Interchanged& Percentage of revision	Courses having focus on employability/ entrepreneurship/ skill development
V	Part III Core	Computer Algorithms	4	4	25+75=100			Skill Development
	Part III Core	LAB: PHP and MYSQL	5	3	40+60=100			Employability, Entrepreneurship
	Part III Core	LAB: Python Programming	5	3	40+60=100			Employability, Entrepreneurship
	Part III Elective	Data Science / Computer Networks/ Information Security	5	5	25+75=100			Employability
	Part III Elective	Data Base Management Systems / Artificial Intelligence/Cloud Computing	5	5	25+75=100			Skill Development
	SBE-3	LAB: Full Stack Development - I	2	2	25+75=100			Employability, Entrepreneurship
	SBE-4	Employability Skills	2	2	25+75=100			
	NME1	LAB: Office Automation	2	2	40+60=100			Employability
TOTAL			30	26				

VI	Part III Core	Software Engineering	4	4	25+75=100			Skill Development
	Part III Core	Computer Graphics and Digital Image Processing	4	4	25+75=100			Skill Development
	Part III Core	Mobile Computing	5	4	25+75=100			Employability
	Part III Elective	Project: Software Development	5	4	Internal 100=100			Employability, Skill Development
	Part III Core	LAB: Mobile APP Development	6	3	40+60=100			Employability, Entrepreneurship
	SBE-5	LAB: Advanced Python Programming	2	2	40+60=100			Employability, Entrepreneurship
	SBE-6	LAB: Full Stack Development - II	2	2	40+60=100			Employability, Entrepreneurship
	NME2	LAB: Fundamentals of Web Designing	2	2	40+60=100			Employability
		TOTAL	30	25				

C Programming

Subject Code: U24CSC11 **Credit: 4**
Contact Hrs per Week: 4 **Contact Hrs per Semester: 90**

Objectives of the course:
To familiarize the entry-level fresh students with basic concepts of computer programming, to present the syntax and semantics of the "C" language as well as data types offered by the language and to allow the students to write their own programs and develop their debugging skills.

Course Outcomes:
CO 1: Understand about computer language
CO 2: Understand about basic operations
CO 3: Understand Decision making statements and writing programs using that
CO 4: Understand how to keep large volume of data and string operations
CO 5: Understand how to handle repeated tasks in a program

UNIT 1 **(12 Hrs)**

Overview of C: History of C – Importance of C – Sample Programs– Basic Structure of C Programs – Programming Style. **Constants, Variables and Data Types:** Introduction – Character Set – C Tokens – Keywords and Identifiers – Constants – Variables – Data Types – Declaration of Variables – Assigning Values to Variables – Defining Symbolic Constants – Declaring a Variable as Constants – Declaring a Variable as Volatile. **Operators and Expressions:** Operators: Arithmetic, Relational, Logical, Assignment, Increment & Decrement, Conditional, Bitwise and Special Operators – Arithmetic Expressions – Evaluation of Expressions – Precedence of Arithmetic Operators –Computational Problems – Type Conversion – Operator Precedence and Associativity. **Managing Input and Output Operations:** Introduction – Reading & Writing a Character – Formatted Input & Output.

UNIT 2 **(12 Hrs)**

Decision Making and Branching: Introduction – Decision Making with **If** Statement – Simple **If** Statement – The **If…Else** statement – Nesting of **If…Else** Statements – The Else **If** Ladder – The **Switch** Statement – The **?:** question Operator – The **goto** Statement.

UNIT 3 **(12Hrs)**

Decision Making and Looping: Introduction – The **while** Statement – The **do** Statement – The **for** Statement – Jumps In Loops – Concise Test Expressions.

UNIT 4 **(12Hrs)**

Array: Introduction – One-Dimensional Arrays: Declaration & Initialization— Two-Dimensional Arrays: Initialization – Multi-Dimensional Arrays – Dynamic Arrays – More About Arrays. **Character Arrays & Strings:** Introduction – Declaring and Initializing String Variables – Reading Strings from Terminal – Writing Strings to Screen – Arithmetic Operations on Characters – Putting Strings Together – Comparison of Two Strings – String-Handling Functions – Table of Strings – Other Features of Strings.

