

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

**Course Code: BSCS101**

**Course Title: Applied Mathematics**

|                                                                                    |                                                       |                                                           |
|------------------------------------------------------------------------------------|-------------------------------------------------------|-----------------------------------------------------------|
| <b>Programme:</b> B.Tech.                                                          | <b>L: 3 T: 1 P: 0</b>                                 | <b>Credits: 4</b>                                         |
| <b>Semester:</b> 3                                                                 | <b>Theory/Practical:</b> Theory                       | <b>Teaching Hours:</b> 45(L) + 15(T) = 60 hrs             |
| <b>Total Max. Marks:</b><br>100                                                    | <b>Continuous Assessment (CA)</b><br><b>Marks: 40</b> | <b>End Semester Examination (ESE)</b><br><b>Marks: 60</b> |
| <b>Minimum Percentage of Numerical / Design / Programming Problems in ESE: 95%</b> |                                                       |                                                           |
| <b>Duration of End Semester Examination (ESE): 3 hours</b>                         |                                                       |                                                           |
| <b>Course Type:</b> Core Course                                                    |                                                       |                                                           |

**Prerequisites (if any):** Knowledge of matrices, differentiation and integration and differential equations

**Additional Material Allowed in ESE: NIL**

**On completion of the course, the student will have the ability to:**

| <b>CO#</b> | <b>Course Outcomes</b>                                                                                     |
|------------|------------------------------------------------------------------------------------------------------------|
| 1          | Apply concepts of inner product spaces to solve problems involving orthogonalization Gram Schimdt process. |
| 2          | Use Laplace transforms to solve ordinary differential equations with initial conditions.                   |
| 3          | Apply Fourier integrals to solve problems in engineering.                                                  |
| 4          | Implement interpolation algorithm in computation tools for engineering applications.                       |
| 5          | Implement numerical techniques for solving algebraic equations and for performing numerical integration.   |

## Contents

### Part-A

#### Unit-1 Laplace Transforms

**11 (L)hrs**

Definition and existence of Laplace Transforms, Laplace transforms of various standard functions, properties of Laplace transforms, inverse Laplace transforms, transform of derivatives and integrals, Transform of multiplication and division by t, convolution theorem, Laplace transform of unit step function. Applications to solution of ordinary linear differential equations with constant coefficients, Applications to solve initial and boundary value problems.

# **Guru Nanak Dev Engineering College, Ludhiana**

**An Autonomous College under UGC Act 1956**

**B.Tech. (CSE), Scheme - 2024**

## **Unit-2 Fourier Series**

**12 (L)hrs**

Introduction, even and odd functions, periodic functions, Dirichlet's conditions for Fourier series, Euler's formulae for Fourier series expansion, change of interval, Half Range series expansions, Fourier series on arbitrary intervals, Application of Fourier series to solve different waveforms.

### **Part-B**

## **Unit-3 Numerical Methods**

**10 (L)hrs**

Bisection Method, Newton Raphson Method to solve the algebraic equations, Simpson's Method and Trapezoidal method to evaluate the definite integral, interpolation, forward and backward differences, Newton's Forward Interpolation Formula and Newton's Backward Interpolation formula.

## **Unit-4 Advanced Linear Algebra**

**12 (L)hrs**

Solution of system of linear equations, Linear Spaces, Vector Spaces, Vector Sub Spaces, Basis and dimension, Inner product, Gram Schmidt orthogonalization process, Eigen-values, Eigen-vectors and Diagonalisation, Projection matrix, orthogonal matrix, idempotent Matrix, Partition matrix, Nullity, LU decomposition.

*Tutorials may be conducted in lab settings and SageMath/ GeoGebra/ GNU Octave could be used to execute the lessons (as the case may be).*

### **Text Books**

1. B.S. Grewal, "Higher Engineering Mathematics", 44<sup>th</sup> edition, Khanna Publishers, 2021.
2. M.K. Jain, S.R.K. Iyengar, R.K. Jain, "Numerical Methods for Scientific and Engineering Computation", 6<sup>th</sup> edition, Newage International Publishers, 2003.

### **Reference Books**

1. Erwin Kreyszig, "Advanced Engineering Mathematics", 9<sup>th</sup> edition, Wiley India Pvt Ltd, 2016.
2. B.S. Grewal, "Numerical methods in Engineering and Science with Programmes in C, C++ & Matlab", 11<sup>th</sup>, Khanna Publishers, 2013
3. Axler, Sheldon, "Linear Algebra Done Right", 4<sup>th</sup>, Cham, Switzerland: Springer International Publishing, 2024.
4. Iyengar, T.K.V., B.Krishna Gandhi and S. Ranganatham & M.V.S.S.N. Prasad, "Laplace Transforms, Numerical Methods & Complex Variables", S. Chand Publishing, 2018.
5. Sreenadh, S, "Fourier Series and Integral Transforms", New Delhi, India: S Chand, 2014.
6. Atkinson, Kendall, Nashville, TN "An Introduction to Numerical Analysis", 2nd edition, John Wiley & Sons, 1989.

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

7. Strang, Gilbert, "Introduction to Linear Algebra", 6<sup>th</sup> edition, Wellesley, MA: Wellesley-Cambridge Press, 2023.

**Online Learning Materials**

1. [https://www.cimt.org.uk/projects/mepres/alevel/fstats\\_ch5.pdf](https://www.cimt.org.uk/projects/mepres/alevel/fstats_ch5.pdf)  
Accessed on April 12,2025
2. <https://www.cs.ox.ac.uk/files/12921/book.pdf>  
Accessed on April 14,2025
3. <https://www.youtube.com/watch?v=nH05UiErAX4>  
Accessed on April 12,2025
4. [https://www.youtube.com/watch?v=d7NF-\\_8vVv4&t=2s](https://www.youtube.com/watch?v=d7NF-_8vVv4&t=2s)  
Accessed on April 14,2025
5. <https://www.webpages.uidaho.edu/~stevel/519/Applied%20Multivariate%20Statistical%20Analysis%20by%20Johnson%20and%20Wichern.pdf>  
Accessed on April 14,2025
6. <http://ndl.ethernet.edu.et/bitstream/123456789/55096/1/Tsuneo%20Arakawa.pdf>  
Accessed on April 14,2025

**Supplementary SWAYAM Course**

| Sr. No. | Course Name                           | Instructor                      | Host Institute               | URL                                                                                                                     |
|---------|---------------------------------------|---------------------------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| 1       | Linear Algebra                        | Prof. Pranav Haridas            | Kerala School of Mathematics | <a href="https://onlinecourses.nptel.ac.in/noc25_ma37/preview">https://onlinecourses.nptel.ac.in/noc25_ma37/preview</a> |
| 2       | Numerical Analysis                    | Prof. S. Baskar                 | IIT Bombay                   | <a href="https://onlinecourses.nptel.ac.in/noc25_ma41/preview">https://onlinecourses.nptel.ac.in/noc25_ma41/preview</a> |
| 3       | Fourier Analysis and its applications | Prof. Gopala Krishna Srinivasan | IIT Bombay                   | <a href="https://onlinecourses.nptel.ac.in/noc25_ma23/preview">https://onlinecourses.nptel.ac.in/noc25_ma23/preview</a> |

