

**Program:** Diploma  
**Semester:** Third  
**Course:** Data Structure  
**Course Code:** 9D.202

L	T	P	C
3	0	0	3

### Course Objective:

- Be familiar with basic techniques of algorithm analysis
- Be familiar with writing recursive methods
- Master the implementation of linked data structures such as linked lists and binary trees
- Be familiar with advanced data structures such as balanced search trees, hash tables, priority queues and the disjoint set union/find data structure
- Be familiar with several sub-quadratic sorting algorithms including quicksort, mergesort and heapsort
- Be familiar with some graph algorithms such as shortest path and minimum spanning tree

### Unit I:

**Basics of C:** C character set, tokens, constants, variables, keywords. Operators (arithmetic, Logical, assignment, relational, increment and decrement, conditional, special, operator precedence), data types, Formatted input, formatted output. If statement (if, if-else, else-if ladder, nested if-else), Switch case statement, looping: for, while, do-while statements. Introduction to Arrays, Declaration, initialization and working of one & two dimensional arrays. Functions: defining functions, function call (call by value, call by reference).

### Unit II:

Introduction to datastructure: Data Representation: Abstract data Types, Data Structures (Linear and Non-Linear), Atomic Type. Data Types: Primitive data type, Derived data type, Operations on data structures, Traversing, Inserting, Deleting, Searching and sorting.  
Principles of programming and Analysis of Algorithms: Algorithms, Different approaches for designing an algorithm, Complexity in terms of time and space, Big O Notation.  
Searching & Sorting: Sorting, An Introduction, Sorting Techniques: Bubble Sort, Selection Sort, and Insertion Sort. Searching: An Introduction, Linear search, Binary Search.

### Unit III:

Stacks: Introduction to Stacks: Stacks as an Abstract Data Type, Primitive operations of stacks. Representation of Stacks through Arrays, Application of Stacks, Arithmetic expression: Polish Notation  
Queues: Introduction, Queue as an Abstract Data Type, Representation of Queues, Operations on queue: Searching, Insertion, Deletion. Types of queues: Circular Queues, Priority Queue, De-queues, Application of Queues  
Linked List: Introduction: Terminologies Node, Address, Pointer, Information, Next, Null pointer, Empty list etc. Operations on list Searching, Insertion and Deletion. Types of lists: Linear list, Circular list, doublylist, Array, stacks, queues, implementation using list.

### Unit IV:

Trees: Introduction to Trees, Types of Trees, General tree, Binary tree, Binary search tree, Traversal—In order, Preorder and Post order, searching--Depth-first search and Breadth-first search

Graphs: Introduction, Terminology graph, node (vertices), arcs(edge), directed graph, in-degree, out-degree, adjacent, successor, predecessor, relation, weight, path, length, Sequential Representation of Graphs, Linked Representation of Graphs.

**Suggested Readings:**

1. E. Balagurusamy – Programming in ANSI C, 3rd Edn. , TMH, New Delhi ; 2004
2. Y. Kanetkar – Let us C, 4th Edition, BPB Publication , New Delhi; 2002
3. E. Horowitz and S. Sahani, “Fundamentals of Data Structures”, Galgotia Booksource Pvt. Ltd, 2003
4. R. S. Salaria, “Data Structure & Algorithms”, Khanna Book Publishing Co. (P) Ltd., 2002.
5. P. S. Deshpande and O.G. Kakde, “C & Data Structure”, Wiley Dreamtech, 1<sup>st</sup> Edition, 2003.
6. Y. Langsam et. al., “Data Structures using C and C++”, PHI, 1999.
7. Schaum’s outline series, “Data Structure”, TMH, 2002

**Program:** Diploma  
**Semester:** Third  
**Course:** Data Structure Lab  
**Course Code:** 3DP.202

L	T	P	C
0	0	2	1

---

### Program for C

1. Program in C to print "Hello".
2. Program in C to add 3 integer numbers.
3. Program in C to print your name 10 times.
4. Program in C to find factorial of the given number.
5. Program in C to find greatest of two numbers using conditional operator.
6. Program in C to check whether the given number is even or odd.