UNIT 5 **(12 Hrs)**

User-Defined Functions: Introduction – Need for User-Defined Functions – A Multi-Function Program – Elements & Definition – Return Values and Their Types – Function Calls – Function Declaration – Category of Functions– Nesting of Functions – Recursion – Passing Arrays to Functions – Passing Strings to Functions – The Scope, Visibility and Lifetime of Variables – Multi-file Programs.

Text Book:

- E. Balagurusamy, **Programming in ANSI C,** Mc Graw hill Education., Seventh Edition, 2017.

UNIT I: **Chapters** 1 and 2
UNIT II: **Chapters** 3 and 4
UNIT III: **Chapters** 5 and 6
UNIT IV: **Chapters** 7 and 8
UNIT V: **Chapters** 9

QUESTIONS: In the Semester Summative Examination, Minimum 50% questions to be asked to write Programs.

[Key concepts, Brief Cases, Review Questions, Debugging & Programming Exercises of each chapter must be read & practiced at home is absolutely necessary to fulfill the objectives.]

LAB: PROGRAMMING IN C

Contact Hours per week : 6 **Subject Code: U24CSCP11**
Contact Hours per semester : 90 **Credit: 3**

Objectives:

Train the students to write simple programs using procedure oriented language C.

Course Outcomes:

CO 1: Get practiced to covert small problems into computer instructions
CO 2: Get Practiced to handle large volume of data and perform operations on it
CO 3: Get practiced to handle strings and perform operations on it
CO 4: Get practiced to write code to handle repeated task in a problem.

I. Simple Programs

1. Write C programs for simple problems like sum & average of marks, Eb-Bill generation, Salary Bill generation.
2. Write a C program to print greatest of three numbers and print multiplication table for any given number.
3. Write a C program to find whether the given number is odd, even, negative, OR positive
4. Write a C program to carry out four arithmetic operations and display Sunday to Saturday using switch statement.
5. Write a C program to generate Fibonacci numbers and check whether generated numbers are prime or not.
6. Write C programs to find value for Sin, and Cos series; Adam number or not, Armstrong or not, Perfect number or not.
7. Write a C program to perform sum of digits, reverse a given number and to check whether it is palindrome or not.

II. Programs Using Arrays

8. Write a C program to compute sum of 10 elements of an array & display the sum and average.
9. Write a C program to find the biggest & smallest of elements of an array.
10. Write a C program to sort a set of numbers of an array.
11. Write a C program to search an element in an array using linear search technique.
12. Write a program to find number of odd and even numbers; Number of positive numbers and negative numbers.
13. Write a C program to add and subtract two matrices of order mXn.
14. Write a C program to compute multiplication of two matrices of order m X n and p X q.

III. Programs Using Functions and Strings

15. Write a C program to find square of a number & find smallest of 3 numbers using functions.
16. Write a C program to compute factorial of a given number using recursion.
17. Write a C program to find whether a string is palindrome or not.
18. Write C programs to find factorial, Fibonacci, and GCD using recursion.
19. Write a C program to count the number of vowels, consonants and words in a given text.
20. Write a C function to compare two strings without using string function.
21. Write a C function to reverse a given string without using string function.

DIGITAL PRINCIPLES AND APPLICATIONS

Subject Code: U24CSC12 **Credit: 4**
Contact Hours per week: 4 **Contact Hours per semester: 60**

Objectives: To provide basic knowledge on Digital Electronics and to understand the working principles of Digital computer building blocks like ALU (i.e. Combinational logic circuit) and Registers (i.e. Sequential logic circuit)

Course Outcomes:

CO 1: To understand number systems, codes and conversion as well as logic gates

CO2 : To facilitate understanding of Boolean simplification in logic circuit design

CO 3: To know the applications of different combinational logic circuits

CO 4: To understand the logic circuit Adder and binary level arithmetic manipulation

CO 5: To know the basic operation of Flip Flop and design of sequential logic circuits using it

UNIT I Digital Logic

Binary Number System –Fractions-Octal Number System- Hexadecimal Number System – ASCII - Excess 3- Gray codes - Basic Gates – Boolean algebra – NOR Gates – NAND Gates

