West Bengal State Council of Technical & Vocational Education and Skill Development
(Technical Education Division)

Syllabus
of

Diploma in Computer Science and Technology [CST], Computer Science and Engineering [CSE], Computer Software Technology [CSWT] & Information Technology [IT]

Part-II (3rd Semester)

Revised 2022

Detailed Syllabus for 3rd Semester

Computer Science and Technology, Computer Science and Engineering, Computer Software Technology & Information Technology

Sl. No	Category	Code No.	Course Title	Hours per week			Total contact hrs/ week	Credits
				L	T	P		
1.	Program core course	CST201	Computer Programming	2	0	0	2	2
2.	Program core course	CST203	Scripting Languages (Python)	2	0	0	2	2
3.	Program core course	CST205	Data Structures	2	0	0	2	2
4.	Program core course	CST207	Computer System Organization	3	1	0	4	4
5.	Program core course	CST209	Algorithms	3	1	0	4	4
6.	Summer Internship-I (4 weeks) after 2nd Sem	SI201	Summer Intern-ship-1					2
7.	Program core course	COPC211	Computer Programming Lab	0	0	4	4	2
8.	Program core course	CST213	Scripting Languages Lab	0	0	4	4	2
9.	Program core course	CST215	Data Structures Lab	0	0	2	2	1
	Total Credits							21

Course Title: Computer Programming in C	
Course Code	CST201
Number of Credits :2	2 (L: 2, T: 0, P: 0)
Prerequisites	Ability to develop logic / flow of simple problem.
Course Category	PC
Course code: CST	**Semester: THIRD**
Duration: 15 weeks	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory: 3 hrs/week Total Contact Hours: 45 Hours	Continuous Internal Assessment: 20 Marks Attendance: 10 Marks Viva/Presentation/Assignment/Quiz etc: 10 Marks End Semester Examination: 60 Marks

Aim of the Course

- **To study the structure programming concept.**
- **To study Linear Data Structure.**
- **To study Looping and Branching.**
- **To study subscripted variables and user defined data types.**
- **To study user defined functions.**
- **To study pointers in depth.**

Course Objectives

To enable student, develop structured solutions to problems and implementing them using computers. This involves two parts:

- **Formulating a solution for a given problem as a well-defined sequence of actions, and**
- **Expressing solution in a machine-readable form or a programming language.**

For the second part, we will learn the common units of programming languages. The first part can only be learned through the repeated practice of solving problems.

Course Content:

Contents (Theory)	Hrs./Unit	Module
UNIT 1: Basics of C	5	A

- History of C, Advantages of Structured Program, Files (source, header, object, binary executable) used in C, Characteristics of C.
- C character set, Tokens, Constants, Variables, Keywords, Data types used in C.
- C operators (arithmetic, logical, assignment, relational, unary, binary, increment and decrement, conditional, ternary, bit wise, special, comma, sizeof, postfix, prefix etc.), Operator precedence, Associativity of operators, Type conversion, Typecasting.
- Formatted input, Formatted output.

UNIT 2: Decision Control and Looping Statements	**5**	A

- Decision making and branching statements, if statement (if, if-else, else-if ladder, nested if-else), Switch case statement & applications.
- Conditional and unconditional ‘goto’ statement and drawbacks.
- Iterative/Loop statement, Entry controlled & exit controlled loop structure & differences, Example like while, do- while, and for loop structure, Break and continue statement & their uses, nested loop structure & applications.

UNIT 3: Subscripted Variables / Arrays	10	B

- Advantages of subscripted variables/ arrays & accessing array elements, Declaration and initialization of one dimensional, two dimensional, multidimensional (idea only) and character arrays & Strings, Accessing array elements.
- Declaration and initialization of string variables, String handling functions from standard library (strlen(), strcpy(), strcat(), strcmp()), Applications like string operations to extract substring from left, right, middle of a string, Replacement of string characters, concatenation of two strings etc.

UNIT 4: User defined functions	10	C

- Definition of functions, advantages of functions in modular approach problem solving, Prototype declaration, Scope and lifetime of variables & Storage Class (Auto, Extern, Static, Register), defining functions, function signature, passing parameter types, Function call (call by value, call by reference), Return values.
- Recursion and use of memory stack, Types of recursion. Recursion vs Iteration. Applications.

UNIT 5: Pointers in C	15	C

- Understanding pointers, difference between memory variables and pointer variables, Declaring and accessing pointers, constant pointers and pointer to a constant, Null Pointers, Generic Pointers, Pointers arithmetic and expressions.
- Passing arguments to function using pointers, Pointers and arrays, Passing an array to a function, Array name and Pointer.
- Pointers and Strings, Array of pointers, Function pointers, Pointer to a pointer.
- Dynamic memory allocation using malloc(), calloc(), realloc(). Uses of free(). Pointer to a structure.