### Program for Data Structure

1. Create 3x3 matrix then display right diagonal element.
2. Create 4x4 matrix then display left diagonal element.
3. Take ten numbers input from keyboard then print in ascending order using bubble sort..
4. WAP to delete a number from a given location in an array.
5. WAP to create a linked list and perform insertions a) at beginning b) at end c) before a given node d) after a given node
6. WAP to create a linked list and perform deletions a) from beginning b) from end c) at a given node d) after a given node
7. WAP to create a circular linked list and perform insertion at the beginning of list.
8. WAP to create a circular linked list and perform insertion at the end of list.
9. WAP to create a circular linked list and perform deletion from the beginning of list.
10. WAP to create a circular linked list and perform deletion from the end of list.
11. WAP to perform Push, Pop and Peep operations on a stack.
12. WAP to implement a linear queue.
13. WAP to implement a linked queue.
14. WAP to implement a priority queue.
15. WAP to implement a Binary Search tree and perform the following:
  - a) Insert Element
  - b) Preorder Traversal

**Program:** Diploma  
**Semester:** Third  
**Course:** VB.Net  
**Course Code:** 3D.203

L	T	P	C
3	0	0	3

---

**Course Objective:** The student will use Visual Basic.Net to build Windows applications using structured and object-based programming techniques. Students will be exposed to the following concepts and/or skills at an *Introductory concepts level*:

- Analyze program requirements
- Design/develop programs with GUI interfaces
- Code programs and develop interface using Visual Basic .Net
- Perform tests, resolve defects and revise existing code

#### **Unit I:**

Introduction to .NET framework, Common Language Runtime, .NET Framework Class Library, Need of .NET, familiarization with visual studio .NET IDE, The editor, Solution Explorer, Properties window, output window and command window, Toolbox

#### **Unit II:**

Different .NET Languages, Variables and constant, operators, Conditional Statement and looping statement, String functions, Math functions, formatting data, goto statement

#### **Unit III:**

Array, Functions, Sub procedures, Recursion, OOP Concept, Advantages of OOP, Creating object instances in VB.Net, Type Name and Type Of, Building own classes, Constructor

#### **Unit IV:**

Windows forms: Anchor and dock properties, Tab order menu, MDI form, Context menus, Color dialog, Font dialog, File dialog boxes, Adding controls at runtime

#### **Suggested Readings:**

1. Programming VB.Net: A guide for experienced programmers, Gary Cornell and Jonathan Morrison, a press.
2. Microsoft Visual Basic .NET Step by Step, Michael Halvorson, Microsoft Press

**Program:** Diploma  
**Semester:** Third  
**Course:** VB.Net Lab  
**Course Code:** 3DP.203

L	T	P	C
0	0	2	1

---

### List of programs:

1. Accept a character from console and check the case of the character.
2. Accept any character from keyboard and display whether it is  
Vowel or not.
3. Write a program to implement a calculator.
4. Write a program to input a number and check whether the number is Armstrong or not.
5. Write a program to find prime number between range of start number and end number.
6. Write the program that display Fibonacci series up to nth term
7. WAP to print the multiplication table from 2 to 10
8. Design a digital watch using timer control.
9. Develop a VB.Net application to perform timer based quiz of 10 questions.
10. Program to find the greatest among two numbers.
11. Program to determine whether a person is eligible to vote or not.
12. Develop a program to illustrate picture box and image list box
13. Program to Illustrate color dialog, font dialog, and open file dialog
14. Develop a menu based VB.Net application to implement a text editor with cut, copy,  
paste, save and close operations.
15. Program to read and print the details of a student using class and objects
16. Program to implement the concept of MDI form
17. Program to find the sum of digits of a number. Use the concept of function for it.
18. WAP to sort the elements of an array.
19. Program to add controls during runtime
20. Program to implement the concept of constructor

**Program:** Diploma  
**Semester:** Third  
**Course:** Object Oriented Programming  
**Course Code:** 3D.205

L	T	P	C
0	0	2	1

---

### Course Objective:

- Understand the features of C++ supporting object oriented programming
- Understand the relative merits of C++ as an object oriented programming language
- Understand how to produce object-oriented software using C++
- Understand how to apply the major object-oriented concepts to implement object oriented programs in C++, encapsulation, inheritance and polymorphism
- Understand advanced features of C++ specifically stream I/O, templates and operator overloading

### Unit 1

**History & features:** It's need & requirement, procedure oriented programming versus object oriented programming, basic concepts object oriented programming, object oriented languages. **Beginning with C++:** Concepts & structure of C++ programming, concepts of structure.