UNIT II Boolean Simplifications

Boolean Laws and Theorems – Sum of Products Methods – Truth Table to Karnaugh Map – Pairs, Quads and Octets – Karnaugh Simplification (SOP Method) - Don't care conditions - Product of Sums method

UNIT III Data Processing Circuits

Multiplexers – De Multiplexers - Decoders –BCD to Decimal Decoder – Seven Segment Decoder - Encoders – Exclusive OR Gates – Parity Generators and Checkers – Magnitude Comparator

UNIT IV Arithmetic Circuits

Binary Addition – Binary Subtraction – Sign-Magnitude – 1's Complement, 2's Complement representation –1's Complement Arithmetic – 2's Complement Arithmetic – Arithmetic Building Blocks – The Adder/Subtrater.

UNIT V Flip Flops

RS Flip Flops- D Flip Flops –Flip Flop Timing - JK Flip Flops – JK Master Slave Flip Flops – Types of Shift Registers – Serial In Serial Out – Serial In Parallel Out – Asynchronous counters –Synchronous Counters- Mod Counters

Text Book: *Digital Principles and Applications: Donald P.Leach, Albert Paul Malvino, Goutan Saha, Mc Graw Hill, 8th Edition, .*

UNIT I – Chapters - 5.1 to 5.10 & 2.1 to 2.2

UNIT II – Chapters – 3.1 to 3.8

UNIT III – Chapters – 4.1 to 4.9

UNIT IV – Chapters – 6.1 to 6.8

UNIT V – Chapters – 8.1 to 8.8, 9.1 to 9.3 & 10.1 to 10.3

Reference Book: Digital Electronics: principles and applications, Roger L. Tokheim, Mc Graw Hill, 1998.

ALLIED - MATHEMATICAL FOUNDATIONS I

Contact Hours per week :4 **Subject Code: U24MAAC11**
Contact Hours per semester: 60 **Credit: 4**

Course Outcomes:
On the successful completion of the course, students will be able to
CO 1: Able to apply the rules of propositional logic and rules of inference in verifying the validity of an argument or set of atatements.
CO 2: Well versed in using graph models for several problems in science and engineering such as network analysis, scheduling problem, social networks etc to get solution
CO 3: Well equipped in solving problems which are in recursive nature by the methods of recurrence relations
CO 4: Able to use / develop suitable algorithms for finding various closure of a relation which are vital in the field of networks
CO 5: Potential enough to use the concept of eigen values and eigen vectors in communication networks, designing, manufacturing, image processing and so on

UNIT I:
The Foundation-logic and proofs: Propositional logic and its applications – Propositional equivalences – (Exclude propositional satisfiability, applications of satisfiability, solving satisfiabilty problems, and its related problems)- predicates and quantifiers – rules of inference.

UNIT II:
Relations: Relations and their properties – representing relations – closures of relations – partial orderings (Theorems statement only; exclude lexicographic ordering – exclude lattices).

UNIT III :
Counting: The basic of counting – the pigeonhole principle – permutation and combinations – applications of recurrence relations – solving recurrence relations – divide and conquer algorithms and recurrence relations (All theorems and result statement only)

UNIT IV:
Graphs : Graphs and graphs model (Excluding biological networks; tournaments; all its related examples, and problems) – graph terminology and special types of graphs – representing graphs and isomorphism – connectivity (paths – connectedness in undirected graph – paths and isomorphism- counting paths between vertices) –shortest path problems.

UNIT V:
Matrices : Introduction – operations – inverse – Rankof a matrix – solution of simultaneous linear equation – Eigen values and eigen vectors.