Course outcomes

- **Student should be able to computationally formulate basic problems and write code to execute them.**
- **The focus of the course as mentioned above should be on example-based learning.**

Reference Books

1. Programming in C, Author: Reema Thareja, OXFORD University Press
2. Computer Fundamentals and Programming In C, Author Anita Goel & Ajay Mittal, Pearson
3. Head first C, Author by Griffiths, SPD publication
4. C: Concepts & Programming, Author: Bhusry, Willy
5. Programming in C, Author: E. Balagurusamy, TMH
6. Mastering C, Author: Venugopal, TMH
7. C programming: Test your skills, Author: Kamthane Ashok, Pearson Education India

Course Title : Scripting Languages	
Course Code	CST203
Number of Credits :2	2(L: 2, T: 0, P: 0)
Prerequisites	NIL
Course Category	PC
Course code : CST	Semester : THIRD
Duration : 15 weeks	Maximum Marks : 100
Teaching Scheme	Examination Scheme
Theory : - 2 hrs/week	Continuous Internal Assessment : 20 Marks
Tutorial: - 0	Attendance-10 Marks Viva/Presentation/Assignment /Quiz etc : - 10 Marks
Total Contact Hoirs:30 Hours	
Practical : NIL	End Semester Examination : 60 Marks
Aim:	Learn Scripting Language features and programming

Course Objectives:

- **To learn how to work with a scripting language.**
- **To introduce Python programming language through its core language basics and program design techniques suitable for modern applications.**
- **To understand the wide range of programming facilities available in Python covering graphics, GUI, data visualization.**
- **To utilize high-performance programming constructs available in Python to strengthen applications and development in practical scenarios.**

Course Content:

Contents (Theory)	Hrs./Unit	Module
UNIT 1: Introduction, Variables and Data Types	**7**	**A**

- **History, Features, setting up path, Installation and Working with Python, Basic Syntax**
- **Understanding Python variables**
- **Numeric data types: int, Long int, float, complex/imaginary**
- **String data type and string operations: String literals, manipulating strings, comparing strings, Unicode string literals, converting between Simple Types, Converting to strings, String Formatting, String Methods.**
- **Date and Time**
- **Other Data Types:**
  - **Tuples**
  - **List: Defining list, list slicing, Split, Join, Manipulating Lists, Copying Lists**
  - **Dictionary**
  - **Arrays**
- **Basic Operators: Arithmetic Operators, Relational Operators, Assignment Operators (Simple assignment and Multiple Assignment-Tuple packing & unpacking) Logical Operators, Bitwise Operators, Membership Operators, Identity Operators**

- **Operator Precedence**
- **Understanding coding blocks**

UNIT 2: Control Structures	**5**	**A**

- **Conditional blocks using if, else and elif**
- **For loops and iterations**
- **while loops**
- **Loop manipulation using continue, break and else and pass**
- **Programming using conditional and loops block**
- **Modify loops : break and continue**

UNIT 3: Functions, Modules and Packages	**6**	**B**

Organizing codes using functions

- **Defining Functions & Calling Functions**
- **Pass by object reference**
- **Parameters**
- **Arbitrary arguments**
- **Optional and Named Arguments**
- **Passing arguments from a tuple**
- **Variable Scope and Binding: Local Variables, Nonlocal Variables, Global Variables, class scope**

Organizing projects into modules: Grouping Code with Modules
Importing own module as well as external modules
Understanding Packages: Grouping Modules into Packages

UNIT 4: : File I/O, Text Processing, Regular Expressions	**6**	**C**

Accessing Keyboard Input: raw_input and input
Printing to the Screen: print
File modes and permissions
read functions: read(), readline(), readlines()
write functions :write(),writelines()
other file operations: open(),close(),tell(),seek(),flush(),fileno(), isatty(),next()
Redirecting output streams to files.
Programming using file operations
Powerful pattern matching and searching: re.match(),re.searh(),re.findall(), re.finditer()
Creating and Using Regular Expression Objects- import re, re.compile(),re.sub(), re.subn(), re.split()
Power of pattern searching using regex

UNIT 5: **Frameworks**	**6**	**C**

Frameworks - The MVC framework, Django- What is Django and why should you use Django?