**Objects & classes:** Specifying a class, Defining member functions, Access specifiers (public, private) Arrays within a class, Creating objects, memory allocation for objects, static data & member function, Arrays of objects, objects as function argument.

### Unit 2

**Constructors and Destructors.:** Concept of Constructor, Types of constructors (Default, Parameterized, copy,), Overloaded Constructors (Multiple Constructor), Constructor with default argument, Destructors. Function overloading, Operator overloading (overloading unary & binary operators), rules for overloading operators.

### Unit 3

**Inheritance:** Concepts of inheritance, Derived classes, Member declaration (Protected), Types of inheritance (Single, multilevel, multiple, hierarchical, Hybrid inheritance).

**Polymorphism:** Concepts of polymorphism, types of polymorphism, function Overloading &function overriding, Virtual function, Static & dynamic binding.

## Unit 4

**Pointers in C++ :** Concepts of pointer (Pointer declaration, pointer operator, address operator, pointer expressions, and pointer arithmetic), Pointers & functions (Call by value, call by reference, pointer to functions, passing function to another function), Pointers in arrays (Searching, insertion & deletion), Pointers & objects (Pointers to objects, this pointer, and pointer to derived classes).

### ***Suggested Readings:***

1. Object Oriented Programming in C++ by E. Balagruswamy
2. Object Oriented Programming in C++ Saurav Sahay Oxford University Press.
3. Object Oriented Programming in C++ R Rajaram New Age International Publishers 2nd Edition.
4. Object Oriented Programming in C++ by Robert Lafore Techmedia Publication.

**Program:** Diploma  
**Semester:** Third  
**Course:** Object Oriented Programming Lab  
**Course Code:** 3DP.205

L	T	P	C
0	0	2	1

---

### List of programs:

1. A C++ program to print the following by reading number of rows to be printed from the user.

```
      *
     * *
    * * *
   * * * *
  * * * * *
```

2. A C++ program to display “Hello, Welcome to C++ Programming”.
3. A C++ program to print details name, roll number in a single and two lines.
4. A C++ program to print your personal details name, surname(single character), total marks, gender(M/F), result(P/F) by taking input from the user.
5. A C++ program to convert centigrade into Fahrenheit. Formula:  $C=(F-32)/1.8$
6. A C++ program that prompts the user to enter two integer values in int variables val1, val2 and find largest, sum, difference, product and ratio of these values.
7. Develop a simple calculator using if-else if and switch-case.
8. A C++ program to find all the prime numbers between 1 and 100. Write a function to do this.
9. A C++ program that uses functions to swap two integers.
10. A C++ program to print the Fibonacci series 0 1 1 2 3 5 8 13 .... By getting number of number to be displayed is given as input. Eg. 5 is input value means it should print first 5 numbers 0 1 1 2 3
11. Create a Structure called employee with the following details as variables within it.
  1. Name of the employee
  2. Age
  3. Designation
  4. Salary

Make a C++ program to create array of objects for the structure to access these and print the name, age, designation and salary

12. A C++ program to find the number of vowels present in the given character array using pointer arithmetic.
13. A C++ program to find the sum of factorial of a given number using recursive function.
14. An inline function to obtain largest of three numbers.
15. A Bank gives 4% interest on current account and 6% interest on savings account. An additional 3% interest is provided for savings duration of 5 years and above. Using dynamic initialization of constructor write banking program using C++.

**Program:** Diploma  
**Semester:** Third  
**Course:** Digital Electronics  
**Course Code:** 6D.202

L	T	P	C
3	0	0	3

### Course Objective:

- To understand the fundamental concepts and techniques used in digital electronics
- To understand structure of various number systems and its application in digital design
- The ability to understand and analyze various combinational and sequential circuits
- Ability to identify basic requirements for a design application and propose a cost effective solution
- To develop skill to build, and troubleshoot digital circuit.

### Unit I:

**Introduction To Digital Techniques:** Digital circuit. Digital signal. Use of digital circuit and digital signal. Advantages and Disadvantages of Digital circuits. Generation of digital signal. Introduction to digital ICs, Characteristics of digital ICs. Logic families comparison of TTL, CMOS and ECL logic Families (No circuits) (To be covered in Practical). Number System-Introduction to Binary, Octal, Decimal, Hexadecimal number system. Conversion of number systems. 1's complement and 2's complement. Binary arithmetic (addition, subtraction). BCD code, BCD arithmetic (addition, subtraction).

