

**B.Tech. in COMPUTER SCIENCE AND ENGINEERING**  
**Course Structure & Syllabus (R-25 Regulations)**

**Applicable from A.Y: 2025-2026 Batch**

**I Year I Semester**

S. No.	Course Code	Course Title	L	T	P	Credits
1.	25MA101BS	Matrices and Calculus	3	1	0	4
2.	25CH102BS	Engineering Chemistry	3	0	0	3
3.	25EN103HS	English for Skill Enhancement	3	0	0	3
4.	25EC104ES	Electronic Devices and Circuits	3	0	0	3
5.	25CS105ES	Programming for Problem Solving	3	0	0	3
6.	25CH106BS	Engineering Chemistry Lab	0	0	2	1
7.	25CS107ES	Programming for Problem Solving Lab	0	0	2	1
8.	25EN108HS	English Language and Communication Skills Lab	0	0	2	1
9.	25ME109ES	Engineering Workshop	0	0	2	1
10.		Induction Program				
<b>Total Credits</b>			<b>15</b>	<b>1</b>	<b>8</b>	<b>20</b>

**I Year II Semester**

S. No.	Course Code	Course Title	L	T	P	Credits
1.	25MA201BS	Ordinary Differential Equations and Vector Calculus	3	0	0	3
2.	25PH202BS	Advanced Engineering Physics	3	0	0	3
3.	25ME203ES	Engineering Drawing and Computer Aided Drafting	2	0	2	3
4.	25EE204ES	Basic Electrical Engineering	3	0	0	3
5.	25CS205ES	Data Structures	3	0	0	3
6.	25PH206BS	Advanced Engineering Physics Lab	0	0	2	1
7.	25CS207ES	Data Structures Lab	0	0	2	1
8.	25CS208ES	IT Workshop	0	0	2	1
9.	25EE209ES	Basic Electrical Engineering Lab	0	0	2	1
10.	25CS210ES	Python Programming Lab	0	0	2	1
<b>Total Credits</b>			<b>14</b>	<b>0</b>	<b>12</b>	<b>20</b>

**II YEAR I SEMESTER**

S. No.	Course Code	Course Title	L	T	P	Credits
1.	25CS301PC	Discrete Mathematics	3	0	0	3
2.	25CS302PC	Computer Organization and Architecture	3	0	0	3
3.	25CS303PC	Object Oriented Programming through java	3	0	0	3
4.	25CS304PC	Software Engineering	3	0	0	3
5.	25CS305PC	Database Management Systems	3	0	0	3
6.	25MS306HS	Innovation and Entrepreneurship	2	0	0	2
7.	25CS307PC	Object Oriented Programming through java Lab	0	0	2	1
8.	25CS308PC	Software Engineering Lab	0	0	2	1
9.	25CS309PC	Database Management Systems Lab	0	0	2	1
10.	25CS310SD	Skill Development Course	0	0	2	1
11.	25VA300ES	Environmental Science	1	0	0	1
<b>Total Credits</b>			<b>18</b>	<b>0</b>	<b>08</b>	<b>22</b>

**II YEAR II SEMESTER**

S. No.	Course Code	Course Title	L	T	P	Credits
1.	25MA401PC	Computer oriented Statistical Methods	3	0	0	3
2.	25CS402PC	Operating Systems	3	0	0	3
3.	25CS403PC	Algorithm design and Analysis	3	0	0	3
4.	25CS404PC	Computer Networks	3	0	0	3
5.	25CS405PC	Machine Learning	3	0	0	3
6.	25MA406BS	Computational Mathematics Lab	0	0	2	1
7.	25CS407PC	Operating Systems Lab	0	0	2	1
8.	25CS408PC	Computer Networks Lab	0	0	2	1
9.	25CS409PC	Machine Learning Lab	0	0	2	1
10.	25CS410SD	Skill Development Course	0	0	2	1
<b>Total Credits</b>			<b>15</b>	<b>0</b>	<b>10</b>	<b>20</b>

**III YEAR I SEMESTER**

S. No.	Course Code	Course Title	L	T	P	Credits
1.	25CS501PC	Automata Theory and Compiler Design	3	0	0	3
2.	25CS502PC	Artificial Intelligence	3	0	0	3
3.	25CS503PC	DevOps	3	0	0	3
4.		Professional Elective-I	3	0	0	3
5.		Open Elective-I	2	0	0	2
6.	25CS504PC	Compiler Design Lab	0	0	2	1
7.	25CS505PC	Artificial Intelligence with Python Lab	0	0	2	1
8.	25CS506PC	DevOps Lab	0	0	2	1
9.	25CS507PC	Field-Based Research Project	0	0	4	2
10.	25CS508SD	Skill Development Course	0	0	2	1
11.	25VA502HS	Indian Knowledge System	1	0	0	1
<b>Total Credits</b>			<b>15</b>	<b>0</b>	<b>12</b>	<b>21</b>