Creating URL, Templates

• Send data to a template
• Display data in a template
• Display object lists in a template

- **Handle chains with filters in Django**
- **Use URLs effectively**
- **Create base templates in order to extend other templates**
- **Insert static files in our templates**

Django Form

- **Create an HTML form**
- **Handle the data sent by a form**
- **Create a Django form**
- **Validate and manipulate data sent from a Django form**
- **Create forms based on models**
- **Customize error messages and usage of widget**

Reference Books

1. **Taming Python by Programming, Jeeva Jose, Khanna Publishing House**
2. **Core Python Programming, Dr. R. Nageswara Rao, Dreamtech Press**
3. **Python Programming: Using Problem Solving Approach, Reema Thareja, Oxford University Press**
4. **Core Python Programming, Wesley J. Chun, Pearson**
5. **Python for programming, P. Deitel, H. Deitel, Pearson**
6. **Starting Out with Python, Tony Gaddis, Pearson**
7. **Introduction to Computation and Programming Using Python. John V. Guttag, MIT Press.**
8. **An Introduction to Python, G.v. Rossum, SPD**
9. **Practical Programming: An Introduction to Computer Science using Python 3, Paul Gries, The Pragmatic Bookshelf**

Course outcomes: **At the end of the course student will be able to build program with a scripting language and will be able to learn any other scripting language on their own.**
Design real life situational problems and think creatively about solutions of them. Apply a solution clearly and accurately in a program using Python. Apply the best features of Python to program real life problems.

Course Title: Data Structures	
Course Code	CST205
Number of Credits :2	2 (L: 3, T: 0, P: 0)
Prerequisites	Basic Knowledge of Computer system
Course Category	Computer Science and Technology
Course code: CST	**Semester: THIRD**
Duration: 15 weeks	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory: 3 hrs/week **Total Contact Hours: 45 Hours**	Continuous Internal Assessment: 20 Marks Attendance: 10 Marks Viva/Presentation/Assignment/Quiz etc: 10 Marks End Semester Examination: 60 Marks

Aim of the Course

- **To study the structure programming concept.**
- **To study Linear Data Structure.**
- **To study Dynamic Memory Allocation.**
- **To study Non-Linear Data Structure.**
- **To study user defined Graf theory.**
- **To study Recursion in depth.**

Course Objectives

To enable student, develop structured solutions to problems and implementing them using computers. This involves two parts:

- **Formulating a solution for a given problem as a well-defined sequence of actions, and**
- **Expressing solution in a machine-readable form or a programming language.**

To provide strong foundation for implementing programming language to formulate, analyze and develop solutions related to various data structures problems.

Course Content:

Contents (Theory)	Hrs./Unit	Module
UNIT 1: Introduction to Data Structures	3	A

- ✓ **Basic Terminology**
- ✓ **Classification of Data Structures**
- ✓ **Operations on Data Structures.**

UNIT 2: Linear Data Structures	**12**	A

- ✓ **Stacks:**
  - **Introduction to Stacks**
  - **Array Representation of Stacks**
  - **Operations on a Stack**
  - **Applications of Stacks-Infix-to-Postfix Transformation**
  - **evaluating Postfix Expressions.**
- ✓ **Queues:**
  - **Introduction to Queues**
  - **Array Representation of Queues**
  - **Operations on a Queue**
  - **Types of Queues-De-Queue**
  - **Circular Queue**
  - **Applications of Queues-Round Robin Algorithm.**
- ✓ **Recursion (GCD, Tower of Hanoi Problem)**

UNIT 3: Linked Lists	15	B

- ✓ **Singly Linked List**
  - **Representation in Memory,**
  - **Operations on a Single Linked (add new node- first, in-between, end position. Delete-- first, in-between, end position),**
- ✓ **Circular Linked Lists**
  - **Operations on a circular Single Linked (add new node- first, in-between, end position.**
  - **Delete-- first, in-between, end position),**
- ✓ **Doubly Linked Lists**

- **Operations on a Double Linked (add new node- first, in-between, end position. Delete-- first, in-between, end position),**

✓ **Circular Linked Lists**

- **Operations on a circular Double Linked (add new node- first, in-between, end position.**
- **Delete-- first, in-between, end position),**

✓ **Linked List Representation**

- **Operations of Stack,**
- **Operations of Queue.**

UNIT 4: Non-Linear Data Structures	15	C

✓ **Trees:**

- **Basic Terminologies**
- **Definition and Concepts of Binary Trees**
- **Representations of a Binary Tree using Arrays and Linked Lists**
- **Operations on a Binary Tree-Insertion, Deletion**
- **Traversals, Types of Binary Trees**
- **B-Tree**
- **AVL Tree**

✓ **GRAPHS:**

- **Graph Terminologies**
- **Representation of Graphs- Set, Linked**
- **Matrix**
- **Graph Traversals**
- **BFS and DFS**