# **Guru Nanak Dev Engineering College, Ludhiana**

**An Autonomous College under UGC Act 1956**

**B.Tech. (CSE), Scheme - 2024**

**Course Code: CCS101**

**Course Title: Object Oriented Programming**

|                                                                                    |                                                                |                                                 |
|------------------------------------------------------------------------------------|----------------------------------------------------------------|-------------------------------------------------|
| <b>Programme:</b> B.Tech.                                                          | <b>L: 2 T: 0 P: 4</b>                                          | <b>Credits: 4</b>                               |
| <b>Semester:</b> 3                                                                 | <b>Theory/Practical:</b><br>Theory                             | <b>Teaching Hours:</b> 30(L) + 60 (P) = 90 hrs  |
| <b>Total Max. Marks:</b> 150                                                       | <b>Continuous Assessment Marks:</b> 90<br><b>Internal (CA)</b> | <b>End Semester Examination (ESE) Marks:</b> 60 |
| <b>Minimum Percentage of Numerical / Design / Programming Problems in ESE: 30%</b> |                                                                |                                                 |
| <b>Duration of End Semester Examination (ESE): 3 hours</b>                         |                                                                |                                                 |
| <b>Course Type:</b> Core Course                                                    |                                                                |                                                 |

**Prerequisites (if any):** Basics of Programming

**On completion of the course, the student will have the ability to:**

| <b>CO#</b> | <b>Course Outcomes</b>                                                               |
|------------|--------------------------------------------------------------------------------------|
| 1          | Demonstrate programming skills in writing, compiling and debugging C++ programs.     |
| 2          | Use class and object to implement object-oriented approach.                          |
| 3          | Apply inheritance and polymorphism to implement object-oriented programs.            |
| 4          | Analyse the relevance of dynamic memory management for effective memory utilization. |
| 5          | Make use of exception handling and file handling in program development.             |
| 6          | Develop solution for programming problems using object-oriented principles.          |

## **Contents**

### **Part-A**

#### **Unit-1 Object Oriented Programming Concepts**

**4(L) hrs**

Introduction, Comparison between Procedure oriented approach and Object-oriented approach, Basic data types, Derived data types, Keywords, Identifiers, Constants and variables, Type casting, Operators, and Operator precedence. Control Structures- if statement, switch-case, for, while and do-while loops, break and continue statement. Features of object-oriented programming- Class, Objects, Encapsulation, Abstraction, Data hiding, Polymorphism and Inheritance.

#### **Unit-2 Classes and Objects**

**5(L) hrs**

# **Guru Nanak Dev Engineering College, Ludhiana**

**An Autonomous College under UGC Act 1956**

**B.Tech. (CSE), Scheme - 2024**

Implementation of a class, Class objects, Access specifiers, Accessing class members, Constructor and destructor, Types of constructors, Empty class, Nested class, Friend function, Inline function, Friend class, Static members.

## **Unit-3 Inheritance**

**5(L) hrs**

Introduction, Defining derived classes, Types of inheritance, Ambiguity in inheritance, Virtual base class, Objects slicing, Overriding member functions.

## **Part-B**

## **Unit -4 Dynamic Memory Management using Pointers**

**5(L) hrs**

Declaring and initializing pointers, Accessing data through pointers, Pointer arithmetic, Memory allocation – Comparison of Static and Dynamic, Dynamic memory management using new and delete operators, Pointer to an object, this pointer, Pointer related problems – Dangling/wild pointers, Null pointer assignment, Memory leak and Allocation failures.

## **Unit-5 Polymorphism and Type Conversion**

**6(L) hrs**

Concept of binding – Early binding and late binding, Virtual functions, Pure virtual functions, Abstract class. Function overloading, Constructor overloading, Operator Overloading, Rules for overloading operators, Overloading of various operators. Type conversion – Basic type to class type, Class type to basic type, Class type to another class type.

## **Unit-6 Exceptions Handling and File Handling**

**5(L)hrs**

Review of traditional error handling, Basics of exception handling, Exception handling mechanism, Throwing and catching mechanism, Rethrowing an exception, Specifying exceptions. File streams, Error handling during file operations, Reading/writing of files, accessing records randomly, Updating files.

## **Laboratory Work**

In the laboratory work, students will implement programs related to above topics using any open-source C++ tools, such as Code Blocks, Dev-C++, VS Code etc.

**Following is only the suggested list of Practical's. Instructor may frame additional Practical's relevant to the course contents.**

| <b>Experiment No.</b> | <b>Experiment Title</b>                                            |
|-----------------------|--------------------------------------------------------------------|
| 1.                    | Program to illustrate the use of class and object.                 |
| 2.                    | Program to use decision control statements using class and object. |
| 3.                    | Program to use Loop control statements using class and object.     |

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

|     |                                                                                      |
|-----|--------------------------------------------------------------------------------------|
| 4.  | Program to illustrate the concept of nesting of member functions.                    |
| 5.  | Program to illustrate the concept of default parameters in a function.               |
| 6.  | Program to implement a local class.                                                  |
| 7.  | Program to illustrate the concept of nested class.                                   |
| 8.  | Program to illustrate the concept of inline function.                                |
| 9.  | Program to illustrate the concept of friend function.                                |
| 10. | Program to illustrate the concept of empty class.                                    |
| 11. | Program to show the working of static data members in a class.                       |
| 12. | Program to show the working of static functions in a class.                          |
| 13. | Program to illustrate the concept of friend class.                                   |
| 14. | Program to illustrate the concept of default constructor.                            |
| 15. | Program to illustrate the use of parameterized constructor.                          |
| 16. | Program to illustrate the use of copy constructor.                                   |
| 17. | Program to illustrate the concept of dynamic constructor.                            |
| 18. | Program to illustrate the concept of destructors.                                    |
| 19. | Program to illustrate the concept of single inheritance.                             |
| 20. | Program to illustrate the concept of multilevel inheritance.                         |
| 21. | Program to illustrate the concept of multiple inheritance.                           |
| 22. | Program to illustrate the concept of hierarchical inheritance.                       |
| 23. | Program to illustrate the concept of hybrid inheritance.                             |
| 24. | Program to illustrate the concept of ambiguity in single inheritance.                |
| 25. | Program to illustrate the concept of ambiguity in multiple inheritance.              |
| 26. | Program to illustrate the concept of virtual base class/multipath inheritance.       |
| 27. | Program to illustrate the concept of dynamic memory management using new and delete. |
| 28. | Program to illustrate the use of this pointer.                                       |
| 29. | Program to show pointer arithmetic operations.                                       |
| 30. | Program to overload unary minus operator.                                            |
| 31. | Program to overload prefix and postfix increment operators.                          |
| 32. | Program to overload binary addition operator.                                        |
| 33. | Program to add two complex numbers using binary operator overloading.                |
| 34. | Program to illustrate the use of virtual function.                                   |
| 35. | Program to illustrate the concept of abstract class.                                 |
| 36. | Program to illustrate basic to class type conversion.                                |
| 37. | Program to illustrate class to basic type conversion.                                |

