Scheme & Syllabi

of

B. Tech. (Computer Science and Engineering)

July 2026

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

Department of Computer Science & Engineering
Updated UG (CSE) Scheme

First Semester					
SN	**Course Code**	**Subjects**	**L-T-P**	**Credit**	**Type**
		Mathematics I	**3-0-0**	**3**	**IC**
		Physics	**3-0-0**	**3**	**IC**
		Python Programming	**2-0-0**	**2**	**IC**
		Programming with Python Lab	**0-0-2**	**1**	**IC**
		Environmental Science	1-0-0	1	IC
		Ethics and Human Values	**1-0-0**	**1**	**IC**
		Soft Skills Foundational Lab	**0-0-2**	**1**	**IC**
		Physics Lab	0-0-2	1	IC
		Engineering Drawing & Sketching Lab	0-0-2	1	IC
				14	
	22CST103	Problem Solving using C	2-0-0	2	PC
	22CST104	Discrete Mathematic	3-0-0	3	PC
	22CSP105	Problem Solving using C Lab	0-0-2	1	PC
				6	
				20	

Second Semester					
SN	**Course Code**	**Subjects**	**L-T-P**	**Credit**	**Type**
		Mathematics II	**3-0-0**	**3**	**IC**
		Data Science /Analytics /Visualization	**2-0-0**	**2**	**IC**
		Makers Lab	**0-0-12**	**6**	**IC**
				11	
	22CST151	Data Structures	3-0-0	3	PC
	22CST152	Digital Logic and Circuits Design	3-0-0	3	PC
	22CSP153	Data Structures Lab	0-1-2	2	PC
	22CST154	Digital Logic and Microprocessor Lab	0-0-2	1	PC
				9	
				20	

Programming with Python					
Prerequisite: :NiL		L	T	P	C
Total hours: 28		2	0	0	2
Course Content					Hrs
Unit 1	Introduction to **computer system and binary number systems**; Representation of Negative Numbers, signed and unsigned numbers, Binary arithmetic operation: addition, subtraction (2's complement), multiplication, left shifting and right shifting, Number Systems and conversions				3
Unit 2	**Introduction to Python:** Python's basic Data Types, Python Variables, Python basic operators, operands and expressions, understanding Python blocks. Python data types. Declaring and using numeric data types: int, float etc. ; Taking user Input and Printing output (input() & print ()), formatted output - f-strings, comments in Python **Python program flow control**: Branching and Conditional Constructs: if, else and elif. Loop Control structure: loops in Python. For loops using ranges, strings, lists and dictionaries. Use of while loops in Python. Loop manipulation using pass, continue, break and else in looping. Programming using Python conditional and loop blocks.				8
Unit 3	**Complex Data Types in Python**: defining Lists, Tuples, Strings, their manipulation operations, indexing & slicing, List comprehension, **Dictionary, Set**: various operations and related in-built-method: items(), map(), split() ; String manipulation methods, list manipulation, dictionary manipulation, and programming using string, list and dictionary in-built functions.				7
Unit 4	**Python Functions:** Definition and use , Organizing Python code using functions, function Types, Built-in & User defined function, nested function, Parameter passing, block structure, Different Types of Arguments, scope, Lambda functions, Recursive function				4
Unit 5	**Object Oriented Programming (OOP) concepts**: Introduction to Classes and objects: creation and usage, Instance and Class Variables, attributes, methods, Constructors				2
Unit 6	**Python File Operations:** Reading and writing files, File Handling and processing, file related operations, working with .csv and json data files (reading, wrting) **Modules, Package, Library**: introduction & creation of Modules, Packages & Library, introduction to basic libraries and packages. Case study: Development of mini projects using various python libraries				4
References					
1.	Wesley J. Chun, "Core Python Applications Programming", 3rd Edition, Pearson				