Course outcomes

- **Student should be able to computationally formulate basic problems and write code to execute them and have a good understanding of Data Structures and its applications in algorithms**
- **The focus of the course as mentioned above should be on example-based learning.**

Reference Books

1. Introduction to Data Structures in C, Kamthane, Pearson
2. Data Structures Using C, Reema Thareja, Oxford University Press India.
3. Data Structures, Lipschutz Seymour, McGraw-Hill Education
4. A simplified approach to data structures, Pawan Goyal, Published by SPD
5. Data Structures Using C, 1e, Tenenbaum, Pearson
6. Data Structures and Algorithms, Aho, Pearson

Course Title : Computer System Organization	
Course Code	CST207
Number of Credits :4	4 (L: 3, T: 1, P: 0)
Prerequisites	Basic Knowledge of Computer system and Digital gates
Course Category	Computer Science
Course code : CST	Semester : THIRD
Duration : 15 weeks	Maximum Marks : 100
Teaching Scheme	Theory : - 3 hrs/week, Tutorial: - 1 hr/week
Examination Scheme	Continuous Internal Assessment : 20 Marks
	Attendance-10 Marks Viva/Presentation/Assignment /Quiz etc : - 10 Marks
Total Contact Hoirs:60 Hours	
Practical : NIL	End Semester Examination : 60 Marks
Aim:	Develop the concept of how computer works internally.

Course Learning Objectives: To have a thorough understanding of the basic structure and operation of a digital computer, it's Architecture, computational designs and how computer works.

Course Content:

Contents (Theory)	Hrs./Unit	Module
UNIT 1:	6	A

Structure of Computers:

- **Computer Functional units, Von-Neumann architecture, Bus structures, Basic Operational Concepts, Data representation (Fixed and Floating point), Error detecting codes.**

Register Transfer and Micro Operations:

- **Register transfer, Memory transfers, Arithmetic micro-operations, Logic micro-operations, Shift micro-operations, and Arithmetic logic shift unit.**

UNIT 2:	**20**	A

Micro Programmed Control:

- **Control memory, Address sequencing, and design of control unit.**

Computer Arithmetic:

- **Addition and Subtraction, Multiplication and Division algorithms, Float- ing-point arithmetic operation,**

Pipelining:

- **Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline**
- **Vector Processing, Array Processors.**

UNIT 3:	**10**	B

Introduction to Microprocessor Architecture:

- **Instruction Set, Architecture design principles from programmer's perspective.**
- **One example microprocessor (Intel 8086), Block diagram, Pin functions, Register structure, Segmentation, Interrupt mechanism, Addressing modes, Instructions.**

UNIT 4:	**9**	B

Assembly Language Programming:

- **Simple programs, Assembly language programs involving logical, branch and call instructions,**
- **sorting, evaluation of arithmetic expressions, string manipulation,**
- **assembler directives, procedures and macros.**

UNIT 5:	**15**	C

Memory and Digital Interfacing:

- **addressing and address decoding, interfacing RAM, ROM, EPROM,**
- **programmable peripheral interface,**
- **Cache Memory (Mapping and Hit ratio),Virtual Memory Technique (Logical address, Physical address, TLB etc.)**

Reference Books

8. **Computer Organization, Carl Hamacher, Zvonks Vranesic, SafeaZaky, McGraw-Hill**
9. **Computer Organization and architecture, William Stallings, Pearson**
10. **Microprocessors Interface, Douglas V.Hall, Tata McGraw-Hill.**
11. **Microprocessors and Microcontrollers, Senthil Kumar, Saravanan, Jeevananthan, Oxford**
12. **Computer System Architecture, M. Moris Mano, Pearson/PHI, India.**
13. **The X86 Microprocessors - Architecture and Programming and Interfacing, Das, Pearson**
14. Computer organization design and Architecture 5th edition, Shiva (SPD/CRC press)
15. **Advanced Microprocessors and Peripherals- Architecture, Programming and interfacing,A.K.Ray, K.M.Bhurchandi, Tata McGraw-Hill, New Delhi, India.**
16. **Computer Organization and Design: A Hardwar/Software Interface (MIPS Edition) by Patterson and Hennessy**

Course outcomes:

- **Have a good understanding of functioning of computer system as such and its various subcomponents.**
- **Student will be able to understand computing requirement for a specific purpose,**
- **Analyseperformance bottlenecks of the computing device and choose appropriate computing device fora given use case.**
- **Acquire Knowledge to write assembly language programs**

