Mathematics-I					
Prerequisite: None		L	T	P	C
Total hours: 28		2	1	0	3
Course Content					Hrs.
Unit 1	Differential Calculus: Review of calculus of one variable and ε-δ definition, functions of several variables, Limit, Continuity, Partial derivatives, Euler's theorem on homogeneous functions, Chain rule, Jacobians, Taylor's theorem for two variables (Statement only), Error approximations.				10
Unit 2	Integral Calculus: Double and Triple integrals, Change of order of integration. Change of variables. Improper Integrals(Gamma and Beta functions).				8
Unit 3	Vector Calculus: Differentiation of vectors, gradient, directional derivative, divergence, curl and their physical meaning. Identities involving gradient, divergence and curl, Line and surface integrals, Green's, Gauss and Strokes' theorem (Statement only) and their applications				10
Text Books					
1.	Kreyszig E., Advanced Engineering Mathematics, 10th Edition, Wiley, 2011.				
2.	Hass J., Heil C. and Weir M. D., Thomas' Calculus, 14th Edition, Pearson Education, 2018.				
Reference books					
1.	Jain R. K. and Iyengar S. R. K., Advanced Engineering Mathematics, 5th Edition, Narosa Publishing House, 2016.				
2.	Zill D. G. and Wright W. S., Advanced Engineering Mathematics, 4th Edition, Viva, 2011.				
3.	O'Neil P. V., Advanced Engineering Mathematics, 8th Edition, Cengage Learning, 2017.				
4.	Grewal B.S., Higher Engineering Mathematics, 44th Edition, Khanna publishers, 2021.				

Discrete Mathematics for Computing					
Prerequisite: None		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Logic: Syllogism in Nyay (Logic in India). Well-formed formula, interpretations, propositional logic, predicate logic, theory of inference for propositional logic and Proof techniques.				9
Unit 2	Set Theory: Review of Basic Set Operations, cardinality of a set, relations, equivalence relations, partially ordered sets, functions, countability. Halayudha commentary. Paravartya sutra method for division of polynomials, lattices and Boolean algebras.				9
Unit 3	Combinatorics: Permutations, combinations, Pigeon Hole principle, inclusion exclusion principle, recurrences, generating functions, partitions, Virahanka Fibonacci sequence, Recurrence relations, special numbers.				12
Unit 4	Group Theory: Modular Arithmetic, Kuttaka problem. Groups-Binary operation, and its properties, Definition of a group, Groups as symmetries, cyclic, dihedral, symmetric, matrix groups, Subgroups, Cosets, normal subgroups and quotient groups, Conjugacy classes, Lagrange's theorem, Historical evolution of algebra and symmetry: Bhaskara II, development of group theory, and applications in modern computing, Monoids. Introduction to rings and fields.				12
Course outcomes: • Apply the concepts of sets, relations, functions, logic, and proof techniques to solve discrete mathematical problems. • Use combinatorial methods and recurrence relations to analyze and solve computational problems. • Analyze algebraic structures such as groups, monoids, rings, and fields and their applications in computing. • Appreciate the historical development of discrete mathematics, including significant Indian contributions, and their relevance to modern computing.					
Text Books					
1.	Herstein I. N., Abstract Algebra. 3rd ed. John Wiley & Sons, 1996.				
2.	Tremblay J. P., and Manohar R., Discrete Mathematics with Applications to Computer Science. Tata McGraw-Hill, 1997.				

3.	Raju C. K., Cultural Foundations of Mathematics: The Nature of Mathematical Proof and the Transmission of the Calculus from India to Europe in the 16th c. CE. Pearson Longman, 2007.
Reference books	
1.	Huth M., and Ryan M., Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press, 2004.
2.	Gallian J. A., Contemporary Abstract Algebra. 9th ed. Cengage Learning, Boston, 2017.
3.	Liu C. L., and Mohapatra D.P., Elements of Discrete Mathematics. 4th ed. Tata McGraw-Hill Education, 2012.
4.	Rosen K. H., Discrete Mathematics and Its Applications. 6th ed. Tata McGraw-Hill, 2007.
5.	Joseph G. G., The Crest of the Peacock: Non-European Roots of Mathematics. 3rd ed. Princeton University Press, 2011.