### Unit II:

**Logic Gates And Boolean Algebra:** Logical symbol, logical expression and truth table of AND, OR, NOT, NAND, NOR, EX-OR and EX-NOR gates. Universal gates—NAND and NOR gates. Logical circuits of basic gates using universal gates. Gates using more than two inputs. TTL and CMOS logic gate ICs and the pin configurations. (To be covered in Practical). Basic laws of Boolean algebra, Duality theorem. De Morgan's theorems.

**Combinational Logic Design/ Circuits :** Simplification of Boolean expression using Boolean algebra. Construction of logical circuits forms Boolean expressions. Boolean expressions using Sum of products and product of sums forms. K-map representation of logical functions. Minimization of logical expressions using K-map (2,3,4 variables). Standardization of SOP & POS equations.

### Unit III:

Concept of Adders/ Subtractors: Truth table, K-map, Simplified logical expression and logical circuit using basic gates and universal gates of: (a) Half adder and full adder. (b) Half subtractor and full subtractor. Block diagram, Truth table, Logical expression and logic diagram of Multiplexers (4:1 and 8:1), Multiplexer IC. Block diagram and Truth table of De-multiplexer (1:4; 1:8; 1:16), Demultiplexer IC. Block diagram and Truth table of Encoders, Priority Encoders ICs and Decoder. Block diagram, Truth table, working principle, Applications, pin functions of Decimal to BCD Encoder (IC74147) and BCD to 7 segment Decoder. Block diagram and function table of Parity generator (IC74180), Digital comparator IC (7485); Block diagram and pin functions of ALU74181

### Unit IV:

**Memories :** Classification of memories, RAM, ROM, PROM, EPROM, E<sup>2</sup>PROM. Circuit diagram using CMOS transistors and working of Static and dynamic RAM

**A-D And D-A Converters :** Circuit diagram and working of R-2R ladder DAC and Weighted resistor DAC. DAC specifications. Block diagram and working of Ramp ADC, Dual slope ADC and Successive approximation ADC. ADC specification. Advantages and Disadvantages of various methods.

**Program:** Diploma  
**Semester:** Third  
**Course:** Digital Electronics Lab  
**Course Code:** 6DP.202

L	T	P	C
0	0	2	1

---

### List of Experiments

- 1) To illustrate the working of AND, OR & NOT gate.
- 2) To illustrate the De-Morgan's theorem with basic gates.
- 3) To illustrate the working of adders.
- 4) To illustrate the working of subtractors.
- 5) To illustrate the working of EX-OR gate & EX-NOR gate.
- 6) To illustrate the working of four to one multiplexer & verify the truth table.
- 7) To study IC7404, IC7432, & IC7408 and verify the NOT gate, OR gate & AND gate present in it.
- 8) Design a 4-bit R -2- R ladder D/A converter using OP-AMP. Determine its accuracy & resolution.

**Program:** Diploma  
**Semester:** Third  
**Course:** Applied Mathematics-III  
**Course Code:** 9D.204

L	T	P	C
3	0	0	3

### Course Objective:

- The subject helps the students to develop the fundamentals and basic concepts in vector calculus, ODE, Laplace transform and complex functions. Students will be able to solve problems related to engineering applications by using these techniques.
- Students will use mathematics concepts in real world situations.
- Students will simplify and perform operations with nonlinear expressions.
- Apply the principles of Vector algebra to solve a variety of basic problems in engineering and Applied Science.

### Unit I:

Complex Variable: Continuity, Differentiability, and analyticity of a function of a complex variable, Cauchy – Riemann equations in Cartesian and Polar form, Harmonic Function.

### Unit II:

Ordinary Differential Equation :Formation of Ordinary differential equation ,Variable Separable ,Homogeneous Equation ,Solution of Bernoulli’s Equation ,Exact-Differential Equation.

### Unit III:

Graph Theory and Laplace Transform: Graph, Sub graph, Walk travel and Path, Connected and Disconnected graph. Handshaking theorem or Edge and Vertex of a graph. Matrix representation of graph (Incidence and Adjacency Matrices). Spanning Tree. Kruskal’s Algorithms for minimal Spanning tree. Laplace Transform: Definition of Laplace Transform, Inverse Laplace Transform.