Course Title : Algorithms	
Course Code	CST209
Number of Credits :4	4 (L: 3, T: 1, P: 0)
Prerequisites	NIL
Course Category	PC
Course code : CST	Semester : THIRD
Duration : 15 weeks	Maximum Marks : 100
Teaching Scheme	Examination Scheme
Theory : - 3 hrs/week	Continuous Internal Assessment : 20 Marks
Tutorial: - 1 hr/week	Attendance-10 Marks Viva/Presentation/Assignment /Quiz etc : - 10 Marks
Total Contact Hoirs:60 Hours	
Practical : NIL	End Semester Examination : 60 Marks
Aim:	Develop basic concept of Algorithms in Computer Science

Course Objectives: The objective of this course is to prepare the student with the algorithmic foundations of computing. A sound grasp of algorithms is essential for any computer science engineer. Almost all programming involves algorithms at some level.

Course Content:

Contents (Theory)	Hrs./Unit	Module
UNIT 1: Fundamentals of Algorithms	6	A

- **Definitions and Characteristics of Algorithm. Examples.**
- **Data Abstraction.**
- **Sets, Multisets,**
- **Stacks, Queues.**
- **Asymptotic Notations (*Order Notation*, Omega Notation, Theta Notation) with Examples.**
- **Time and Space Complexity. Best Average and worst-case analysis of algorithms.**
- **Programming Models Concepts: -Divide and Conquer, Greedy Methods, Dynamic Programming,**

UNIT 2: Sorting	**16**	A

The sorting problem.

- **Bubble sort.**
- **Selection sort.**
- **Insertion sort.**

- **Shell sort**
- **Merge sort.**
- **Quicksort.**
- **Heapsort.**

Computation of Best Average and worst-case Time complexity of all the above sorting algorithms.

Linear Time sorting

- **Count Sort**
- **Bucket Sort**
- **Radix Sort**

UNIT 3: Searching	**16**	B

Linear Search Algorithm.
Binary Search Algorithm.
Computation of Best, Average and Worst-case Time complexity of Linear and Binary Search

Binary Search Trees: Algorithms, Searching Time & space complexity.

Balanced Search Trees: What is the signification and advantage of height balancing?
Insertion, Deletion and Searching Algorithms of different types of Balanced Search Trees and their comparative study.

Hashing, Hash Tables Hash functions, Collision and Collision resolving techniques.

Symbol Tables

UNIT 4:	**16**	C

Definitions: Graph Directed and Undirected graph-Examples.
Paths, Cycles, Spanning trees-Examples.
Directed Acyclic Graphs-Examples.

Topological Sorting.

Minimum Spanning Tree algorithms:

- **Prim's Algorithm with Examples**
- **Kruskal's Algorithm with Examples**

Shortest Path algorithms:

- ✓ **Dijkstra's algorithm.**
- ✓ **Bellman–Ford Algorithm,**

Floyd-Warshall all pairs shortest path algorithm

UNIT 5: **Strings**	**6**	C

- **String Sort.**
- **Tries.**
- **Search a Substring within a string.**

- **String Matching Algorithms and their complexity analysis**
  - ✓ **Simple/Naive String Matching Algorithm**
  - ✓ **Rabin-Karp Algorithm**
  - ✓ **Knuth-Moris-Pratt Algorithm**
  - ✓ **Horspool String Matching Algorithm**
  - ✓ **Boyer-Moore String Matching Algorithm**
- **Regular Expressions.**
- **Elementary Data compression.**

Reference Books

1. **Algorithm Design, Jon Kleinberg| Eva Tardos, Pearson**
2. **Introduction to Algorithms, Cormen, Leiserson, Rivest and Stein. MIT Press**
3. **Design and Analysis of Algorithms, S.Sridhar, Oxford University Press.**
4. **Fundamentals of Computer Algorithms, E. Horowitz, S. Sahani, S. Rajasekaran, Galgotia.**
5. **Introduction to the Design and Analysis of Algorithms, Anany Levitin, Pearson**
6. **Algorithms in a Nutshell, G.T.Heinemam, SPD**
7. **Design & Analysis of Algorithms, Gajendra Sharma, Khanna Publishing House**

Course outcomes:

- **The student should be able to understand the basic notions of time and space complexity of algorithms.**
- **The student should be able to design basic algorithms for sorting and searching.**
- **The student should be able to realize Graph concepts, Minimum spanning Tree algorithms and shortest path algorithms.**
- **The student should be able to implement sorting, searching, tree and graph algorithms in a modern computer programming language**
- **The student should be able to understand String sort and how to search a substring within a string.**