**III YEAR II SEMESTER**

S. No	Course Code	Course Title	L	T	P	Credits
1.	25CS601PC	Cryptography and Networks Security	3	0	0	3
2.	25CS602PC	Deep Learning	3	0	0	3
3.	25MS603HS	Business Economics and Financial Analysis	3	0	0	3
4.		Professional Elective-II	3	0	0	3
5.		Open Elective - II	2	0	0	2
6.	25CS604PC	Cryptography and Networks Security Lab	0	0	2	1
7.	25CS605PC	Deep Learning Lab	0	0	2	1
8.	25CS606PC	Advanced Data Structures using Python Lab	0	0	2	1
9.	25EN607HS	English for Employability Skills Lab	0	0	2	1
10.	25CS608SD	Skill Development Course	0	0	2	1
11.	25VA600HS / 25VA601HS	Gender Sensitization*/ Human Values and Professional Ethics*	1	0	0	0.5+0.5
<b>Total Credits</b>			<b>15</b>	<b>0</b>	<b>10</b>	<b>20</b>

**\*Note: For the courses Gender Sensitization and Human Values and Professional Ethics - one hour of instruction will be conducted on alternate weeks. For example, if a one-hour class for Gender**

Sensitization is conducted this week, then a one-hour class for Constitution of India will be conducted in the following week.

**IV YEAR I SEMESTER**

S. No.	Course Code	Course Title	L	T	P	Credits
1.	25CS701PC	Natural Language Processing	3	0	0	3
2.	25CS702PC	Cyber Security	3	0	0	3
3.	25MS703HS	Fundamentals of Management	3	0	0	3
4.		Professional Elective-III	3	0	0	3
5.		Professional Elective - IV	3	0	0	3
6.		Open Elective - III	2	0	0	2
7.	25CS704PC	Natural Language Processing Lab	0	0	2	1
8.	25CS705PC	Cyber Security Lab	0	0	2	1
9.	25CS706PC	Industry Oriented Mini Project/ Internship	0	0	4	2
<b>Total Credits</b>			<b>17</b>	<b>0</b>	<b>08</b>	<b>21</b>

**IV YEAR II SEMESTER**

S. No.	Course Code	Course Title	L	T	P	Credits
1.		Professional Elective - V	3	0	0	3
2.		Professional Elective - VI	3	0	0	3
3.	25CS801PC	Project Work	0	0	42	14
<b>Total Credits</b>			<b>6</b>	<b>0</b>	<b>42</b>	<b>20</b>

**PROFESSIONAL ELECTIVES****Professional Elective - I**

25CS511PE	Computer Graphics
25CS512PE	Introduction to Data Science
25CS513PE	Software Testing Methodologies
25CS514PE	Data Mining
25CS515PE	Web Programming
25CS516PE	Distributed Systems

**Professional Elective - II**

25CS621PE	Image Processing
25CS622PE	Blockchain Technology
25CS623PE	Software Project Management
25CS624PE	Mining Massive Datasets
25CS625PE	Full Stack Development
25CS626PE	Generative AI

**Professional Elective - III**

25CS731PE	Computer Vision
25CS732PE	Scripting Languages
25CS733PE	Vulnerability and Penetration Testing
25CS734PE	Data Stream Mining
25CS735PE	Cloud Computing
25CS736PE	Information Retrieval Systems

**Professional Elective - IV**

25CS731PE	Augmented Reality & Virtual Reality
25CS732PE	Agile Methodology
25CS733PE	Big Data Analytics
25CS734PE	Quantum Computing
25CS735PE	Robotic Process Automation
25CS736PE	Cyber Forensics

**Professional Elective - V**

25CS731PE	Social Media Mining
25CS732PE	Nature Inspired Computing
25CS733PE	Internet of Things
25CS734PE	Game Theory
25CS735PE	Mobile Application Development
25CS736PE	Human Computer Interaction

**Professional Elective - VI**

25CS731PE	High Performance Computing
25CS732PE	Edge Computing
25CS733PE	Graph Theory
25CS734PE	Adhoc and Sensor Networks
25CS735PE	Sustainable Engineering
25CS736PE	Distributed Databases

