

GANPAT UNIVERSITY

FACULTY OF ENGINEERING & TECHNOLOGY

Programme	Bachelor of Technology	Branch/ Spec.	Computer Science & Engineering Computer Science & Engineering (Big Data and Analytics) Computer Science & Engineering (Cyber Security) Computer Science & Engineering (Artificial Intelligence and Machine Learning)
Semester	I	Version	1.0.0.0
Course Code	2ESC1114	Course Name	ESSENTIAL OF SOFTWARE FOUNDATION & PROGRAMMING – I
Effective from Academic Year	2026-27	Effective for the batch Admitted in	July 2026
Course Category	ESC		

Teaching, Learning, and Examination Scheme

Teaching & Learning Scheme							Examination scheme (Marks)					
Hours (per week)				Cr	SL	TSL	Theory			Practical		
L	TU	P	TTH				CE	SEE	Total	CE	SEE	Total
3	0	4	7	5	45	150	50	50	100	50	50	100

L: Lecture, TU: Tutorial, P: Practical, TTH: Total Teaching Hours/Week, Cr: Credit, SL: Min. Self Learning/Term Work Hours/Sem, TSL: Min. Total Teaching & Learning Hours/Sem, CE: Continuous Evaluation, SEE: Semester End Examination,

Pre-requisites

Basic Operational knowledge of Computer Fundamentals

Course Outcomes

COs	After completing the Course, students will be able to
CO1	Understand the fundamentals of programming through flowcharts & logic building steps.
CO2	Analyze, write, debug, and test a well-structured, robust computer program using C Language.
CO3	Apply conditions, loops, basic data types, functions, file handling, and structure to various problems.
CO4	Develop the project and enhance the skills of code versioning, testing, and documentation.

Theory Syllabus

Unit	Content	Hrs
1	Programming Basics & Logical Building Pseudocode, Flowcharts elements, developing algorithms, text editors, translators, debuggers, and indentation. Variables, data types, Operators and I/O functions, Basic structure of c program, Character Set, C tokens, Keywords and Identifiers, Constants, Variables, Data types, typedef and Defining symbolic constants, printf - scanf function, Operators Introduction, reading and writing a character, formatted input and output, Different types of Operator, Evaluation of expressions, Precedence of arithmetic operators, Type conversions in expressions, Operator precedence and associativity.	8
2	Decision Making and Looping Branching, Various decision making statements, and the ternary Operator, entry-controlled and exit-controlled loops, Break and continue, GOTO statement.	4
3	Array: One-dimensional arrays, Two-dimensional arrays, Concept of Multidimensional arrays. String handling: Declaring and Initializing string variables, Reading and Writing string from terminal, Various string functions, array of strings.	6
4	User-Defined Functions Need for user-defined functions, The form of C function, elements of UDF, Return values and their types, category of functions, Nesting of functions, Recursion , Functions with arrays, The scope of variables in functions.	6
5	Pointers Understanding pointers, Declaring and initializing pointers, Accessing a variable through its pointer, Pointer expressions, Pointer increments and scale factor, Pointers and arrays, Pointers & character strings, Pointers & Functions Dynamic memory allocation & revisiting structures in C malloc(), calloc(), realloc(), Free(), Structure overview, Structure limitations, need of object oriented	8

	programming	
6	File Management & Command Line Arguments File structure, File attributes, Standard i/o, Formatted i/o, File Operations, File Error Handling, Fundamentals of Command Line Arguments (argc, argv)	6
7	Structures and Unions Declaring and initializing Structure variables, accessing Structure members, Copying & Comparison of structures, Arrays of structures, Arrays within structures, Structures within Structures, Structures and functions, Unions, Pointers and structures.	7

Practical Content

Practicals are based on topics like basic data Types, Operators, decision making, looping, arrays, Strings, functions, Pointers and File management in C, Structure & Unions.

Note: All the practicals should be carried out on Linux Operating System and use code versioning using git extensively.

Text Books

1 Let Us C : Yashavant Kanetkar

Reference Books

1 Programming in C By E. Balagurusamy.

2 The C Programming Language By Kernighan and Ritchie.

ICT/MOOCs Reference

1 Course Name: Introduction to Programming in C
[Link: https://onlinecourses.nptel.ac.in/noc19_cs42/preview](https://onlinecourses.nptel.ac.in/noc19_cs42/preview)

2 Spoken Tutorial STP:
[Link: https://spoken-tutorial.org/tutorial-search/?search_foss=C+and+Cpp&search_language=English](https://spoken-tutorial.org/tutorial-search/?search_foss=C+and+Cpp&search_language=English)

Mapping of Cos, POs, and PSOs

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2	PSO 3
CO1	3	3	2	2	3	1	0	3	2	3	3	3	2	1
CO2	3	3	2	1	1	1	0	1	0	1	2	3	3	1
CO3	3	3	3	2	1	1	0	2	1	1	3	3	3	2
CO4	3	2	3	3	2	1	2	3	3	3	3	3	3	3

Unit	Unit Title	Aligned COs	Learning Hours	BTL Level
1	Programming Basics & Logical Building	1	8	R,U,A
2	Decision Making and Looping	1,2,3	4	R,U,A
3	Array & String handling:	2,3	6	R,U,A
4	User-Defined Functions	2,3	6	U,A,E
5	Pointers & Dynamic memory allocation & revisiting structures in C	3,4	8	A,N,E
6	File Management & Command Line Arguments	2,3,4	6	A,N,E,C
7	Structures and Unions	2,4	7	U,A,N,E

Note:

- Version 1.0.0.0 (First Digit= New syllabus/Revision in Full Syllabus, Second Digit=Revision in Teaching Scheme, Third Digit=Revision in Exam Scheme, Forth Digit= Content Revision)
- L=Lecture, TU=Tutorial, P= Practical/Lab., TW= Term work, DT= Direct Teaching, Lab.= Laboratory work
CE= Continuous Evaluation, SEE= Semester End Examination
- 1 Hour Lecture = 1 Credit, 1 Hour Tutorial = 1 Credit, 2 Hours Practical = 1 Credit, 2 Hours Internship/Project/Seminar = 1 Credit
- As per NCrF/NEP 2020, Minimum Self-Learning or Term Work Hours per Semester should be calculated in such a way that 1 Credit should have minimum 30 Hours of Teaching and Self Learning Engagement per semester