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 38. | Program to illustrate class to class type conversion using constructor.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 39. | Program to illustrate class to class type conversion using casting operator.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 40. | Program to illustrate the concept of exception handling for throw point inside the try block.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 41. | Program to illustrate the concept of exception handling for throw point outside the try block.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 42. | Program to illustrate the concept of exception handling for multiple catch statements.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 43. | Program to illustrate the concept of exception handling for rethrowing an exception.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 44. | Program to illustrate the concept of exception handling with single catch statement for all exceptions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 45. | Program to perform read and write operations on file using get(), put().                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 46. | Program to perform read and write operations on file using read(), write().                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 47. | Program to illustrate the concept of file pointers.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 48. | <p>Any one project</p> <ol style="list-style-type: none"> <li>i. Banking System: The Banking System project has account class with data members like account number, name, deposit, withdraw amount and type of account. Customer data is stored in a binary file. A customer can deposit and withdraw amount in his account. User can create, modify and delete account.</li> <li>ii. Library Management System: Creating a simple text-based system for managing library. It allows users (admin/librarian) to add new books, view book details, and update availability status. Users should be able to issue books to students (marking them as issued) and return them. It also maintains details such as book title, author, ID and availability. Implement using file handling to preserve data like issued books and book inventory even after closing the application.</li> <li>iii. Student Grade Management System: Design console-based application that allows the user (e.g., a teacher) to enter student names, roll numbers, and their marks for various subjects. The program will then calculate total marks, average percentage, and assign a grade (A, B, C, etc.) based on predefined criteria. It should also be able to display the individual student's report card and allow editing or viewing student records. Add file handling to save and retrieve student data between sessions.</li> <li>iv. Simple ATM Interface: This project simulates a basic ATM interface that requires users to enter a PIN for authentication. Once logged in, the user can perform operations such as checking balance, withdrawing cash and depositing money. The account details (like current balance) should be</li> </ol> |

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

|  |                                                                                                                                                                                                       |
|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>stored and updated after every transaction. It should work for multiple user accounts and use file handling to save data persistently.</p> <p>Any other project recommended by the instructor.</p> |
|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Text Books**

1. E. Balagurusamy, “Object Oriented Programming with C++”, 8<sup>th</sup> edition, Tata McGraw Hill, 2020.
2. P Yashavant Kanetkar, “Let Us C++”, 3<sup>rd</sup> edition, BPB Publications, 2019.

**Reference Books**

1. Paul Deitel and Harvey Deitel, “C++ How to Program”, 10<sup>th</sup> edition, Pearson, 2016.
2. R. Lafore, “Object Oriented Programming in C++”, 4<sup>th</sup> edition, Waite Group, 2002.
3. Herbert Schildt, “The Complete Reference to C++ Language”, 4<sup>th</sup> edition, McGraw Hill-Osborne, 2017.

**Online Learning Materials**

1. <https://faculty.iitr.ac.in/cs/bala/OOPS/csn103.html>  
Accessed on 24-04-2025
2. [https://www.youtube.com/watch?v=iw1Xf\\_33YM0&list=PLQEaRBV9gAFujcBWJhBT2XXsuMlIfETBy](https://www.youtube.com/watch?v=iw1Xf_33YM0&list=PLQEaRBV9gAFujcBWJhBT2XXsuMlIfETBy)  
Accessed on 24-04-2025
3. <https://www.youtube.com/watch?v=qq3BY4viEB4&list=PLQEaRBV9gAFujcBWJhBT2XXsuMlIfETBy&index=4>  
Accessed on 24-04-2025
4. <https://www.youtube.com/watch?v=p2h8rGnkD0o&list=PLQEaRBV9gAFujcBWJhBT2XXsuMlIfETBy&index=6>  
Accessed on 24-04-2025
5. <https://www.youtube.com/watch?v=NBsmPHXjLfg&list=PLQEaRBV9gAFujcBWJhBT2XXsuMlIfETBy&index=8>  
Accessed on 24-04-2025

**Supplementary SWAYAM Course**

| Sr. No. | Course Name                                 | Instructor                  | Host Institute | URL                                                                                                                     |
|---------|---------------------------------------------|-----------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------|
| 1       | Fundamentals of Object-Oriented Programming | Prof. Balasubramanian Raman | IIT Roorkee    | <a href="https://onlinecourses.nptel.ac.in/noc25_cs34/preview">https://onlinecourses.nptel.ac.in/noc25_cs34/preview</a> |
| 2       | Programming in C++                          | Prof. Partha Pratim Das     | IIT Kharagpur  | <a href="https://onlinecourses.nptel.ac.in/noc20_cs07/preview">https://onlinecourses.nptel.ac.in/noc20_cs07/preview</a> |



# **Guru Nanak Dev Engineering College, Ludhiana**

**An Autonomous College under UGC Act 1956**

**B.Tech. (CSE), Scheme - 2024**

|   |                              |                            |                  |                                                                                                                              |
|---|------------------------------|----------------------------|------------------|------------------------------------------------------------------------------------------------------------------------------|
| 3 | Programming in<br>Modern C++ | Prof. Partha<br>Pratim Das | IIT<br>Kharagpur | <a href="https://onlinecourses.nptel.ac.in/noc25_cs58/preview">https://onlinecourses.nptel.ac.in/<br/>noc25_cs58/preview</a> |
|---|------------------------------|----------------------------|------------------|------------------------------------------------------------------------------------------------------------------------------|

# **Guru Nanak Dev Engineering College, Ludhiana**

**An Autonomous College under UGC Act 1956**

**B.Tech. (CSE), Scheme - 2024**

**Course Code: CCS102**

**Course Title: Data Structures**

|                                                                                    |                                                                |                                                        |
|------------------------------------------------------------------------------------|----------------------------------------------------------------|--------------------------------------------------------|
| <b>Programme:</b> B.Tech.                                                          | <b>L: 3 T: 1 P: 4</b>                                          | <b>Credits: 6</b>                                      |
| <b>Semester:</b> 3                                                                 | <b>Theory/Practical:</b><br>Theory                             | <b>Teaching Hours:</b> 45(L) + 15(T) + 60(P) = 120 hrs |
| <b>Total Max. Marks:</b><br>150                                                    | <b>Continuous Assessment Marks: 90</b><br><b>Internal (CA)</b> | <b>End Semester Examination (ESE) Marks: 60</b>        |
| <b>Minimum Percentage of Numerical / Design / Programming Problems in ESE: 50%</b> |                                                                |                                                        |
| <b>Duration of End Semester Examination (ESE): 3 hours</b>                         |                                                                |                                                        |
| <b>Course Type:</b> Core Course                                                    |                                                                |                                                        |

**Prerequisites (if any):** Basics of Programming

**On completion of the course, the student will have the ability to:**

| <b>CO#</b> | <b>Course Outcomes</b>                                                                                |
|------------|-------------------------------------------------------------------------------------------------------|
| 1          | Apply Arrays and Linked Lists to solve computational problems.                                        |
| 2          | Implement stack and queue data structures to solve programming problems.                              |
| 3          | Apply appropriate indexing and searching techniques to optimize data storage and retrieval.           |
| 4          | Analyze non-linear data structures to perform efficient storage, retrieval, and traversal operations. |
| 5          | Analyze time and space complexity of different sorting algorithms.                                    |
| 6          | Identify the appropriate data structure to provide solution with reduced space and time complexity.   |

## **Contents**

### **Part-A**

#### **Unit-1 Basic concepts and Arrays**

**5(L) hrs**

Concept of data type, Linear and non-linear data structures, Data structures versus data types, Operations on data structures, Algorithm complexity and Asymptotic notations. Linear and multi-dimensional arrays and their representation, Operations on arrays, Sparse matrices and their storage.