**Skill Development Courses**

SD	Node JS
SD	React JS
SD	Django
SD	Data Visualization - R Programming
SD	Python
SD	Power BI
SD	UI Design - Flutter
SD	Android Studio
SD	Prompt Engineering

**OPEN ELECTIVES****Open Elective -1:**

25CS511OE	Operating Systems
25CS512OE	Database Management Systems

**Open Elective -2:**

25CS621OE	Introduction to Computer Networks
25CS622OE	Software Engineering

**Open Elective -3:**

25CS731OE	Algorithms Design and Analysis
25CS732OE	Fundamentals of Cyber Security

**25CS105ES: PROGRAMMING FOR PROBLEM SOLVING**  
(Common to CSE, CSE(AI&ML), CSE(DS), ECE & CE)

**B.Tech. I Year I Sem.**

**L T P C**  
**3 0 0 3**

**Course Objectives:**

1. To learn the fundamentals of computers.
2. To understand the various steps in program development.
3. To learn the syntax and semantics of the C programming language.
4. To learn the usage of structured programming approaches in solving problems.
5. To develop problem-solving skills by implementing efficient algorithms and debugging techniques using C

**Course Outcomes:** Students are able to

1. Understand the basics of C language to write and analyze simple programs.
2. Apply top-down design principles using functions and pointers to develop modular and reusable C programs.
3. Utilize arrays and strings effectively in C programming for data storage.
4. Implement recursive solutions and work with user-defined data types for complex data management.
5. Understand file operations and apply fundamental searching and sorting algorithms to process data efficiently.

**UNIT - I:** Overview of C:C Language Elements, Variable Declarations and Data Types, Executable Statements, General Form of a C Program, Operators, expressions and precedence, Formatting Numbers in Program Output.

Selection Structures: Control Structures, Conditions, if Statement, if Statements with Compound Statements, Decision Steps in Algorithms.

Repetition and Loop Statements: Repetition in Programs, Counting Loops and the while Statement, Computing a Sum or Product in a Loop, for Statement, Conditional Loops, Loop Design, Nested Loops, do-while Statement.

**UNIT - II:** Top-Down Design with Functions: Building Programs from Existing Information, Library Functions, Top-Down Design and Structure Charts, Functions without Arguments, Functions with Input Arguments.

Pointers and Modular Programming: Pointers and the Indirection Operator, Functions with Output Parameters, Multiple Calls to a Function with Input/ Output Parameters, Scope of Names, Formal Output Parameters as Actual Arguments.

**UNIT - III:** Arrays: Declaring and Referencing Arrays, Array Subscripts, Using for Loops for Sequential Access, Using Array Elements as Function Arguments, Array Arguments, Searching and Sorting an Array, Parallel Arrays and Enumerated Types, Multidimensional Arrays.

Strings: String Basics, String Library Functions: Assignment and Substrings, Longer Strings: Concatenation and Whole-Line Input, String Comparison, Arrays of Pointers.

**UNIT - IV:** Recursion: The Nature of Recursion, Tracing a Recursive Function, Recursive Mathematical Functions, Recursive Functions with Array and String Parameters

Structure and Union Types: User-Defined Structure Types, Structure Type Data as Input and Output Parameters, Functions with Structured Result Values, Union Types.

**UNIT - V:** Text and Binary File Pointers: Input/ Output Files - Review and Further Study, Binary Files, Searching a Database.

Searching and Sorting: Basic searching in an array of elements (linear and binary search techniques), Basic algorithms to sort array of elements (Bubble, Insertion and Selection sort algorithms).

**TEXT BOOKS:**

1. Problem Solving and Program Design in C, Jeri R. Hanly and Elliot B. Koffman, 7th Edition, Pearson.
2. C Programming and Data Structures, B.A. Forouzan and R.F. Gilberg Cengage Learning, (3rd Edition), McGraw.

**REFERENCE BOOKS:**

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India.
2. E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill.
3. Yashavant Kanetkar, Let Us C, 18th Edition, BPB.
4. R.G. Dromey, How to solve it by Computer, Pearson (16th Impression).
5. Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
6. Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition.
7. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill.

**25CS107ES: PROGRAMMING FOR PROBLEM SOLVING LAB**  
(Common to CSE, CSE(AI&ML), CSE(DS), ECE & CE)

**B.Tech. I Year I Sem.**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>2</b>	<b>1</b>

*[Note: The programs may be executed using any available Open Source/ Freely available IDE  
Some of the Tools available are:*

*CodeLite: <https://codelite.org/>*

*Code::Blocks: <http://www.codeblocks.org/>*

*DevCpp: <http://www.bloodshed.net/devcpp.html> Eclipse:*

*<http://www.eclipse.org>*

*This list is not exhaustive and is NOT in any order of preference]*

**Course Objectives:** The students will learn the following:

1. To familiarize students with the use of an IDE to create, compile, run, and debug C programs.
2. To provide an understanding of the fundamental steps involved in program development.
3. To introduce the basic concepts of C such as operators, control structures, and functions for problem-solving.
4. To develop knowledge of modular programming, arrays, and dynamic memory allocation for writing efficient programs.
5. To expose students to file handling concepts including creating, reading, and writing text and binary files.