### Unit IV:

Numerical Solution of Algebraic Equation and Simultaneous Equation: Bisection Method, Regula –Falsi Method, Newton-Raphson Method, and Gauss elimination Method, Iterative Method –Gauss Siedal and Jacobi’s Method.

### Suggested Readings:

1. *Higher Engineering Mathematics –Dr.B.S.Grewal Khanna Publication*
2. *Higher Engineering Mathematics –H.K.Das*
3. *Engineering Mathematic -N.P.Bali Laxmi Publication*
4. *Advance Engineering Mathematics –Erwin Kreyszig Wiley Publication.*
5. *Graph Theory: Prabhakar Gupta and Vineet Agarwal Pragati Prakashan.*

**Program:** Diploma

**Semester:** Third

**Course:** Communication & Soft Skills

**Course Code:** 40.201

L	T	P	C
2	0	0	2

---

**COURSE OBJECTIVE** - The aim is to develop students' soft skills, communication, leadership and teamwork skills; and personal development skills using practical approach and exposure of students to the realities of the world.

- To enhance Leadership – assessing the requirements of a task, identifying the strengths/weaknesses within the team, utilising the diverse skills of the group to achieve the set *objectives*.
- To improve *Communication* – demonstrating clear briefing and listening /speaking skills.
- To make them realize that effective *communication* and interpersonal skills are crucial to increase employment opportunities and to compete successfully in the business environment.
- The *course* aims to cause a basic awareness about the significance of *soft skills* in professional and inter-personal *communications and* facilitate an all-round development of personality. Hard or technical skills help securing a basic position in one's life and career. But only *soft skills* can ensure a person retain it, climb.

## **UNIT-1 COMMUNICATION SKILLS**

- Introduction
- Role of Communication in Today's World
- Objective of Communication
- Process of Communication
- Elements of Communication
- Essentials of Communication
- Barriers/ Factors Inhibiting Communication
- Flow of Communication
- Verbal Mode of Communication

## **Unit -2 COMMUNICATION NETWORK**

- Non Verbal Mode of Communication
- Kinesics/Body Language, proxemics , chronemics, para lingual
- Style in Technical Communication
- Communication Skills; Reading, Writing, Speaking, Listening & Talking

### **Unit -3 GRAPHICS**

- Introduction
- Planning of Graphics
- Placing of Graphics
- Construction of Graphics
- Types of Graphics(textual ,visual, tables, bar Charts, pie charts, line charts, organizational charts, flow charts, maps & Pictographs)

### **Unit -4 TELEPHONIC CONVERSATION SKILLS**

- Introduction
- Stages in Telephonic Conversation
- Listening & Speaking Skills
- Telephonic Skills
- Problems in Telephonic Conversation
- Intensive Listening

### ***Suggested Books & Readings:***

- Monippally, Matthukutty. M. 2001. *Business Communication Strategies*. 11<sup>th</sup> Reprint. Tata McGraw-Hill. New Delhi
- Swets, Paul. W. 1983. *The Art of Talking So That People Will Listen: Getting Through to Family, Friends and Business Associates*. Prentice Hall Press. New York
- Lewis, Norman. 1991. *Word Power Made Easy*. Pocket Books
- Sen , Leena .Communication Skills ; Eastern Economy Edition
- Ghanekar , Dr. Anjali . Essentials of Business Communication Skills ; Everest Publishing House
- David Green . *Contemporary English Grammar, Structure & Composition* ; MacMillan
- Dictionary; Oxford
- Dictionary ; Longman

### ***Websites***

- [www.tatamcgrawhill.com/digital\\_solutions/monippally](http://www.tatamcgrawhill.com/digital_solutions/monippally)
- [www.dictionary.cambridge.org](http://www.dictionary.cambridge.org)
- [www.wordsmith.org](http://www.wordsmith.org)
- [www.edufind.com](http://www.edufind.com)
- [www.english\\_the\\_easy\\_eay.com](http://www.english_the_easy_eay.com)
- [www.englishclub.com](http://www.englishclub.com)
- [www.english\\_grammar\\_lessons.com](http://www.english_grammar_lessons.com)
- [www.wikipedia.org/wiki/english\\_grammar](http://www.wikipedia.org/wiki/english_grammar)