#### **Unit-2 Searching and Sorting**

**4(L) hrs**

Linear and binary search techniques, Sorting methods – Bubble sort, Selection sort, Insertion sort, Merge sort, Shell sort and radix sort. Complexities of searching and sorting algorithms.

# **Guru Nanak Dev Engineering College, Ludhiana**

**An Autonomous College under UGC Act 1956**

**B.Tech. (CSE), Scheme - 2024**

## **Unit -3 Linked List**

**7(L) hrs**

Definition and representation of Linked list, Types of Linked list- Linear linked list, Doubly linked list, Circular linked list and Header linked list and their operations, Application of linked lists.

## **Unit -4 Stacks**

**7(L) hrs**

Sequential and Linked representation of stacks, Operations on stacks, Application of stacks – parenthesis checker, Evaluation of postfix expressions, Conversion from infix to postfix, Conversion from infix to prefix representation, Tower of Hanoi problem, implementing recursive functions, Quick sort.

## **Part-B**

## **Unit-5 Queues**

**4(L) hrs**

Sequential and Linked representation of queue, Types of queue- Linear Queue, Circular Queue, Dequeue, Priority Queue, Operations on each types of Queues and their algorithms, Applications of Queues.

## **Unit-6 Trees**

**7(L)hrs**

Basic terminology, Sequential and linked representations of trees, Tree Traversals, Different types of Trees- Binary Tree, Binary search tree, Threaded binary tree, AVL tree, B-tree, B<sup>+</sup>- tree, Red Black tree. Operations on each of the trees. Application of Binary Trees.

## **Unit-7 Heaps**

**2(L)hrs**

Representing a heap in memory, Operations on heaps, Application of heap in implementing priority queue and Heap sort algorithm.

## **Unit-8 Graphs**

**6(L)hrs**

Basic terminology, Representation of graphs – Adjacency matrix, Adjacency list. Operations on graph, Traversal of a graph – Breadth first search, Depth first search. Shortest path algorithms - Dijkstra's and Floyd-Warshall algorithm. Minimum spanning tree – Prim and Kruskal. Applications of graphs.

## **Unit-9 Hashing and Hash Tables**

**3(L)hrs**

Introduction to hash table, Hash functions, Concept of collision and its resolution using open addressing and separate chaining, Double hashing, Rehashing.

*Tutorial hours will be used for practice sessions for numerical/programming problems and design of algorithms related to above contents.*

**Guru Nanak Dev Engineering College, Ludhiana**  
An Autonomous College under UGC Act 1956  
**B.Tech. (CSE), Scheme - 2024**

**Laboratory Work**

In the laboratory work, students should implement these data structures using any open-source C++ tools, such as Code Blocks, Dev-C++, VS Code etc.

**Following is only the suggested list of Practical's. Instructor may frame additional Practical's relevant to the course contents.**

| <b>Experiment No.</b> | <b>Experiment Title</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1                     | Write a program to add two matrices.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 2                     | Design, Develop and Implement a menu driven Program for the following Array operations<br>a. Display of Array Elements with Suitable Headings<br>b. Inserting an Element (ELEM) at a given valid Position (POS)<br>c. Deleting an Element at a given valid Position (POS)<br>d. Exit.                                                                                                                                                                                                                                               |
| 3                     | Write a Program to find the position of an element in an array using Linear Search Algorithm.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 4                     | Write a Program to find the position of an element in an array using Binary search Algorithm.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 5                     | Write a program to sort list using bubble sort.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 6                     | Write a program to sort list using selection sort.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 7                     | Write a program to sort list using insertion sort.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 8                     | Write a program to sort list using merge sort.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 9                     | Design, Develop and Implement a menu driven Program for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Branch, Sem, Phone Number<br>a. Insert new student at beginning of list.<br>b. Insert new student at end of list.<br>c. Insert new student in the middle of the list.<br>d. Display all records and count the number of nodes in it.<br>e. Delete first student from the list.<br>f. Delete last student from the list<br>g. Delete nth student from the list.<br>h. Exit. |

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

| <b>Experiment No.</b> | <b>Experiment Title</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10                    | Design, Develop and Implement a Program to perform insertion, deletion and traversing in a sorted Singly Circular Linked List (SCLL) with header nodes.                                                                                                                                                                                                                                                                                                                                                                                         |
| 11                    | Design, Develop and Implement a menu driven Program for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, Phone Number<br>a. Insert new student at beginning of list.<br>b. Insert new student at end of list.<br>c. Insert new student in the middle of the list.<br>d. Display all records and count the number of nodes in it.<br>e. Delete first student from the list.<br>f. Delete last student from the list<br>g. Delete nth student from the list.<br>h. Exit. |
| 12                    | Implement a Program to find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z)                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 13                    | Design, Develop and Implement a menu driven Program for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX)<br>a. Push an Element on to Stack<br>b. Pop an Element from Stack<br>d. Display the elements of Stack<br>e. Exit                                                                                                                                                                                                                                                                    |
| 14                    | Design, Develop and Implement a menu driven Program for the following operations on STACK of Integers (Linked Implementation)<br>a. Push an Element on to Stack<br>b. Pop an Element from Stack<br>d. Display the elements of Stack<br>e. Exit                                                                                                                                                                                                                                                                                                  |
| 15                    | Design and implement a program to check if a given number is Palindrome or not using stack.                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 16                    | Design, Develop and Implement a Program for evaluating Suffix expression with single digit operands and operators: +, -, *, /, %, ^.                                                                                                                                                                                                                                                                                                                                                                                                            |

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

| <b>Experiment No.</b> | <b>Experiment Title</b>                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 17                    | Design, Develop and Implement a Program for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^(Power) and alphanumeric operands                                                                                                                                                                                         |
| 18                    | Design, Develop and Implement a Program for solving Tower of Hanoi problem with n disks                                                                                                                                                                                                                                                                                                                                                                      |
| 19                    | Write a program to sort list using quicksort.                                                                                                                                                                                                                                                                                                                                                                                                                |
| 20                    | Design, Develop and Implement a menu driven Program for the following operations on Linear QUEUE (Array Implementation of Queue with maximum size MAX)<br>Insert an Element on to Linear QUEUE<br>Delete an Element from Linear QUEUE<br>Display the elements of Linear QUEUE<br>Exit<br>Support the program with appropriate functions for each of the above operations.                                                                                    |
| 21                    | Design, Develop and Implement a menu driven Program for the following operations on QUEUE (Linked Implementation)<br>Insert an Element on to QUEUE<br>Delete an Element from QUEUE<br>Display the elements of QUEUE<br>Exit<br>Support the program with appropriate functions for each of the above operations.                                                                                                                                              |
| 22                    | Design, Develop and Implement a menu driven Program for the following operations on Circular QUEUE (Array Implementation of Queue with maximum size MAX)<br>a. Check Overflow and Underflow situations on Circular QUEUE<br>b. Insert an Element on to Circular QUEUE<br>c. Delete an Element from Circular QUEUE<br>d. Display the elements of Circular QUEUE<br>e. Exit<br>Support the program with appropriate functions for each of the above operations |