**Course Outcomes:** Students are able to:

1. Demonstrate understanding and application of fundamental C programming constructs.
2. Develop, trace, and analyze C programs using control structures.
3. Decompose complex problems into modular solutions using functions.
4. Implement string manipulation and file handling techniques.
5. Apply and evaluate searching and sorting algorithms.

**PRACTICE SESSIONS:**

**Simple numeric problems:**

- a) Write a program for finding the max and min from the three numbers.
- b) Write the program for the simple, compound interest.
- c) Write a program that prints a multiplication table for a given number and the number of rows in the table. For example, for a number 5 and rows = 3, the output should be:
  - 5 x 1 = 5
  - 5 x 2 = 10
  - 5 x 3 = 15
- d) Write a program that shows the binary equivalent of a given positive number between 0 to 255.

**Expression Evaluation:**

- a) Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +, -, \*, /, % and use Switch Statement).
- b) Write a program that finds if a given number is a prime number.
- c) Write a C program to find the sum of individual digits of a positive integer and test given number is palindrome.
- d) A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.

**Arrays, Pointers and Functions:**

- a) Write a C program to find the minimum, maximum and average in an array of integers.
- b) Write a C program that uses functions to perform the following:
  - I. Addition of Two Matrices
  - II. Multiplication of Two Matrices
- c) Write a program for reading elements using a pointer into an array and display the values using the array.
- d) Write a program for display values reverse order from an array using a pointer.

**Files:**

- a) Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents.
- b) Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file).

**Strings:**

- a) Write a C program that uses functions to perform the following operations:
  - I. To insert a sub-string into a given main string from a given position.
  - II. To delete n Characters from a given position in a given string
- b) Write a C program to determine if the given string is a palindrome or not (Spelled same in both directions with or without a meaning like madam, civic, noon, abcba, etc.)
- c) Write a C program that displays the position of a character ch in the string S or - 1 if S doesn't contain ch.
- d) Write a C program to count the lines, words and characters in a given text.

**Sorting and Searching:**

- a) Write a C program that uses non-recursive function to search for a Key value in a given list of integers using linear search method.
- b) Write a C program that uses non-recursive function to search for a Key value in a given sorted list of integers using binary search method.
- c) Write a C program that implements the Bubble sort method to sort a given list of integers in ascending order.
- d) Write a C program that sorts the given array of integers using selection sort in descending order
- e) Write a C program that sorts the given array of integers using insertion sort in ascending order
- f) Write a C program that sorts a given array of names.

**TEXT BOOKS:**

1. Program Design in C, Jeri R. Hanly and Elliot B.Koffman, Problem solving and 7th Edition, Pearson.
2. C Programming and Data Structures, B.A. Forouzan and R.F. Gilberg , (3rd Edition), Cengage Learning

**REFERENCE BOOKS:**

1. The C Programming Language, Brian W. Kernighan and Dennis M. Ritchie, 2<sup>nd</sup> Edition, Prentice Hall of India,
2. Computer fundamentals, E. Balagurusamy, and C, 2nd Edition, McGraw-Hill
3. Let Us C, Yashavant Kanetkar, 18th Edition, BPB
4. How to solve it by Computer, R.G. Dromey, (16th Impression), Pearson
5. Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
6. C: The Complete Reference, Herbert Schildt, 4th Edition Mc Graw Hill.
7. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill

**25CS205ES: DATA STRUCTURES**  
(Common to CSE, CSE(AI&ML), CSE(DS), & ECE)

**B.Tech. I Year II Sem.**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Prerequisites:** A course on “Programming for Problem Solving

**Course Objectives**

1. To introduce the fundamental concepts, terminology, and classifications of data structures.
2. To provide knowledge of linear data structures such as lists, stacks, and queues along with their applications.
3. To impart understanding of non-linear data structures such as trees, search trees, heaps, tries, and graphs.
4. To familiarize students with various algorithms for sorting, searching, and pattern matching.
5. To develop an understanding of hashing techniques, collision resolution strategies, and file organization methods for efficient data management.