	Education, 2016.
2.	Charles Dierbach, “Introduction to Computer Science using Python”, Wiley, 2015.
3	Jeeva Jose & P. Sojan Lal, “Introduction to Computing and Problem Solving with Python”, Khanna Publishers, New Delhi, 2016.
4	Downey, A. et al., “How to Think Like a Computer Scientist: Learning with Python”, John Wiley, 2015.
5	Mark Lutz, “Learning Python”, 5th Edition, O’Reilly Publication, 2013, ISBN 978-1449355739.
6	John Zelle, “Python Programming: An Introduction to Computer Science”, 2nd Edition, Course Technology Cengage Learning Publications, 2013, ISBN 978-1590282410.
7	Michael Dawson, “Python Programming for Absolute Beginners”, 3rd Edition, Course Technology Cengage Learning Publications, 2013, ISBN 978-1435455009.
8	David Beazley, Brian Jones, “Python Cookbook”, 3rd Edition, O’Reilly Publication, 2013, ISBN 978-1449340377.

Problem solving using C					
Prerequisite: :NiL		L	T	P	C
Total hours: 28		2	0	0	2
Course Content					Hrs
Unit 1	Introduction to Computers, Computational Thinking and Problem Solving, Algorithm Design Basics of Compiler, Interpreter and Compilation Process. Introduction to C Programming, Structure of a C Program, Data Types and Storage Classes (auto, static, extern, register), Variables, Constants, Identifiers, Keywords, Operators and Expressions, Formatted Input/Output using printf() and scanf(), Built-in Data Types, Integer Arithmetic, Type Conversion and Type Casting.				5
Unit 2	**Decision-making statements** (if, if-else, nested if, switch, conditional/ternary operator) with programming example. **Looping constructs** (for, while, do-while), significance of initialization, termination condition and increment/decrement, pre/post increment operators. **Arrays:** One-dimensional and two- dimensional arrays, character arrays and strings, standard string library functions, string manipulation, palindrome and related problems.				6
Unit 3	**Functions:** declaration, definition, function prototypes, parameter passing, call by value, call by reference, recursive functions, command-line arguments, variable number of arguments. Searching & Sorting algorithms: Linear Search and Binary Search. Bubble Sort and Selection Sort. Recursion, limitations of recursion.				6
Unit 4	**Pointers:** declaration and initialization, pointer arithmetic, void pointers, pointers versus arrays, pointer to array versus array of pointers, Function pointer, Dangling pointers , wild pointers, const pointers, Multidimensional arrays and matrix operations (addition, multiplication and transpose). **Structures and Unions**, typedef, Nested Structures, Arrays of Structures, Pointers to Structures, Pointer to Functions, Enumeration (enum)				6
Unit 5	**Dynamic Memory Allocation**: using malloc(), calloc(), realloc() and free(), linked list as a case study. **File Handling in C:** text and binary files, file opening and closing, reading, writing, Header Files, Concept of multi-file programming and makefile utilities, Testing and Debugging Techniques, Applications of C Programming.				5
References					
1.	**Brian W. Kernighan and Dennis M. Ritchie**, *The C Programming Language*, 2nd Edition, Pearson Education, 1988.				
2.	**Paul Deitel and Harvey Deitel**, *C: How to Program*, 9th Edition, Pearson, 2021. *(Comprehensive textbook with numerous programming examples and exercises.)*				
3.	**Stephen G. Kochan**, *Programming in C*,4th Edition, Addison-Wesley Professional, 2014.				
4.	**K. N. King**, *C Programming: A Modern Approach*, 2nd Edition, W. W. Norton & Company, 2008.				