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

| <b>Experiment No.</b> | <b>Experiment Title</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 23                    | <p>Design, Develop and Implement a menu driven Program for the following operations on Dequeue QUEUE (Array Implementation of Queue with maximum size MAX)</p> <ol style="list-style-type: none"> <li>Check Overflow and Underflow situations on Dequeue QUEUE</li> <li>Insert an Element on to Dequeue QUEUE</li> <li>Delete an Element from Dequeue QUEUE</li> <li>Display the status of Dequeue QUEUE</li> <li>Exit</li> </ol> <p>Support the program with appropriate functions for each of the above operations</p> |
| 24                    | <p>Design, Develop and Implement a menu driven Program for the following operations on Priority QUEUE (Array Implementation of Queue with maximum size MAX)</p> <p>Insert an Element on to Priority QUEUE</p> <p>Delete an Element from Priority QUEUE</p> <p>Display the status of Priority QUEUE</p> <p>Exit</p> <p>Support the program with appropriate functions for each of the above operations</p>                                                                                                                |
| 25                    | <p>Design, Develop and Implement a menu driven Program for the following operations on Binary Search Tree (BST) of Integers.</p> <ol style="list-style-type: none"> <li>Insertion in a BST.</li> <li>Traverse the BST in Inorder, Preorder and Post Order.</li> <li>Search an element in BST</li> <li>Deletion in BST</li> <li>Exit</li> </ol>                                                                                                                                                                           |
| 26                    | <p>Design, Develop and Implement a menu driven Program for the following operations on AVL Tree of Integers.</p> <p>Insertion in an AVL Tree.</p> <p>Search an element in AVL</p> <p>Deletion in AVL</p> <p>Exit</p>                                                                                                                                                                                                                                                                                                     |
| 27                    | Write a program to sort list using Heapsort.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

**Guru Nanak Dev Engineering College, Ludhiana**  
An Autonomous College under UGC Act 1956  
**B.Tech. (CSE), Scheme - 2024**

| Experiment No. | Experiment Title                                                                                                                                                                                                                    |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28             | Design, Develop and Implement a menu driven Program for the following operations on Graph(G) of Cities<br>a. Create a Graph of N cities using Adjacency Matrix<br>b. Traverse Graph using DFS<br>c. Traverse Graph using BFS method |
| 29             | Design, Develop and Implement a program to construct spanning tree using Kruskal's and Prim's algorithm                                                                                                                             |
| 30             | Design, Develop and Implement a program to find shortest path using Dijkstra's algorithm.                                                                                                                                           |
| 31             | Construct a program to implement Hash Functions and Collision Resolution Technique                                                                                                                                                  |

*Students are required to make one mini-project based on above data structures.*

**Text Books**

1. Seymour Lipschutz, "Schaum's Outline of Data Structures", McGraw-Hill Education, 1st edition, 1986.
2. Sartaj Sahni, "Data Structures, Algorithms, and Applications in C++", 2nd edition, Silicon Press, 2004.

**Reference Books**

1. Michael T. Goodrich, Roberto Tamassia, and David M. Mount, "Data Structures and Algorithms in C++", 2<sup>nd</sup> Edition, Wiley, 2011.
2. Robert L. Kruse, "Data Structures and Program Design in C", 2<sup>nd</sup> Edition, Prentice Hall, 1996.
3. Yedidyah Langsam, Moshe J. Augenstein, and Aaron M. Tenenbaum, "Data Structures Using C and C++", 2<sup>nd</sup> edition, Prentice Hall, 1995.
4. Vishal Goyal, Lalit Goyal and Pawan Kumar, "Simplified Approach to Data Structures", 1<sup>st</sup> edition, Shroff Publishers, 2014.

**E-Books and online learning material**

1. Data Structures and Algorithms: by Granville Barnett, and Luca Del Tongo.  
<https://www.mta.ca/~rrosebru/oldcourse/263114/Dsa.pdf> Accessed on 15-05-2025
2. Data Structures and Algorithms in JAVA: by Michael T. Goodrich and Roberto Scheme and Tamassia  
[https://eduardmandov.wordpress.com/wp-content/uploads/2017/05/c\\_c-data-structures-and-algorithms-in-c.pdf](https://eduardmandov.wordpress.com/wp-content/uploads/2017/05/c_c-data-structures-and-algorithms-in-c.pdf) Accessed on 15-05-2025



**Guru Nanak Dev Engineering College, Ludhiana**  
An Autonomous College under UGC Act 1956  
**B.Tech. (CSE), Scheme - 2024**

**Online Courses and Video Lectures**

- |                                                                                              |                        |
|----------------------------------------------------------------------------------------------|------------------------|
| 1. <a href="https://nptel.ac.in/courses/106102064">https://nptel.ac.in/courses/106102064</a> | Accessed on 15-05-2025 |
| 2. <a href="https://nptel.ac.in/courses/106106127">https://nptel.ac.in/courses/106106127</a> | Accessed on 15-05-2025 |
| 3. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a> | Accessed on 15-05-2025 |
| 4. <a href="https://nptel.ac.in/courses/106106133">https://nptel.ac.in/courses/106106133</a> | Accessed on 15-05-2025 |

**Supplementary SWAYAM Course**

| Sr. No | Course Name     | Instructor        | Host Institute                                         | URL                                                                                                                       |
|--------|-----------------|-------------------|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| 1      | Data Structures | Dr.S.Sasikala     | Institute of distance Education University of Madras   | <a href="https://onlinecourses.swayam2.ac.in/cec19cs04/preview">https://onlinecourses.swayam2.ac.in/cec19cs04/preview</a> |
| 2      | Data Structures | Dr. M. Deivamani, | College of Engineering, Guindy, (CEG), Anna University | <a href="https://onlinecourses.swayam2.ac.in/cec25mal5/preview">https://onlinecourses.swayam2.ac.in/cec25mal5/preview</a> |