**Course Outcomes:** Students are able to

1. Understand and implement basic linear data structures like linked lists, stacks, and queues.
2. Construct and manipulate various trees, including binary search and balanced trees, applying traversal and balancing techniques.
3. Analyze and implement multi-way search trees and heaps, applying efficient search algorithms.
4. Represent graphs, perform graph traversals, and apply advanced sorting algorithms.
5. Implement hashing techniques, handle collisions, and understand file organization and indexing methods.

**UNIT – I**

Introduction to Data Structures: Basic Terminology, Classification of Data Structures, Operation on Data Structures, abstract data types, selecting a Data Structure, Linear list – Introduction, singly linked list, Circular Linked Lists, Doubly Linked List, Stacks- Operations, Stack algorithm, Stack ADT, Stack applications, Queues- operations, Queue Algorithm, Queue ADT, Queue Applications.

**UNIT - II**

Trees: Introduction, Types of Trees, creating a Binary Tree from a General Tree, traversing a Binary Tree, Binary Search Trees (BST), BST Operations- Searching, Insertion and Deletion, BST ADT, BST Applications, Threaded Binary Trees, AVL Trees, Red -Black Trees, Splay Trees

**UNIT – III**

Pattern Matching and Tries: Pattern matching algorithms-Brute force, the Boyer-Moore algorithm, the Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, Suffix tries.

**UNIT - IV**

Graphs: Introduction, Directed Graphs, Bi-connected Components, Representation of Graphs, Graph Traversal Algorithms, Graph ADT, Applications of Graphs  
Sorting: Radix Sort, Heap sort, Shell Sort, Tree Sort,

**UNIT – V**

Hashing and Collision: Introduction, Hash Tables, Hash Functions, Different Hash Functions: Division Method, Multiplication Method, Mid-square Method, Folding Method; collisions: Collision Resolution by Open Addressing, Collision Resolution by Chaining  
Files and their Organization: Introduction, Data hierarchy, File Attributes, Text and Binary Files, Basic File Operations, File Organization, Indexing

**TEXTBOOKS:**

1. Data Structures: A Pseudocode Approach with C, R. F. Gilberg and B.A.Forouzan, 2nd Edition, Cengage Learning
2. Data Structure using C- Reema Thareja, 3<sup>rd</sup> Edition, Oxford University Press.

**REFERENCE:**

1. Data Structures using C - A. S.Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson Education.

**25CS207ES: DATA STRUCTURES LAB**  
(Common to CSE, CSE(AI&ML), CSE(DS), & ECE)

**B.Tech. I Year II Sem.**

**L T P C**  
**0 0 2 1**

**Prerequisites:** 1. A Course on “Programming for problem solving”.

**Course Objectives:**

1. To enable students to implement programs in C for solving problems using fundamental programming constructs.
2. To provide practical exposure to the implementation of linear data structures such as arrays, stacks, and queues.
3. To develop skills in applying searching and sorting algorithms to process and organize data efficiently.
4. To impart hands-on experience with non-linear data structures such as linked lists, trees, and graphs.
5. To train students in analyzing the performance of different algorithms and selecting appropriate data structures for problem-solving.

**Course Outcomes:** Students are be able to

1. Implement and manipulate various types of linked lists with fundamental operations.
2. Design and develop stack and queue data structures using arrays and abstract data types.
3. Apply and implement different sorting algorithms to organize data in ascending order.
4. Construct and traverse complex tree data structures including Binary Search Trees, B- Trees, B+ Trees, AVL Trees, and Red-Black Trees.
5. Implement graph traversal algorithms and various hashing functions for efficient data searching and storage.

**List of Experiments**

1. Write a program that uses functions to perform the following operations on singly linked list.:  
i) Creation      ii) Insertion      iii) Deletion      iv) Traversal
2. Write a program that uses functions to perform the following operations on doubly linked list.:  
i) Creation      ii) Insertion      iii) Deletion      iv) Traversal
3. Write a program that uses functions to perform the following operations on circular linked list.:  
i) Creation      ii) Insertion      iii) Deletion      iv) Traversal
4. Write a program that implement stack (its operations) using  
i) Arrays      ii) ADT
5. Write a program that implement Queue (its operations) using  
i) Arrays      ii) ADT
6. Write a program that implements the following sorting methods to sort a given list of integers in ascending order  
i) Radix Sort, ii) Heap sort, iii) Shell Sort, iv) Tree Sort
7. Write a program to implement the tree traversal methods (Recursive and Non-Recursive).
8. Write a program to implement  
i) Binary Search tree      ii) AVL trees      iii) Red - Black trees
9. Write a program to implement the graph traversal methods.
10. Write a program to implement the following Hash Functions: i) Division Method, ii) Multiplication Method, iii) Mid-square Method, iv) Folding Method

**TEXT BOOKS:**

1. Fundamentals of Data Structures in C, E. Horowitz, S. Sahni and Susan Anderson Freed, 2<sup>nd</sup> Edition, Universities Press.
2. Data Structures using C A. S. Tanenbaum, Y. Langsam, and M. J. Augenstein, - 2<sup>nd</sup> Edition, PHI/Pearson Education.