Discrete Mathematics					
Prerequisite: : None		L	T	P	C
Total hours: 43		3	0	0	3
Course Content					Hrs
Unit 1	Logic: Truth Tables, Conditionals (P ⇒ Q), and Bi-conditionals (P ⇔ Q), Negation, Converse, and Contrapositive, Existential and Universal Quantifiers (∀,∃, ∃!), Proof Techniques (Contrapositive, Contradiction, Induction), Counterexamples, and Proving Statements with Quantifiers, Predicate logic, first order logic, Logical Inferences, SAT solvers and how to use these for formal logic verification.				10
Unit 2	Set Theory and Relations: Foundations of Sets, Set Operations, Subsets, and Set Equality, Power Sets, Cardinality, and Infinite Sets, Cartesian Products and Relations, Equivalence Relations and Partitions, Partial Orderings, Lattices.				5
Unit 3	Counting and Combinatorics: Basic counting principles, permutations and combinations, binomial coefficients and identities, pigeonhole principle, inclusion and exclusion principle.				6
Unit 4	Functions: Functions and their types, composition, inverse and fundamental properties, Recurrence Relations, Recurrence Relations for Algorithm Complexity Analysis (Master Theorem, Divide-and-Conquer), Tail Recursion, Generating functions.				6
Unit 5	Abstract Algebra: Binary operations and their properties, definition of a group, groups as symmetries, cyclic groups, subgroups, cosets, Lagrange's theorem, monoids, and application of cyclic groups in cryptographic key exchange.				6
Unit 6	Number Theory: Prime Numbers, Euclid's Algorithm for GCD, The GCD-LCM product theorem, Extended Euclid's Algorithm, Linear Diophantine Equations, Modular Arithmetic, Chinese Reminder Theorem, Fast Modular Exponentiation, Fermat's little theorem, Euler's totient theorem, Euler's theorem. Applications in Cryptography: Mathematical foundations of RSA cryptography and efficient RSA computation using the Chinese Remainder Theorem.				10

References

1.	Graham, Ronald L., Donald E. Knuth, and Oren Patashnik. Concrete Mathematics: A Foundation for Computer Science. 2nd ed. Addison-Wesley, 1994.
2.	Rosen, Kenneth H. Discrete Mathematics and Its Applications. 2025 Release. McGraw Hill, 2025.
3.	Lipschutz, Seymour, and Marc Lipson. Schaum's Outline of Discrete Mathematics. 4th ed. McGraw Hill, 2022.
4.	Velleman, Daniel J. How to Prove It: A Structured Approach. 3rd ed. Cambridge University Press, 2019.

Data Structures					
Prerequisite: :NiL		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs
Unit 1	Introduction to Data Structures, Classification of Data Structures, Memory Allocation (Static and Dynamic), Abstract Data Types (ADT), Arrays and their Representation, Operations on Arrays, Sparse Matrices, Algorithm Analysis, Time and Space Complexity, Introduction to Asymptotic Notations				6
Unit 2	Introduction to Linked Lists, Singly, Doubly and Circular Linked Lists, Operations on Linked Lists, Dynamic Memory Allocation using Linked Lists, Stack Implementation using Arrays and Linked Lists, Stack Applications, Queue Implementation, Sequential Queue, Circular Queue, Double-Ended Queue (Deque), Dynamic Queue Implementation using Linked Lists, Queue Applications.				7
Unit 3	Searching Techniques: Linear Search and Binary Search. Sorting Techniques: Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, Counting Sort, Radix Sort, Heap Sort. Introduction to Hashing, Hash Functions, Hash Tables, Open and Closed Hashing, Hash Collisions, Collision Resolution Techniques, Separate Chaining, Linear Probing, Quadratic Probing, Double Hashing.				10
Unit 4	Introduction to Trees, Binary Trees and their Representations, Tree Traversals (Preorder, Inorder, Postorder and Level Order), Binary Search Trees (BST), Threaded Binary Trees, AVL Trees and Rotations, Trie, Huffman Encoding Tree, Tree Applications, M-Way Search Trees, B-Tree and its Variants, B+ Tree, Heaps (Min Heap and Max Heap) and their Applications.				11
Unit 5	Fundamentals of Graphs, Graph Terminology, Graph Representations (Adjacency Matrix and Adjacency List), Graph Traversals using Depth-First Search (DFS) and Breadth-First Search (BFS), Minimum Spanning Trees using Prim's Algorithm, Shortest Path using Dijkstra's Algorithm, Applications of Graphs.				8
References					
1.	Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). MIT Press.				
2.	Lipschutz, S. (2017). Schaum's Outline of Data Structures with C. McGraw-Hill Education.				
3.	Horowitz, E., Sahni, S., & Anderson-Freed, S. (2008). Fundamentals of Data Structures in C (2nd ed.). Universities Press.				
4.	Weiss, M. A. (2014). Data Structures and Algorithm Analysis in C (2nd ed.). Pearson.				