# **Guru Nanak Dev Engineering College, Ludhiana**

**An Autonomous College under UGC Act 1956**

**B.Tech. (CSE), Scheme - 2024**

**Course Code: CCS103**

**Course Title: Computer Architecture**

|                                                                                    |                                                      |                                                 |
|------------------------------------------------------------------------------------|------------------------------------------------------|-------------------------------------------------|
| <b>Programme:</b> B.Tech.                                                          | <b>L: 3 T: 1 P: 0</b>                                | <b>Credits: 4</b>                               |
| <b>Semester: 3</b>                                                                 | <b>Theory/Practical:</b><br>Theory                   | <b>Teaching Hours: 45(L) + 15(T) = 60 hrs</b>   |
| <b>Total Max. Marks: 100</b>                                                       | <b>Continuous Internal Assessment (CA) Marks: 40</b> | <b>End Semester Examination (ESE) Marks: 60</b> |
| <b>Minimum Percentage of Numerical / Design / Programming Problems in ESE: 30%</b> |                                                      |                                                 |
| <b>Duration of End Semester Examination (ESE): 3 hours</b>                         |                                                      |                                                 |
| <b>Course Type: Core Course</b>                                                    |                                                      |                                                 |

**Prerequisites (if any): NIL**

**On completion of the course, the student will have the ability to:**

| <b>CO#</b> | <b>Course Outcomes</b>                                                                           |
|------------|--------------------------------------------------------------------------------------------------|
| 1          | Understand the Data representation, Register Transfer and micro-operations in computer system.   |
| 2          | Illustrate the CPU architecture and I/O interfacing for different modes of transfer.             |
| 3          | Analyze memory hierarchy in computer system for efficient memory management.                     |
| 4          | Understand the organization of components of a computer system and their interconnectivity.      |
| 5          | Develop assembly language programs using 8085 instruction set to perform fundamental operations. |
| 6          | Interface peripheral devices with microprocessors using 8255, 8259, and 8237.                    |

## **Contents**

### **Part-A**

#### **Unit-1 Data Representation**

**4(L) hrs**

Data types, Complements, Fixed point representation, IEEE 754 Floating point representation (32bit/64bit), Error detection and correction.

#### **Unit-2 Register Transfer and Micro-operations**

**8(L) hrs**

Addition, Subtraction, Multiplication and division algorithms and hardware, Three-state buffer, Binary Adder, Binary Incrementor, Register-transfer language and operations, Arithmetic micro-operations, Logic micro-operations, Shift micro-operations, Arithmetic logic shift unit.

#### **Unit-3 Computer Organization and Design**

**8(L) hrs**

# **Guru Nanak Dev Engineering College, Ludhiana**

**An Autonomous College under UGC Act 1956**

**B.Tech. (CSE), Scheme - 2024**

Instruction codes - Direct and Indirect Address, Computer registers, Computer instructions, Timing and control, Instruction cycle, Instruction format, Memory, register, and input-output reference instructions, Input/ Output and interrupts, Design and working of a complete basic computer, Control functions, Design of accumulator logic.

## **Unit-4 Memory Organization**

**4(L) hrs**

Memory hierarchy, High-speed memories, Main Memory, Cache memory, Associative memory, Memory management techniques.

## **Part-B**

## **Unit-5 Central Processing Unit and Input-Output Organization**

**8(L) hrs**

General register organization, Stack organization, RISC and CISC architecture, I/O interface and types, Modes of transfer, DMA.

## **Unit-6 8085 Microprocessor Architecture and Programming**

**10(L) hrs**

Introduction to microprocessors, 8085 microprocessor architecture Pin Diagram, Bus structure, Addressing modes, Instruction classification, Instruction formats, Data transfer operations, Arithmetic operations, Logical operations, Branch operations, Stack and subroutine operations.

## **Unit-7 Interfacing**

**3(L) hrs**

8255 Programmable Peripheral Interface, 8259 interrupt controller and 8237 DMA controller.

*Tutorial hours will be used for practical sessions of Assembly Language Programming of 8085 microprocessor using 8085 trainer kits and practice problems related to the course.*

## **Text Books**

1. M. Morris Mano, "Computer System Architecture", 3<sup>rd</sup> edition, Pearson Education, 1993.
2. William Stallings, "Computer Organization and Architecture", 11<sup>th</sup> edition, Pearson Education, 2019.
3. D.A. Patterson and J.L. Hennessy, "Computer Architecture: A Quantitative Approach", 5<sup>th</sup> edition, Morgan Koffman, 2011.
4. Ramesh S. Gaonkar, "Microprocessor Architecture, Programming, and Applications with 8085", 6<sup>th</sup> edition, Penram International Publication, Prentice Hall, 2002.

## **Reference Books**

1. D.A. Patterson and J.L. Hennessy, "Computer Organization and Design: The Hardware/Software Interface", 5<sup>th</sup> edition, Elsevier India, 2016.
2. C. Hamacher, Z. Vranesic and S. Zaky, "Computer Organization", 5<sup>th</sup> edition, McGraw Hill, 2011.

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

3. B. Ram, “Fundamentals of Microprocessors and Microcomputers”, 7<sup>th</sup> edition, Dhanpat Rai Publications, 2011.

**Online Learning Materials**

1. <https://www.cse.iitd.ac.in/~srsarangi/archbooksoft.html>  
Accessed on March 20, 2025.
2. <https://www.youtube.com/watch?v=UpU5svo87Oo>  
Accessed on March 20, 2025.
3. [https://en.wikipedia.org/wiki/Computer\\_architecture](https://en.wikipedia.org/wiki/Computer_architecture)  
Accessed on March 20, 2025.
4. <https://www.coursera.org/learn/comparch>  
Accessed on March 20, 2025.

**Supplementary SWAYAM Course**

| Sr. No. | Course Name                          | Instructor                     | Host Institute | URL                                                                                                                     |
|---------|--------------------------------------|--------------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------|
| 1.      | Computer Architecture                | By Prof. Smruti Ranjan Sarangi | IIT Delhi      | <a href="https://onlinecourses.nptel.ac.in/noc23_cs67/preview">https://onlinecourses.nptel.ac.in/noc23_cs67/preview</a> |
| 2.      | Microprocessors and Microcontrollers | Prof. Santanu Chattopadhyay    | IIT Kharagpur  | <a href="https://onlinecourses.nptel.ac.in/noc25_ee49/preview">https://onlinecourses.nptel.ac.in/noc25_ee49/preview</a> |

# **Guru Nanak Dev Engineering College, Ludhiana**

**An Autonomous College under UGC Act 1956**

**B.Tech. (CSE), Scheme - 2024**

**Course Code: ESCS101**

**Course Title: Digital Electronics**

|                                                                                    |                                                      |                                                 |
|------------------------------------------------------------------------------------|------------------------------------------------------|-------------------------------------------------|
| <b>Programme:</b><br>B.Tech.                                                       | <b>L: 3 T: 0 P: 2</b>                                | <b>Credits: 4</b>                               |
| <b>Semester: 3</b>                                                                 | <b>Theory/Practical: Theory</b>                      | <b>Teaching Hours: 45 (L) + 30 (P)= 75 hrs</b>  |
| <b>Total Max. Marks: 150</b>                                                       | <b>Continuous Internal Assessment (CA) Marks: 90</b> | <b>End Semester Examination (ESE) Marks: 60</b> |
| <b>Minimum Percentage of Numerical / Design / Programming Problems in ESE: 50%</b> |                                                      |                                                 |
| <b>Duration of End Semester Examination (ESE): 3 hours</b>                         |                                                      |                                                 |
| <b>Course Type: Core Course</b>                                                    |                                                      |                                                 |