**REFERENCE BOOK:**

1. Data Structures: A Pseudocode Approach with C, R. F. Gilberg and B. A. Forouzan, 2<sup>nd</sup> Edition, Cengage Learning.

**25CS108ES / 25CS208ES: IT WORKSHOP**  
**(Common to CSE, CSE(AI&ML), & CSE(DS))**

**B.Tech. I Year II Sem.**

**L T P C**  
**0 0 2 1**

**Course Objectives:** The IT Workshop for engineers is a training lab course spread over 60 hours. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel, PowerPoint and Publisher.

**Course Outcomes:** Students are able to

1. Perform Hardware troubleshooting
2. Understand Hardware components and inter dependencies
3. Safeguard computer systems from viruses/worms
4. Document/ Presentation preparation
5. Perform calculations using spreadsheets

**PC Hardware**

**Task 1:** Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.

**Task 2:** Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva. Also students need to go through the video which shows the process of assembling a PC. A video would be given as part of the course content.

**Task 3:** Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.

**Task 4:** Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both Windows and Linux. Lab instructors should verify the installation and follow it up with a Viva

**Internet & World Wide Web**

**Task1: Orientation & Connectivity Boot Camp:** Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should demonstrate, to the instructor, how to access the websites and email. If there is no internet connectivity preparations need to be made by the instructors to simulate the WWW on the LAN.

**Task 2: Web Browsers, Surfing the Web:** Students customize their web browsers with the LAN proxy settings, bookmarks, search toolbars and pop up blockers. Also, plug-ins like Macromedia Flash and JRE for applets should be configured.

**Task 3: Search Engines & Netiquette:** Students should know what search engines are and how to use the search engines. A few topics would be given to the students for which they need to search on Google. This should be demonstrated to the instructors by the student.

**Task 4: Cyber Hygiene:** Students would be exposed to the various threats on the internet and would be asked to configure their computer to be safe on the internet. They need to customize their browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

**LaTeX and WORD**

**Task 1 – Word Orientation:** The mentor needs to give an overview of LaTeX and Microsoft (MS) office or equivalent (FOSS) tool word: Importance of LaTeX and MS office or equivalent (FOSS) tool Word as

word Processors, Details of the four tasks and features that would be covered in each, Using LaTeX and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word.

**Task 2: Using LaTeX and Word** to create a project certificate. Features to be covered:- Formatting Fonts in word, Drop Cap in word, Applying Text effects, Using Character Spacing, Borders and Colors, Inserting Header and Footer, Using Date and Time option in both LaTeX and Word.

**Task 3: Creating project abstract** Features to be covered:-Formatting Styles, Inserting table, Bullets and Numbering, Changing Text Direction, Cell alignment, Footnote, Hyperlink, Symbols, Spell Check, Track Changes.

**Task 4: Creating a Newsletter:** Features to be covered:- Table of Content, Newspaper columns, Images from files and clipart, Drawing toolbar and Word Art, Formatting Images, Textboxes, Paragraphs and Mail Merge in word.

### **Excel**

**Excel Orientation:** The mentor needs to tell the importance of MS office or equivalent (FOSS) tool Excel as a Spreadsheet tool, give the details of the four tasks and features that would be covered in each. Using Excel – Accessing, overview of toolbars, saving excel files, Using help and resources.

**Task 1: Creating a Scheduler** - Features to be covered: Gridlines, Format Cells, Summation, auto fill, Formatting Text

**Task 2: Calculating GPA** - Features to be covered:- Cell Referencing, Formulae in excel - average, std. deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function, LOOKUP/VLOOKUP

**Task 3:** Split cells, freeze panes, group and outline, Sorting, Boolean and logical operators, Conditional formatting

### **PowerPoint**

**Task 1:** Students will be working on basic power point utilities and tools which help them create basic PowerPoint presentations. PPT Orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in PowerPoint.

**Task 2:** Interactive presentations - Hyperlinks, Inserting -Images, Clip Art, Audio, Video, Objects, Tables and Charts.