Digital Logic and Circuit Design					
Prerequisite: :NiL		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs
Unit 1	Fundamentals of Number System and Codes :Representation of Negative Numbers, Digital Codes; Binary Coded Decimal (BCD) Arithmetic operation, Different binary codes and conversions, Gray code, Error Detection Codes Floating point representations: Fixed-point representation, Floating-point representation, IEEE 754 (introductory level)				6
Unit 2	Introduction to Logic Gates, Universal Gates and their characteristic: K-Map, SOP, POS. Combinational circuits: Adders, Subtractors, Binary Parallel Adder – Carry look ahead Adder, BCD Adder, Multiplexer, Demultiplexer, Comparator, Decoder and Encoder				8
Unit 3	Sequential Circuits: Latches, Flip-Flops: RS, D Type, JK, and T Type and their conversion, Master-Salve Flip and Race Conditions. Registers: Design of shift registers and their operations. Counters: Asynchronous and Synchronous counters, Applications of counters. Setup time and Hold time violations, metastability in clock domains, Fundamental Mode Circuits Analysis, Synthesis of Flow Tables, Minimization, Transition Tables, Excitation maps, Output Maps				16
Unit 4	Introduction to Instruction Set Architecture: Addressing Modes, Instruction Set, Code Conversation, Directives, Control Operations, String Manipulation Operations. Interfacing Devices. I/O Controller. Timing and delay, Stack and Subroutine, interrupts, Introduction to x86 , Control unit (microinstructions) and stack-based vs. register-based machines, microprocessor and Assembly Language programming				12
References					
1.	Herbert Taub, Donald L. Schilling, Digital Integrated Electronics, McGraw-Hill.				
2.	Fredrick J. Hill, Gerald R. Peterson, Computer aided logical design with emphasis on VLSI, Wiley				
3.	M. Morris Mano, Digital Logic and Computer Design, Person Education.				
4.	Lance A. Leventhal, S. Cordes, Assembly language subroutines for the 8086, McGraw-Hill Book Co.				
5	Randall Hyde, The Art of Assembly Language, 2nd Edition				

Programming with Python Lab					
Prerequisite: :NiL		L	T	P	C
Total hours: 28		0	0	2	1
Course Content					Hrs
1	**The following proposed coverage is broad guiding areas lab. The programs mentioned here just sample programs and they are just for reference purpose. The instructor offering the course in consultation with the theory offered can adopt further variations in tune with concerned theory course.** --- **Expressions and Operators** [2 Lab] Installation of Python tools, introduction to Python programming, and Data types, Input/Output using input() and print() in a program, Formatted output, library imports				4
2	**Python List and Strings operation, Doc strings** [2 Labs] Writing program using List and Strings , various String and List manipulation methods and operations				4
3	**Complex Python Data Types** [2 Labs] Writing Programs and solving problems using Tuples, Dictionaries and sets , it's various manipulation operations				4
4	**Branching and Conditional Constructs** [2 Labs] Writing Programs using decision making statements- if, else, elif, nested elif; Python Looping using for, while and nested loops Problem solving Programming using Python conditional and loop blocks				6
5	**Functions and Recursion and built-in advanced functions** [2 Labs] Programs demonstrating user-defined functions, recursive functions Problem solving and implementations using function concepts				4
6	**Object Oriented Programming concepts**[1 Labs] Implementation of programs using Classes and objects; creation of Instances, Class Variables, attributes, methods and Constructors				3
7	**Python File Handling Operations** [1 Labs] Writing Programs involving creation, reading, writing, appending, copying, and merging text files etc.				2
8	**Modules, Package, Library**[1 Labs] Demonstration of usage of libraries and packages such as NumPy , Matplotlib and other utility functions etc.				1