Text Book:
1. Kenneth H. Rosen, "Discrete mathematics and applications", McGrawhill.
2.M.Venkataraman, N.Sridharan, N.Chandrasekaran, " Discrete Mathematics", The national Publishing company, 2009

UNIT I : Text Book 1: Chapter 1 Sections ; 1.1 – 1.4, 1.6
UNIT II : Text Book 1: Chapter 9 Sections ; 9.1, 9.3 – 9.6
UNIT III : Text Book 1: Chapter 6 sections: 6.1 – 6.3; Chapter 8 sections: 8.1-8.3 (pages 527-529 only) (Exclude algorithms and relations on page 507 and its related problems).
UNIT IV: Text Book 1: Chapter 10 sections: 10.1-10.4, 10.6
UNIT V : Text Book 2: Chapter 6 sections 6.1-6.5, 6.7

SEMESTER II

Advanced C Programming

Contact Hours per week: 4 **Subject Code: U24CSC21**
Contact Hours per semester: 60 **Credit: 4**

Objectives:

The course is intended for intermediate-level students who practiced basic features of C and who want to take their skills to the next level. It builds on students' existing background in C to complete their knowledge in the conceptual and syntactic structures needed to master dynamic data structures, string parsing and numeric conversion, memory management, bit-level manipulation, and other advanced techniques.

Course Outcomes:

CO 1: Understand structure concept and to write code using that
CO 2: Understand to access data in the computer memory directly
CO 3: Understand to keep data in a computer permanently and to access those data
CO 4: Understand to dynamically access memory and return them when it is unwanted
CO 5: Understand to keep data efficiently and process those data.
CO 6: Understand related concepts on this course from Peer Students.

UNIT 1: Structures and Unions **(12 Hrs)**

Introduction – Defining a Structure – Declaring Structure Variables – Accessing Structure Members – Initialization – Copying and Comparing Structure variables – Operations on Individual Members – Arrays of Structures – Arrays within Structures – Structures within Structures – Structures and Functions – Unions – Size of Structures – Bit Fields.

PEER TEACHING: Finding area and volume of a cone, Adding two complex numbers, adding two train times using structure.

UNIT 2: Pointers **(12 Hrs)**

Introduction – Understanding Pointers – Accessing the Address of a Variable – Declaring Pointer Variables – Initialization of Pointer Variables – Accessing a Variable Through its Pointer – Chain of Pointers – Pointer Expressions – Pointer Increments and Scale Factor – Pointers and Arrays – Pointers as Function Arguments – Functions Returning Pointers – Pointers to Functions – Pointers and Structures – Troubles with Pointers.

PEER TEACHING: Using pointer Searching Element in an array, searching even numbers in a Matrix, Adding Two Matrices.

UNIT 3: File Management in C **(12 Hrs)**

Introduction – Defining and Opening File – Closing a File – Input/Output Operations on Files – Error Handling During I/O Operations – Random Access to Files – Command Line Arguments.

PEER TEACHING: Generating Salary bill, Electricity bill, Mark statement using Sequential File.

UNIT 4: Dynamic Memory Allocation and Linked Lists **(12 Hrs)**

Introduction – Dynamic Memory Allocation– Allocating a Block of Memory: **malloc** – Allocating Multiple Blocks of Memory: **calloc** – Releasing the Used Space: **free** Altering the Size of a Block: **realloc** – Concepts of Linked Lists – Advantages of Linked Lists –

Types of Linked Lists – Pointers Revised – Creating a Linked List – Inserting and Deleting an Item

PEER TEACHING: Application of Linked Lists.

UNIT 5:The Preprocessor, Bit-Level Programming (12 Hrs)

Introduction – Macro Substitution – File Inclusion – Program Design and Coding, Errors, Program Testing and Debugging, Writing Efficient Programs - Bitwise Logical, Shift, Complement Operators - Masking.

PEER TEACHING: Graphics Programming- Basic shapes, Graphics effects, Lines with different colors.

Text Book:

- E. Balagurusamy, **Programming in ANSI C,** Mc Graw hill Education., Eighth Edition, 2017.

UNIT I: Chapter 10;
UNIT II: Chapter 11;
UNIT III: Chapter 12;
UNIT IV: Chapter 13;
UNIT V: Chapter 14, 15 and Appendix I

Reference Book:

NOTE ABOUT QUESTIONS: In the Summative Examination, 50% of the Questions to be asked to write Programs.

[Key concepts, Brief Cases, Review Questions, Debugging & Programming Exercises of each chapter must be read & practiced at home is absolutely necessary to fulfill the objectives. Summer Projects may be assigned to students.]