Examination Scheme of ESE (End Semester Examination)				
	Question Type	**Question to be set**	**Questions to be answered**	**Marks**
Theoretical	**MCQ type questions carrying 1 mark.**	**15**	**10**	**10**
	Fill in the blanks type questions carrying 1 mark.	**15**	**10**	**10**
	Short answer type questions carrying 1 mark.	**15**	**10**	**10**
	Subjective type questions carrying 2 marks.	**10**	**6**	**12**
	Subjective type questions carrying 6 marks.	**9 (3 each from each of 3 modules)**	**3**	**18**
TOTAL				**60**

Course Title: Computer Programming Lab in C	
Course Code	CST211
Number of Credits :2	3 (L: 0, T: 0, P: 3)
Prerequisites	Basic Operations on Computer
Course Category	PC
Course Code: CST	**Semester: THIRD**
Duration: 15 weeks	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Laboratory: 3 hrs/week Total Contact Hours: 45 Hours	Continuous Internal Assessment: 60Marks External Assessment: 40 Marks

Course Objectives:

This Lab course is intended to practice what is taught in theory class of 'Computer Programming' and become proficient in computer programming. Computer programming is all about regular practice. Students should work on solved and unsolved problems listed in the text books, and the problems given by the teacher. Some of the topics that should necessary be covered in lab are listed below.

Course Content:

Sr. No.	Topics for Practice	
01	Familiarization with programming environment (Editor, Compiler, etc.)	Skill Area
02	a) Displaying hexadecimal, decimal, octal number format of the entered numbers. b) Displaying entered number with leading zeros and trailing zeros. c) Displaying entered number with right and left justification. d) Displaying with different formatting specifiers.	Programs using Formatted input and output.
03	a) To find greatest / smallest of three numbers. b) To display pass class, second-class, distinction according to the marks entered from the keyboard. c) To find even or odd numbers. d) To display spellings of number 1-10 on entry. e) Implementation and displaying the menu to execute 1. ADD, 2. SUBTRACT 3. MULTIPLICATION, 4. DIVISION using switch case. f) To check whether there exist real (float) roots of a quadratic equation and if exist find them.	Programs using decision making statements and branching statements
04	a) To display our College name twenty times on screen. b) To demonstrate Continue and Break statements within loop structure. c) To add first 'n' natural, even, odd numbers using different loop structures. d) To find GCD, LCM of two integral numbers. e) To generate simple number triangle for n rows. f) To generate Pascal triangle for n rows. g) To add the series 1 + (1 + 2) + (1 + 2 + 3) + …+ (1 + 2 + 3 + … + n) h) To generate all prime numbers within the given range.	Programs using loop statements

	i) To find all the Armstrong numbers within 100 to 1000.	
05	a) To find the largest and smallest numbers from array elements. b) To sort array elements in ascending / descending order. c) To enter elements for 3X3 matrix and display them. d) To calculate addition / subtraction of 2-dimensional matrix. e) To calculate multiplication of 2-dimensional matrix. f) To find the number of vowels and consonants in a string. g) Implementation of strlen(), strcpy(), strcat() and strcmp() functions. h) To check whether a string is palindrome or not. i) To replace a specific character/string by another character/string in a multiword string. j) To make the abbreviated form of a multiword string.	Programs to demonstrate applications of 1 & multi-dimensional arrays & Strings
06	a) To calculate the value of ${}^{n}C_{r}$, $n \geq r$ using function b) To find the sum of the series $1 + \frac{x}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}$ for $n \geq 1$, $x \geq 0$ using function. c) To interchange the biggest and smallest number in to calculate factorial a one-dimensional array using function. d) To calculate addition, subtraction and multiplication of 2-dimensional matrix using function. e) Write a program in C to find GCD of two numbers using recursion. f) To calculate factorial of any given number using recursion. g) To demonstrate call by reference, call by value.	Programs to demonstrate parameter passing mechanism & recursion.
07	a) To read and display an integer array using pointer. b) To read and display a text using a character pointer to a string. Also count the number of characters, words and lines in the text. c) To read, display, add and subtract of two times defined using hour, minutes and values of seconds. d) To read and display the contents of a structure variable using pointer to a structure. e) Write a program in C to create a singly linked list of n nodes and display it in reverse order. f) Write a program in C to insert a new node to a Singly Linked List after a desired node and display the list. g) Write a program in C to delete a node from a Singly Linked List after/ before a desired node and display the list. h) Implement Stack and Queue data structure using dynamic memory allocation.	Programs to demonstrate use of pointers and dynamic memory allocation.

Course outcomes

- Use of programming language constructs in program implementation.
- Apply different logics to solve given problem.
- Write program using different implementations for the same problem.
- Identify different types of errors as syntax, semantic, fatal, linker & logical.
- Debugging of programs.
- Understanding different steps and stages to develop complex program.