References

1.	Wesley J. Chun, "Core Python Applications Programming", 3rd Edition, Pearson Education, 2016.

2.	Charles Dierbach, “Introduction to Computer Science using Python”, Wiley, 2015.
3	Jeeva Jose & P. Sojan Lal, “Introduction to Computing and Problem Solving with Python”, Khanna Publishers, New Delhi, 2016.
4	Downey, A. et al., “How to Think Like a Computer Scientist: Learning with Python”, John Wiley, 2015.
5	Mark Lutz, “Learning Python”, 5th Edition, O’Reilly Publication, 2013, ISBN 978-1449355739.
6	John Zelle, “Python Programming: An Introduction to Computer Science”, 2nd Edition, Course Technology Cengage Learning Publications, 2013, ISBN 978-1590282410.
7	Michael Dawson, “Python Programming for Absolute Beginners”, 3rd Edition, Course Technology Cengage Learning Publications, 2013, ISBN 978-1435455009.
8	David Beazley, Brian Jones, “Python Cookbook”, 3rd Edition, O’Reilly Publication, 2013, ISBN 978-1449340377.

Problem solving using C Lab					
Prerequisite: :NiL		L	T	P	C
Total hours: 28		0	0	2	1
Course Content					Hrs
I	**The following proposed coverage is broad guiding areas lab. The programs mentioned here just sample programs and they are just for reference purpose. The instructor offering the course in consultation with the theory offered can adopt further variations in tune with concerned theory course** --- Expressions and Operators (2 Lab) Programs demonstrating arithmetic, relational, logical, assignment, increment/decrement, conditional (ternary), and bitwise operators. Applications such as roots of a quadratic equation, area and circumference of a circle, simple and compound interest, temperature conversion, and unit conversions. Decision Making and Looping (2 Labs) Programs using if, if-else, nested if, switch, for, while and do-while. Applications such as leap year, palindrome, Armstrong number, prime numbers, GCD/LCM, multiplication tables, series computation, and trigonometric series.					6
II	Arrays and Strings (2 Labs) Programs on one-dimensional and two-dimensional arrays, matrix operations, string handling, reversing strings, palindrome checking, word counting, and character frequency. Searching and Sorting Algorithms (3 Labs) Implementation of Linear Search, Binary Search, Bubble Sort, and Selection Sort. Applications involving insertion, deletion, and merging of arrays.				8
III	Functions and Recursion (2 Labs) Programs demonstrating user-defined functions, recursive functions, call by value, and call by reference. Applications such as factorial, Fibonacci series, GCD, power computation, binary search using functions, and menu-driven calculator. Pointers and Dynamic Memory Allocation (2 Labs) Programs using pointers, pointer arithmetic, arrays and pointers, arrays of pointers, pointer to functions, pointers to structures, and dynamic memory allocation using malloc(), calloc(), realloc(), and free(). Simple linked list implementation.				8
IV	Structures, Unions and Enumerations (2 Lab) Programs using structures, unions, nested structures, arrays of structures, typedef, and enum. Applications such as student record management and complex number operations. File Handling (1 Lab) Programs involving creation, reading, writing, appending, copying, and merging text files. Applications such as counting characters, words, and lines, searching data, and simple record management.				6

References	
1.	Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, 2nd Edition, Pearson Education, 1988.
2.	Paul Deitel and Harvey Deitel, C: How to Program, 9th Edition, Pearson, 2021
3.	Stephen G. Kochan, Programming in C, 4th Edition, Addison-Wesley Professional, 2014.
4.	K. N. King, C Programming: A Modern Approach, 2nd Edition, W. W. Norton & Company, 2008.