LAB: Programming in Advanced C

Subject Code: U24CSCP21 **Credit: 3**
Contact Hours per week: 6

Course Outcome:

CO 1: Get practiced to code using structure
CO 2: Get practiced to directly access and operate the data in the memory
CO 3: Get practiced to write code to keep large data and access it
CO 4: Get practiced to write code to dynamic access the data from memory and perform operations on it

I Cycle: Structure Concepts

1. Write a program to process student marks in an examination using Structure.
2. Write a program to process salary of an employee using Structure.
3. Write a program to process inventory of a book shop using Structure.
4. Write a program to add two complex numbers using Structure.
5. Write a program to increment time by one second using Structure.

II Cycle: Pointer Concepts

6. Write a program to compute sum of elements stored in an array using Pointers
7. Write a program to determine the length of a given string and to reverse it using Pointers.
8. Write a function using pointers to exchange the values of two variables.
9. Write a function that compares two integer arrays to see if they are identical using pointers.
10. Write a program to process exam marks of a set of students using structure pointers.

III Cycle: File Concepts

11. Write a program to separate Odd and Even numbers in different files using File concepts.
12. Write a program to prepare Electricity Bill using File concepts.
13. Write a program to prepare inventory report of a book shop using File concepts.
14. Write a program to prepare students marks sheet using File concepts.
15. Write a program to prepare salary bill of employees of a company using File concepts.

IV Cycle: Dynamic Memory Allocation and Bit-Level Programming concepts

16. Write a program to store a string in a memory buffer, and, to modify the original buffer size to store larger string using dynamic memory allocation concepts.
17. Write a program to create a linear linked list of vowels and to display the number of vowels.
18. Write a program to insert a given number into a linked list of sorted numbers.
19. Write a program to delete a given number from a linked list of sorted numbers.
20. Write a program to test whether a given number is odd or even using bitwise AND operator.
21. Write a C program to perform addition and multiplication using bitwise operators.

Computer Organization

Subject Code: U24CSC22 **Credit: 4**
Contact Hours per week: 4 **Contact Hours per semester: 60**

Objectives:
Give in depth knowledge on architecture and operation of digital computers to understand the different functional units of the digital computer and how they co-ordinate together to carry out processing.

Course Outcomes:
CO 1: Know how the computer operations are specified with register transfer statements and how these are executed with clock pulses. Know about microprogramming and designing control unit
CO 2: Know about how registers communicate with the ALU, the operations of the memory stack, different instruction formats, addressing modes and RISC.
CO 3: Know about different arithmetic algorithms implement with digital hardware
CO 4: Know about different peripheral devices and how these devices communicate with each other

CO 5: know about different memory and its need, implementation

UNIT I **(12 Hours)**

Basics of Computer Organization: Instruction Codes – Computer Registers – Timing and Control –Hardwired control- Instruction Cycle – Micro programmed control – Control Memory – Address Sequencing

UNIT II **(12 Hours)**

Central Processing Unit: Introduction – General Register Organization – Stack Organization – Instruction Formats – Addressing Modes- Program control - RISC –CISC.

UNIT III **(12 Hours)**

Pipeline and Vector Processing: Parallel processing – Flynn's Classification – Pipelining – Arithmetic pipeline – Instruction pipeline – Vector processing – Array processing

UNIT IV **(12 Hours)**

Input – Output Organization: Peripheral devices - I/O Interface – Asynchronous Data Transfer – Modes of Transfer – Priority Interrupt – Direct Memory Access

UNIT V **(12 Hours)**

Memory organization: Memory Hierarchy – Main Memory – Auxiliary Memory - Associative Memory – Cache memory – Virtual memory.

TEXT BOOK:

- Computer System Architecture – M.Morris Mano & Rajib Mall, Pearson revised 3rd edition

UNIT I – Chapters - 6.1, 6.2, 6.3, 6.4, 6.5, 8.1, 8.2
UNIT II– Chapters -9.1, 9.2, 9.3, 9.4, 9.5, 9.7, 9.8
UNIT III– Chapters -10.1, 10.2, 10.3, 10.4, 10.6, 10.7
UNIT IV– Chapters -12.1, 12.2, 12.3, 12.4, 12.5, 12.6
UNIT V– Chapters -13.1, 13.2, 13.3, 13.4, 13.5, 13.6

REFERENCE BOOK:

- Computer Organization and Architecture, William Stallings, Pearson 7th Edition.