Reference Books

1. Programming in C, Author: Reema Thareja, OXFORD University Press
2. Programming in C, Author: E. Balagurusamy, TMH
3. C in Depth, Author: Srivastava, BPB
4. Mastering C, Author: Venugopal, TMH
5. C programming: Test your skills, Author: Kamthane Ashok, Pearson Education India
6. Head First C, David Griffiths, SPD

Course Title: Scripting Languages Lab	
Course Code	CST213
Number of Credits :2	2(L: 0, T: 0, P: 4)
Prerequisites	NIL
Course Category	PC
Course code : CST	Semester : THIRD
Duration : 15 weeks	Maximum Marks : 100
Teaching Scheme	**Examination Scheme** **1.Continuous Internal Assessment :60 Marks** This 60 Marks will be comprised of the following Marks division: • **Assignment: 20 Marks** • **Class Performance: 20 Marks** • **Viva Voce: 10 Marks** • **Attendance: 10 Marks** **2.End Semester Examination (External Assessment/ Sessional) :40 Marks**
Practical: - 4 hrs/week	

Course Objectives: This Lab course is intended to practice whatever is taught in theory class of Scripting Languages' and become proficient in scripting. Computer programming is all about regular practice. Students should work on solved and unsolved problems listed in the text books, and the problems given by the teacher. Some of the topics that should necessary be covered in lab are listed below.

Course Content:

Contents (Practical)	
Installing Python in Windows/Linux/Ubantu/Mac OS	
Sr. No.	**Topics for Practice**

S.No.	Topics for Practice
1	Practice basic coding syntax
2	Write and execute scripts based on data types
3	Write and execute Python scripts with conditionals and loops
4	Write and execute Scripts based on Functions and Modules
5	File Processing scripts
6	Write and execute Regular Expressions
7	Develop a simple web application

Some of the suggested sample programs are:

- **Running instructions in Interactive interpreter and a Python Script**
- **Write a script to purposefully raise Indentation Error and Correct it**
- **Write a script to find Sum and average of first n natural numbers**
- **Given 2 strings, s1 and s2, create a new string by appending s2 in the middle of s1**
- **Write a script to check whether a given string is palindrome or not.**
- **Write a program add.py that takes 2 numbers as command line arguments and prints its sum.**
- **Write a script using a for loop that loops over a sequence**
- **Write a script to count the numbers of characters in the string and store them in a dictionary data structure**
- **Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.**
- **Write a script that combines more than one lists into a dictionary**
- **Compute the GCD & LCM of two numbers.**
- **Check a number is prime or not**
- **Find the square root of a number**
- **Exponentiation (power of a number)**
- **Find all primes within a given range**
- **Find First n Fibonacci numbers.**
- **Find the maximum of a list of numbers**
- **Linear search and Binary search**
- **Write a function nearly equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.**
- **Find the most frequent words in a text read from a file**
- **Programs that take command line arguments (word count)**
- **Write a function to find all duplicates in the list**
- **Remove empty strings from the list of strings**
- **Write a program to print each line of a file in reverse order.**
- **Write a program to compute the number of characters, words and lines in a file.**
- **Write a script to Calculate age in year month days of a person taking his/her date of birth as input and accessing current system date.**
- **Write a regular expression to search digit inside a string**

Finding Whether PIP is installed, Install virtual environment using pip, Installing Django, Setting PATH , Creating and Deploying Applications Using Django

Some of the suggested sample Exercises using Django

- **Send data to a template**
- **Display data in a template**
- **Display object lists in a template**
- **Handle chains with filters in Django**
- **Create and Use URLs in Django**
- **Create base templates in order to extend other templates**
- **Insert static files in our templates**
- **Create an HTML form**
- **Handle the data sent by a form**
- **Create a Django form**
- **Validate and manipulate data sent from a Django form**
- **Create forms based on models**
- **Customize error messages and usage of widget**

Course outcomes

Course outcomes: At the end of the course student will be able to build program with a scripting language and will be able to learn any other scripting language on their own.