Data Structures Lab					
Prerequisite: :NiL		L	T	P	C
Total hours: 30		0	0	2	1
Course Content					Hrs
I	**The following proposed coverage provides broad guiding areas for the laboratory. The programs mentioned here are only sample programs, and instructors may include additional exercises to strengthen the understanding of data structures and algorithms.** Arrays and Linked Lists (3 Labs) Familiarization with the programming environment and implementation of Abstract Data Types (ADT). Programs on one-dimensional and two-dimensional arrays. Array operations: traversal, insertion, deletion, searching, and updating. Programs on sparse matrix representation and matrix operations. Implementation of singly, doubly, and circular linked lists. Operations on linked lists: insertion, deletion, searching, traversal, and applications.				6
II	Stacks and Queues (3 Labs) Stack implementation using arrays and linked lists. Applications of stacks: parenthesis matching, infix to postfix conversion, postfix evaluation, recursion simulation. Queue implementation using arrays and linked lists. Programs on circular queue, deque, and priority queue. Applications of queues such as CPU scheduling and simulation problems.				6
III	Searching, Sorting, and Hashing (3 Labs) Implementation of linear search and binary search algorithms. Programs on Bubble, Selection, Insertion, Merge, Quick, Heap, Counting, Bucket, and Radix sorting algorithms. Comparative analysis of sorting algorithms based on execution time. Implementation of hash tables using separate chaining, linear probing, quadratic probing, and double hashing.				7
IV	Trees (3 Labs) Binary tree creation and tree traversals (Preorder, Inorder, Postorder, and Level Order). Binary Search Tree (BST): insertion, deletion, and searching. AVL tree implementation with tree rotations. Heap implementation and Heap Sort. Demonstration/implementation of B-Tree and B+ Tree operations.				7
V	Graphs and Applications (2 Labs) Graph representation using adjacency matrix and adjacency list. Implementation of Depth-First Search (DFS) and Breadth-First Search (BFS). Programs on Dijkstra's shortest path algorithm. Programs on Prim's minimum spanning tree algorithm.				4

References	
1.	Horowitz, E., Sahni, S., & Anderson-Freed, S. (2008). Fundamentals of Data Structures in C (2nd ed.). Universities Press.
2.	Weiss, M. A. (2014). Data Structures and Algorithm Analysis in C (2nd ed.). Pearson Education.
3.	Thareja, R. (2014). Data Structures Using C (2nd ed.). Oxford University Press.
4.	Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). MIT Press.

Digital Logic and Microprocessor Lab				
Prerequisite: :NiL	L	T	P	C
Total hours: 40	0	0	2	1

Lab Course Content

1. Design and test a 2-bit and 4-bit half adder.
2. Design and test a 2-bit and 4-bit adder (ripple, carry look ahead).
3. Design and test of a 2-bit multiplier.
4. Design and test of n -bit comparator.
5. Design and test of flip flops – RS/JK/D/T.
6. Design and test of SISO and PIPO shift registers.
7. Design and test of counters.
8. Design and synthesis of pulse mode and clock mode circuits
9. Design and synthesis of clock mode circuits
10. Design and Simplification of incompletely specified sequential circuits
11. SPICE simulator based design and evaluation
12. X86 based Assembly language Programming
13. Athematic and logical operations
14. Number Conversion and String manipulation
15. Implementing a real-time clock.
16. Writing a program to control and monitor external devices through parallel port communication.
17. Creating a program to implement a basic graphical user interface (GUI) using assembly language.
18. Developing a program to interface with an external memory device, such as EEPROM or flash memory
19. Analysis of Timing Hazards and Critical Path
20. Memory Read/Write Cycles and Interrupt Handling in 8085/8086.
21. Register-Based versus Stack-Based Computation.

References:

1.	Herbert Taub, Donald L. Schilling, Digital Integrated Electronics, McGraw-Hill.
2.	Microprocessor Architecture, Programming and Applications with the 8085-Ramesh S. Gaonkar
3.	Fredrick J. Hill, Gerald R. Peterson, Computer aided logical design with emphasis on VLSI, Wiley
4.	M. Morris Mano, Digital Logic and Computer Design, Person Education.
5.	Lance A. Leventhal, S. Cordes, Assembly language subroutines for the 8086, McGraw-Hill Book Co.
6.	Randall Hyde, The Art of Assembly Language, 2nd Edition