ALLIED - MATHEMATICAL FOUNDATIONS II

Contact Hours per week : 4 **Subject Code: U24MAAC21**
Contact Hours per semester: 60 **Credit: 4**

Course Outcomes:
On the successful completion of the course, students will be able to

CO1:	Extract various parameters like mean, median, mode and Standard deviation etc., according to the types of sampled data.
CO2:	Pick out more stabled data among various observations by analyzing the factors like standard deviation and coefficient of variance.
CO3:	Take decisions based on the correlation coefficient between two or more variables.
CO4:	Understand the concept of random variables, probability density function, expectation
CO5:	of a random variable and essential properties.
CO6:	Apply statistical tools like t-test, F-test for large sample and and χ^2-testforsmall samples to get more detail about the population from the samples. Interpret and demonstrate the behavior / happening of the population under consideration

Objectives:

- To enable the computational ability of the students on finding various parameters pertaining to the sampled data.
- To give enough awareness about some special types of probability distributions and applications of statistics in the real life.
- Give them sufficient tools of statistics and sampling techniques to handle the real life problems that are mainly of random nature and reach sound conclusions about them.

UNITI: (12 Hours)

Central tendencies: Introduction – Arithmetic mean – Partition values (Median only) - Mode-**Measures of Dispersion:** Introduction- Mean deviation, standard deviation, variance-Coefficient of variation.

UNITII: (12 Hours)

Correlation: Karl Pearson's coefficient of correlation, Covariance between $x$and y, Rank correlation. **Random variables:** Introduction-Random variables-Discrete random variable-Continuous random variable

UNITIII: (12 Hours)

Random variables (continued)**:** Mathematical Expectations (discrete random variable) – Mathematical expectation of continuous random variable - **Some Special Distributions:** Introduction-Binomial Distribution-Poisson Distribution-Normal Distribution.

UNITIV: (12 Hours)

Tests of Significance (Large Samples): Introduction-Sampling-Sampling distribution-Testing of hypothesis-Procedure for testing of hypothesis for large samples-Tests of significance for large samples: II. Test for Means, III. Test for standard deviation (exclude the portions I. Tests for proportion and percentage and IV. Test for correlation and related problems)

UNITV: (12 Hours)

Tests of significance (Small Samples): Introduction-Test of significance based on t-distribution (t-test)-Test of significance based on F-test-χ^2-test to test the goodness of fit.

Text Book:

- **S.Arumugam and A.Thangapand iissac**, Statistics, NewGamma Publication house, June 2007

Unit I: Chapter2: Sections 2.0,2.1,2.2(medianonly),2.3;Chapter3:Sections3.0,3.1 (excluding range, quartile deviation and root mean square deviation)
Unit II: Chapter 6: Sections 6.0, 6.1, 6.2;Chapter12: Sections12.0,12.1, 12.2, 12.3.
Unit III: Chapter 12: Sections12.4; Chapter13:Sections 13.0, 13.1,13.2 and 13.3.
Unit IV: Chapter14: Sections14.0,14.1,14.2,14.3,14.4,14.5(excludetheportionsI.Tests for proportion and percentage and IV. Test for correlation and related problems).
Unit V: Chapter15: Sections 15.0,15.1,15.2,Chapter16: Section16.2.

Reference Book:

1. **S.P.Gupta**, Statistical Methods, Sultan chand and sons,2004.
2. **R.S.N.Pillai and Bagavathi,** Statistics-Theory and Practice, 8th Edition, SChand& Co Ltd, 2016.
3. **H.Mulholland and C.R.Jones,** FundamentalsofStatistics,Springer Science+Business Media, New York, 1968.
4. **Allan G.Bluman,** Elementary Statistics: A step by step approach,7thEdition, McGraw-Hill Higher Education, Singapore, 2009.