**Prerequisites (if any):** Basics of Electrical and Electronics Engineering

**On completion of the course, the student will have the ability to:**

| <b>CO#</b> | <b>Course Outcomes</b>                                                                                                          |
|------------|---------------------------------------------------------------------------------------------------------------------------------|
| 1          | Apply the structure of number systems in digital design.                                                                        |
| 2          | Minimize the Boolean expressions in SOP and POS form using K-maps.                                                              |
| 3          | Apply knowledge of basic characteristics of logic families to design digital circuits.                                          |
| 4          | Demonstrate the basic knowledge of behavioral objects of VHDL in programming.                                                   |
| 5          | Implement combinational logic circuits using Boolean algebra and logic gates and simulate them using VHDL.                      |
| 6          | Analyze Synchronous and Asynchronous sequential circuits using Flip Flops, registers and Counters and simulate them using VHDL. |

## **Contents**

### **Part-A**

#### **Unit-1 Number system**

**7 (L) hrs**

Introduction to number system - Binary, Octal, Decimal, Hexadecimal. Number base conversions- 1's, 2's, rth's complements, signed Binary numbers. Binary Arithmetic, Binary codes - Weighted BCD, Gray code, Excess 3 code, ASCII – conversion from one code to another.

#### **Unit 2 Boolean Algebra**

**7 (L) hrs**

Logic gates, NAND-NOR Implementation, Boolean postulates and laws, basic logic functions, standard forms of logic expressions – Sum of Products (SOP), Product of Sums (POS), Minterm, Maxterm. Minimization of Boolean expressions – using Boolean Algebra and Karnaugh map Minimization.

#### **Unit-3 Logic families**

**4 (L) hrs**

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

Brief overview of Transistor as a switch, Logic gate characteristics – propagation delay, speed, noise margin, fan-out and power dissipation. Standard TTL and static CMOS gates.

**Unit-4 Introduction to VHDL (VHSIC Hardware Description Language) 4 (L) hrs**

Behavioural – data flow algorithmic and structural description, lexical elements, data objects types, attributes, operators. VHDL coding examples.

**Part-B**

**Unit-5 Combinational logic 12 (L) hrs**

Arithmetic circuits, Magnitude Comparator, decoders, encoders, code converters, parity checker, multiplexers, de-multiplexers, and their use in logic synthesis. Implementation of combinational logic using MUX, Hazards in combinational circuits, combinational circuit design examples in VHDL and simulation.

**Unit-6 Sequential Logic Design 12 (L) hrs**

Latches and Flip Flops (SR, D, JK, T), Timing in sequential circuits, Shift register, Counters – synchronous, asynchronous, Modulus and Ring Counters. Sequential circuit design examples in VHDL and simulation.

**Laboratory Work**

To implement the above topics covered with Digital Electronics kits & IC's and VHDL examples for combinational and sequential circuits using XILINX ISE 9.1/9.2.

**Following is only the suggested list of Practical's. Instructor may frame additional Practical's relevant to the course contents.**

| Experiment No. | Experiment Title                                                                                       |
|----------------|--------------------------------------------------------------------------------------------------------|
| 1              | To verify the truth table of various logic gates using ICs' 7408, 7402, 7400, 7404, 7432, 7486.        |
| 2              | To implement basic gates using universal gates NAND and NOR.                                           |
| 3              | 4-Bit Binary-to-Gray and Gray-to-Binary Code Converter: Realization using XOR gates.                   |
| 4              | 4-Bit BCD-to-Excess-3: Realization using basic and universal gates.                                    |
| 5              | Multiplexer: Truth-table verification and realization of half adder and full adder using IC74153 chip. |
| 6              | Implement Arithmetic Circuits using basic VHDL simulation.                                             |
| 7              | Implement 4:1 MUX and 8:1 MUX using VHDL for a given problem.                                          |
| 8              | Implement 1:4 DEMUX and 1:8 DEMUX using VHDL for a given problem.                                      |
| 9              | Implement SR, JK, D and T Flip Flops using VHDL simulation.                                            |
| 10             | Realization of 4-bit counter asynchronous and synchronous counters using VHDL.                         |

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

**Text Books**

1. Mano M.M., Ciletti M.D., “Digital Design with an Introduction to the Verilog HDL”, 6th Edition, Pearson India, 2006.
2. Thomas L Floyd, “Digital Fundamentals”, 11<sup>th</sup> Edition, Pearson, 2018.

**Reference Books**

1. A. K. Maini, “Digital Electronics: Principles, Devices and Applications, Wiley, 2007.
2. Michael D. Ciletti, “Advanced Digital Design with the Verilog HDL”, 2nd edition, Pearson education, 2017.
3. Roth and Kinney, “Fundamentals of Logic Design”, 7th edition, Cengage learning, 2014.
4. Laboratory Manuals.

**Online Learning Materials**

1. <https://www.youtube.com/watch?v=BikBi6ipmyk&list=PLdlPA9pGVVtZF-VSy02wOiUqrPWptjuGt>  
Accessed on April 15, 2025
2. <https://www.youtube.com/watch?v=oNh6V91zdPY&list=PLbRMhDVUMnge4gDT0vBWjCb3Lz0HnYKkX>  
Accessed on April 15, 2025
3. <https://www.youtube.com/watch?v=CeD2L6KbtVM&list=PL803563859BF7ED8C>  
Accessed on April 15, 2025

**Supplementary SWAYAM Course**

| Sr. No. | Course Name                 | Instructor                  | Host Institute | URL                                                                                                                     |
|---------|-----------------------------|-----------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------|
| 1       | Digital System Design       | Prof. Neeraj Goel           | IIT Ropar      | <a href="https://onlinecourses.nptel.ac.in/noc21_ee39/preview">https://onlinecourses.nptel.ac.in/noc21_ee39/preview</a> |
| 2       | Digital Electronic Circuits | Prof. Goutam Saha           | IIT Kharagpur  | <a href="https://onlinecourses.nptel.ac.in/noc25_ee20/preview">https://onlinecourses.nptel.ac.in/noc25_ee20/preview</a> |
| 3       | Digital Circuits            | Prof. Santanu Chattopadhyay | IIT Kharagpur  | <a href="https://onlinecourses.nptel.ac.in/noc21_ee75/preview">https://onlinecourses.nptel.ac.in/noc21_ee75/preview</a> |

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

**Course Code: SMCS101**

**Course Title: Seminar and Technical Report Writing for Engineers**

|                                                       |                                             |                                                  |
|-------------------------------------------------------|---------------------------------------------|--------------------------------------------------|
| <b>Programme:</b><br>B.Tech.                          | <b>L: 0 T: 0 P: 2</b>                       | <b>Credits: 1</b>                                |
| <b>Semester: 3</b>                                    | <b>Theory/Practical: Practical</b>          | <b>Teaching Hours: 30(P) = 30 hrs</b>            |
| <b>Total Max. Marks: 50</b>                           | <b>Continuous Assessment (CA) Marks: 50</b> | <b>End Semester Examination (ESE) Marks: Nil</b> |
| <b>Duration of End Semester Examination (ESE): NA</b> |                                             |                                                  |
| <b>Course Type: Core Course</b>                       |                                             |                                                  |