Reference Books

1. **Taming Python by Programming, Jeeva Jose, Khanna Publishing House**
2. **Python Programming: Using Problem Solving Approach, Reema Thareja, Oxford University**
3. **Starting Out with Python, Tony Gaddis, Pearson**
4. **Core Python Programming, Wesley J. Chun, Prentice Hall**
5. **Introduction to Computation and Programming Using Python. John V. Guttag, MIT Press.**
6. **Beginning Python using Python 2.6 and Python 3, James Payne, Wrox publishing**
7. **Practical Programming: An Introduction to Computer Science using Python 3, Paul Gries, The Pragmatic Bookshelf**

Course Title: Data Structures Lab	
Course Code	CST215
Number of Credits :2	2 (L: 0, T: 0, P: 4)
Prerequisites	Basic Knowledge of Computer system
Course Category	Computer Science and Technology
Course Code: CST	**Semester: THIRD**
Duration: 15 weeks	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Laboratory: 4 hrs/week Total Contact Hours: 60 Hours	Continuous Internal Assessment: 40 Marks Attendance, Assignment & Quiz: - 20 Marks End Semester Examination: 40 Marks

Course Objectives:

This Lab course is intended to practice what is taught in theory class of 'data Structure' and become proficient in computer programming. Computer programming is all about regular practice. Students should work on solved and unsolved problems listed in the text books, and the problems given by the teacher. Some of the topics that should necessary be covered in lab are listed below.

Course Content:

01	**Skills to be developed**	
	➢ Intellectual Skills: ➢ Use of programming language constructs in program implementation. ➢ To be able to apply different logics to solve given problem. ➢ To be able to write program using different implementations for the same problem ➢ Study different types of errors as syntax semantic, fatal, linker & logical ➢ Debugging of programs ➢ Understanding different steps to develop program such as ➢ Problem definition ➢ Analysis ➢ Design of logic ➢ Coding ➢ Testing ➢ Maintenance (Modifications, error corrections, making changes etc.)	Programs using Formatted input and output.
02	g) Motor Skills: Proper handling of Computer System.	

List of Practical:

01	Programs based on: Array operations, insertion, deletion
02	Programs based on Stacks Implementation of PUSH & POP operations, Evaluate postfix expressions, Infix to postfix h) conversions. Programs to demonstrate parameter passing mechanism & recursion.
03	Recursive programs: factorial, Fibonacci, Ackerman function, and tower of Hanoi.(any two)
04	Programs for demonstrating queue operations. one recursive program converted to non-recursive ones
05	Programs based on Linked lists
06	Programs based on trees Creating a binary tree, in order, pre order and post order traversal of binary tree, deleting a node from binary tree.
07	Programs for implementing various sorting techniques. (Minimum three sorting techniques from topics mentioned in the syllabus))
08	Programs for implementing various sorting and searching techniques. (Minimum two searching techniques from topics mentioned in the syllabus.)
09	Assignments based on graph theory.
10	Assignments based on graph theory.

LIST OF SAMPLE PROBLEMS FOR DATA STRUCTURE LAB (Example)

1. To write a program to check whether a word is palindrome or not.
2. To create a two-dimensional array of numbers and calculate & display the row & column sum and thegrand total.
3. To write a program of matrix multiplication.
4. To write a program to insert (Push) an element into the sack and delete (Pop) an element from the stackusing pointer.
5. To write a program to convert an infix expression to a postfix expression.
6. To evaluate a postfix expression.
7. To write a program to insert an element in the queue and delete an element from the queue using pointer.
8. To create a circular queue and add an element and delete an element from a circular queue.
9. To write a program of a structure containing an item name along with the unit price. The user enters the item name and quantity to be purchased. Program print outs total price of item with name using pointer ina structure or array in a structure.
10. To create a single linked list and — (a) insert a node in the list (before header node, in between two nodes, end of the list); (b0 delete a node from the list (1^{st} node, last node, in between two nodes); (c) Concatenate two lists.
11. To create a doubly linked list and — (a) insert a node in the list (before header node, in between two nodes, end of the list); (b) delete a node from the list (1^{st} node, last node, in between two nodes); (c) Concatenate two lists.
12. To create a circular linked list and insert & delete an element from the list.
13. Write a program to merge two shorted linked list.
14. Write a program to reverse a linked list.
15. To write a program to calculate the binomial co-efficient of ${}_nC^r$ of two numbers using recursive function.
    Also write the same program using function in non-recursive way.
16. To write a program to generate Fibonacci Series using recursive function. Also write the same programusing function in non-recursive way.
17. To write a program to create a binary tree and traverse it in pre-order and post-order form.
18. To write a program to create a binary search tree and —

    (a) insert a new node in the BST

    (b) search anode in the BST

    (c) delete a node from the BST

Course outcomes

- Using of programming language implement data Structure.
- Apply different logics to solve given problem.
- Write program using different implementations for the same problem.
- Identify different types of errors as syntax, semantic, fatal, linker & logical.
- Debugging of programs.
- Understanding different steps and stages to develop complex program.

Reference Books

1. Introduction to Data Structures in C, Kamthane, Pearson
2. Data Structures Using C, Reema Thareja, Oxford University Press India.
3. A simplified approach to data structures, Pawan Goyal, Published by SPD