**Task 3:** Master Layouts (slide, template, and notes), Types of views (basic, presentation, slide slotter, notes etc), and Inserting - Background, textures, Design Templates, Hidden slides.

### **REFERENCE BOOKS:**

1. Comdex Information Technology course toolkit Vikas Gupta, *WILEY Dreamtech*
2. The Complete Computer upgrade and repair book, Cheryl A Schmidt, 3rd edition *WILEY Dreamtech*
3. Introduction to Information Technology, ITL Education Solutions limited, 7<sup>th</sup> Edition, *Pearson Education*.
4. PC Hardware - A Handbook - 1<sup>st</sup> Edition KateJ. Chase *PHI (Microsoft)*
5. LaTeX Companion - Leslie Lamport, 2<sup>nd</sup> Edition, *PHI/Pearson*.
6. IT Essentials PC Hardware and Software Companion Guide, David Anfinson and Ken Quamme Third Edition by . - *CISCO Press, Pearson Education*.
7. IT Essentials PC Hardware and Software Labs and Study Guide, by Patrick Regan, Third Edition - *CISCO Press, Pearson Education*.

**25CS210ES: PYTHON PROGRAMMING LAB**  
(Common to CSE, CSE(AI&ML), CSE(DS), & CE)

**B.Tech. I Year II Sem.**

**L T P C**  
**0 0 2 1**

**Course Objectives:**

- To install and run the Python interpreter
- To learn control structures.
- To Understand Lists, Dictionaries in python
- To Handle Strings and Files in Python

**Course Outcomes:** Students are able to

1. Apply fundamental Python programming concepts to solve basic problems.
2. Implement and manipulate Python data structures including list, tuples, dictionaries, strings, and matrices for data processing.
3. Demonstrate file handling operations, module creation, and exception handling for building robust Python applications.
4. Utilize Object-Oriented Programming principles in Python to design and implement custom classes and graphical representations using canvas.
5. Integrate external Python libraries like NumPy, SciePy, and Tkinter for numerical computing, data visualization, and GUI development.

Note: The lab experiments will be like the following experiment examples.

**List of Experiments:**

1.
  - I. Use a web browser to go to the Python website <http://python.org>. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.
  - II. Start the Python interpreter and type `help()` to start the online help utility.
2. Start a Python interpreter and use it as a Calculator.
3. Write a program to calculate compound interest when principal, rate and number of periods are given.
4. Read the name, address, email and phone number of a person through the keyboard and print the details.
5. Print the below triangle using for loop. 5
 

```

4 4
3 3 3
2 2 2 2
1 1 1 1 1
```
6. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character(use 'if-else-if' ladder)
7. Python program to print all prime numbers in a given interval (use break)
8. Write a program to convert a list and tuple into arrays.
9. Write a program to find common values between two arrays.
10. Write a function called `palindrome` that takes a string argument and returns True if it is a palindrome and False otherwise. Remember that you can use the built-in function `len` to check the length of a string.
11. Write a function called `is_sorted` that takes a list as a parameter and returns True if the list is sorted in ascending order and False otherwise.
12. Write a function called `has_duplicates` that takes a list and returns True if there is any element that appears more than once. It should not modify the original list.
13. Write a function called `remove_duplicates` that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order.
14. The wordlist I provided, `words.txt`, doesn't contain single letter words. So you might want to add "l", "a", and the empty string.

15. Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.
16. Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'
17. Remove the given word in all the places in a string?
18. Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?
19. Writes a recursive function that generates all binary strings of n-bit length
20. Write a python program that defines a matrix and prints
21. Write a python program to perform multiplication of two square matrices
22. How do you make a module? Give an example of construction of a module using different geometrical shapes and operations on them as its functions.
23. Use the structure of exception handling all general-purpose exceptions.
24. Write a function called draw\_rectangle that takes a Canvas and a Rectangle as arguments and draws a representation of the Rectangle on the Canvas.
25. Add an attribute named color to your Rectangle objects and modify draw\_rectangle so that it uses the color attribute as the fill color.
26. Write a function called draw\_point that takes a Canvas and a Point as arguments and draws a representation of the Point on the Canvas.
27. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called draw\_circle that draws circles on the canvas.
28. Write a python code to read a phone number and email-id from the user and validate it for correctness.
29. Write a Python code to merge two given file contents into a third file.
30. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.
31. Write a Python code to Read text from a text file, find the word with most number of occurrences
32. Write a function that reads a file *file1* and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters.
33. Import numpy, Plotpy and Scipy and explore their functionalities.
34. Install NumPy package with pip and explore it.
35. Write a program to implement Digital Logic Gates - AND, OR, NOT, EX-OR
36. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.