**Prerequisites (if any):** Knowledge of Text based editors

**On completion of the course, the student will have the ability to:**

| <b>CO#</b> | <b>Course Outcomes</b>                                                      |
|------------|-----------------------------------------------------------------------------|
| 1          | Install BibTeX and PDFLaTeX on Windows and Linux operating systems.         |
| 2          | Utilize LaTeX tools for document preparation and compilation.               |
| 3          | Compare LaTeX and Markdown formats using Pandoc for documentation purposes. |
| 4          | Present technical content effectively through oral presentations.           |
| 5          | Use LaTeX effectively for technical document writing.                       |

### Contents

| <b>Experiment No.</b> | <b>Experiment Title</b>                                                                                                                                                                                                                                                                   |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1                     | Introduction to LaTeX and Its Background <ul style="list-style-type: none"> <li>History of LaTeX, its evolution, and use cases.</li> <li>Comparison with other document preparation tools (Word, Google Docs).</li> <li>Advantages of LaTeX in research and technical writing.</li> </ul> |
| 2                     | Introduction to Supporting Tools for LaTeX <ul style="list-style-type: none"> <li>TeXLive, TeXworks and Overleaf.</li> <li>Online vs. offline compilation of LaTeX documents.</li> </ul>                                                                                                  |
| 3                     | Installation <ul style="list-style-type: none"> <li>Installing BibTeX on Windows: step-by-step process.</li> <li>Installing PDFLaTeX on Linux: dependencies and configurations.</li> </ul>                                                                                                |
| 4                     | Develop a LaTeX script to create a simple document that consists of 2 sections [Section1, Section2], and a paragraph with dummy text in each section. And                                                                                                                                 |



**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

|      | also include header [title of document] and footer [institute name, page number] in the document.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |          |          |              |              |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|----------|--------------|--------------|-------|--|----------|----------|----------|---|------------|--------|----|----|----|---|------------|--------|----|----|----|---|------------|--------|----|----|----|
| 5    | Develop a LaTeX script to create a document that displays the sample Abstract/Summary.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |          |          |              |              |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
| 6    | Develop a LaTeX script to create the Certificate Page of the Report [Use suitable commands to leave the blank spaces for user entry].                                                                                                                                                                                                                                                                                                                                                                                                                                           |          |          |              |              |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
| 7    | <div>Develop a LaTeX script to create a document that contains the following table with proper labels:</div> <table><tr><th rowspan="2">S.No</th><th rowspan="2">URN</th><th rowspan="2">Student Name</th><th colspan="3">Marks</th></tr><tr><th>Subject1</th><th>Subject2</th><th>Subject3</th></tr><tr><td>1</td><td>4XX22XX001</td><td>Name 1</td><td>89</td><td>60</td><td>90</td></tr><tr><td>2</td><td>4XX22XX002</td><td>Name 2</td><td>78</td><td>45</td><td>98</td></tr><tr><td>3</td><td>4XX22XX003</td><td>Name 3</td><td>67</td><td>55</td><td>59</td></tr></table> | S.No     | URN      | Student Name | Marks        |       |  | Subject1 | Subject2 | Subject3 | 1 | 4XX22XX001 | Name 1 | 89 | 60 | 90 | 2 | 4XX22XX002 | Name 2 | 78 | 45 | 98 | 3 | 4XX22XX003 | Name 3 | 67 | 55 | 59 |
| S.No | URN                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |          |          |              | Student Name | Marks |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Subject1 | Subject2 | Subject3     |              |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
| 1    | 4XX22XX001                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Name 1   | 89       | 60           | 90           |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
| 2    | 4XX22XX002                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Name 2   | 78       | 45           | 98           |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
| 3    | 4XX22XX003                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Name 3   | 67       | 55           | 59           |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
| 8    | Develop a LaTeX script to include the side-by-side graphics/pictures/figures in the document by using the subgraph concept.                                                                                                                                                                                                                                                                                                                                                                                                                                                     |          |          |              |              |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
| 9    | Develop a LaTeX script to create a document that consists of two mathematical equations.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |          |          |              |              |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
| 10   | Develop a LaTeX script to demonstrate the presentation of Numbered theorems, definitions, corollaries, and lemmas in the document.                                                                                                                                                                                                                                                                                                                                                                                                                                              |          |          |              |              |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
| 11   | Develop a LaTeX script to create a document that consists of two paragraphs with a minimum of 10 citations in it and display the reference in the section.                                                                                                                                                                                                                                                                                                                                                                                                                      |          |          |              |              |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
| 12   | Develop a LaTeX script to create a simple report and article by using suitable commands and formats of Major/Minor project report.                                                                                                                                                                                                                                                                                                                                                                                                                                              |          |          |              |              |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |
| 13   | <div>Markdown and Pandoc</div> <ul style="list-style-type: none"><li>• Introduction to .md format and Markdown syntax.</li><li>• Use cases of Markdown in documentation and technical writing.</li><li>• Conversion between LaTeX and Markdown using Pandoc.</li></ul>                                                                                                                                                                                                                                                                                                          |          |          |              |              |       |  |          |          |          |   |            |        |    |    |    |   |            |        |    |    |    |   |            |        |    |    |    |

**Note:**

- Students will give 15 minutes presentations on topics related to emerging technologies to showcase their seminar presentation abilities.
- Students will also prepare a simple report based on topic of presentation using formats of Major/Minor project report.
- Evaluation of seminar will be based on Rubrics.

**Reference Books**

1. Donald Knuth E., "The TeXBook", Twentieth revision, Addison Wesley, 1991.
2. Leslie Lamport, "LaTeX: A document preparation system, User's guide and reference manual", 2nd edition, Addison Wesley, 1994.

**Guru Nanak Dev Engineering College, Ludhiana**  
**An Autonomous College under UGC Act 1956**  
**B.Tech. (CSE), Scheme - 2024**

3. Frank Mittelbach, with Ulrike Fischer “The LaTeX Companion, 3rd edition (TTCT series)”, 3rd edition, Addison-Wesley Professional, 2023.
4. Van Emden J., “Effective communication for Science and Technology”, Palgrave, 2001.
5. Van Emden J., “A Handbook of Writing for Engineers”, Macmillan, 1998.
6. Van Emden J. and Eastel J., “Technical Writing and Speaking, an Introduction”, McGraw-Hill, 1996.
7. Pfeiffer W.S., “Pocket Guide to Technical Writing”, Prentice Hall, 2000.
8. Eisenberg A., “Effective Technical Communication”, McGraw-Hill, 1992.

**Online Learning Materials**

1. <https://latex-tutorial.com/tutorials/> Accessed on April 07, 2025
2. <https://guides.nyu.edu/LaTeX/home> Accessed on April 07, 2025
3. <https://www.overleaf.com/learn> Accessed on April 07, 2025
4. <https://www.overleaf.com/gallery/tagged/homework> Accessed on April 07, 2025
5. <https://www.tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf> Accessed on April 07, 2025