**TEXT BOOKS:**

1. Supercharged Python: Take your code to the next level, Overland, 2019.
2. Learning Python, Mark Lutz, 5<sup>th</sup> Edition, O'reilly.

**REFERENCE BOOKS:**

1. Python Programming: A Modern Approach, Vamsi Kurama, Pearson, 2018.
2. Python Programming A Modular Approach with Graphics, Database, Mobile, and Web Applications, Sheetal Taneja, Naveen Kumar, 1<sup>st</sup> Edition, Pearson.
3. Introduction to Python Programming, Gowrishakar S, Veena A, CRC Press, 2018.
4. Programming with Python, A User's Book, Michael Dawson, Cengage Learning, 3<sup>rd</sup> Edition, India.
5. Python for Data Science, Dr. Mohd Abdul Hameed, Wiley publications, 2021.
6. Core Python Programming, Dr. R. Nageswara Rao, 3<sup>rd</sup> Edition, Dreamtech press
7. Introduction to Python, Gowrishankar S, Veena A., CRC Press 2018.

**25CS203ES: PYTHON PROGRAMMING**  
(Common to ECE, & CE)

**B.Tech. I Year II Sem.**

**L T P C**  
**3 0 0 3**

**Prerequisites:** Basic knowledge of computer fundamentals, C programming.

**Course Objectives:**

**Introduce the fundamentals of Python programming for problem-solving.**

1. Develop skills to write structured, modular, and efficient Python code.
2. Enable students to use Python's built-in data structures and libraries effectively.
3. Provide knowledge on file handling, exception handling, and object-oriented programming in Python.
4. Equip students with the ability to apply Python for real-world applications, including data processing and automation.
5. To introduce the principles of object-oriented programming in Python and demonstrate their use in developing real-world applications.

**Course Outcomes:** Students are able to

1. Write Python programs using variables, operators, expressions, and control structures.
2. Implement Python programs using built-in data structures like lists, tuples, sets, and dictionaries.
3. Apply modular and object-oriented programming principles in Python.
4. Handle files, exceptions, and apply Python libraries for problem-solving.
5. Develop small-scale applications in Python for automation and data manipulation.

**UNIT-1 – Introduction to Python and Basics of Programming**

Introduction to Python: Features, Applications, Installation, IDEs, Python Syntax, Indentation, Comments, Variables, Data Types, Type Casting, Operators: Arithmetic, Relational, Logical, Assignment, Membership, Identity, Bitwise, Input/Output functions (input(), print()), Control Structures: if, if-else, if-elif-else, Nested Conditions, Looping: for, while, Nested Loops, break, continue, pass.

**UNIT-2 – Data Structures in Python**

Strings: Creation, Indexing, Slicing, Methods, String Formatting, Lists: Creation, Indexing, Slicing, List Comprehension, Methods, Tuples: Properties, Indexing, Methods, Sets: Creation, Operations, Methods, Dictionaries: Creation, Access, Methods, Dictionary Comprehension, Iterating over data structures.

**UNIT-3 – Functions and Modules**

Functions: Defining, Calling, Parameters, Return Values, Types of Arguments: Positional, Keyword, Default, Variable Length, Scope of Variables: Local and Global, Lambda Functions, Map, Filter, Reduce, Recursion in Python, Modules: Importing, Creating User-defined Modules, Standard Modules (math, random, datetime), Packages in Python.

**UNIT-4 – File Handling and Exception Handling**

File Handling: Opening, Reading, Writing, Appending, File Modes, File Methods, Working with CSV and

JSON Files, Exception Handling: try, except, else, finally, Built-in Exceptions, Raising Exceptions, Introduction to Regular Expressions (re module).

### **UNIT-5 – Object-Oriented Programming and Applications**

OOP Basics: Classes, Objects, Attributes, Methods, Constructor (`__init__`), self keyword, Inheritance: Single, Multiple, Multilevel, Hierarchical, Method Overriding, Method Overloading (conceptual), Encapsulation and Polymorphism, Application Development: Data Processing Script, Basic Calculator, File Organizer, Simple Data Analysis with pandas.

#### **TEXT BOOKS:**

1. Python Programming: Using Problem Solving Approach by Reema Thareja.
2. Python Crash Course by Eric Matthes, Learning Python by Mark Lutz.

#### **REFERENCE BOOKS:**

1. Introduction to Python Programming by Gowrishankar S., Veena A.
2. Python Cookbook by David Beazley and Brian K. Jones.
3. Fluent Python by Luciano Ramalho, Automate the Boring Stuff with Python by Al Sweigart.