<table>
<tr><th colspan="6">Problem Solving with C</th></tr>
<tr><td colspan="3">Prerequisite: None</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="3">Total hours: 28</td><td>2</td><td>0</td><td>0</td><td>2</td></tr>
<tr><td colspan="5">Course Content</td><td>Hrs.</td></tr>
<tr><td>Unit 1</td><td colspan="4">Introduction: Problem-solving strategies, algorithms and flowcharts, introduction to C, structure of a C program, variables, constants, data types, operators, input/output statements, expressions and type conversion.</td><td>6</td></tr>
<tr><td>Unit 2</td><td colspan="4">Decision Making and Iterative Constructs: Conditional statements, switch statement, loops (for, while, do–while), nested loops, break and continue statements, programming applications involving numerical and logical problems.</td><td>6</td></tr>
<tr><td>Unit 3</td><td colspan="4">Functions and Arrays: User-defined and library functions, parameter passing, recursion, one-dimensional and two-dimensional arrays, searching and sorting algorithms, string handling using character arrays, introduction to modular programming.</td><td>7</td></tr>
<tr><td>Unit 4</td><td colspan="4">Pointers and Structures: Pointers and pointer arithmetic, pointers with arrays and functions, structures and unions.</td><td>6</td></tr>
<tr><td>Unit 5</td><td colspan="4">File Handling: File operations, dynamic memory allocation.</td><td>3</td></tr>
<tr><td colspan="6">Course Outcomes:
Ability to develop algorithms and flowcharts to solve computational problems.
Write, compile, debug, and execute C programs using fundamental programming constructs.
Design modular programs using functions, arrays, pointers, and structures.
Apply file handling and dynamic memory allocation techniques to solve real-world programming problems.
Analyze and implement efficient solutions for mathematical and engineering applications with C.</td></tr>
<tr><td colspan="6">Text Books</td></tr>
<tr><td>1.</td><td colspan="5">J.R. Hanly and E.B. Koffman, Problem Solving and Program Design in C, Pearson Education, 2016.</td></tr>
</table>

2.	B.W. Kernighan and D.M. Ritchie, *The C Programming Language*, Pearson Education, 1988.
Reference books	
1.	Dromey R.G., How to solve it by computer, Prentice-Hall International, 1982.
2.	B.S. Gottfried, *Programming with C*, McGraw-Hill Education, 2018.
3.	E. Balagurusamy, *Programming in ANSI C*, McGraw Hill Education, 2019.
4.	P. Deitel and H. Deitel, *C How to Program*, Pearson Education, 2016.

Problem Solving with C Lab				
Prerequisite: Basic Programming Skills	L	T	P	C
Total hours: 24	0	0	2	1

Course Content

Implementation of programs involving:

- Algorithms and flowcharts
- Basic input/output and arithmetic computations
- Decision making and iterative programming
- Number-theoretic and mathematical problems
- Functions and recursive algorithms
- Array manipulation, searching and sorting
- String processing
- Pointers and structures
- File handling
- Mini problem-solving assignments integrating multiple programming concepts.

Course Outcomes:

- Ability to develop algorithms and flowcharts to solve computational problems.
- Write, compile, debug, and execute C programs using fundamental programming constructs.
- Design modular programs using functions, arrays, pointers, and structures.
- Apply file handling and dynamic memory allocation techniques to solve real-world programming problems.
- Analyze and implement efficient solutions for mathematical and engineering applications with C.

Text Books

1.	J.R. Hanly and E.B. Koffman, *Problem Solving and Program Design in C*, Pearson Education, 2016.
2.	B.W. Kernighan and D.M. Ritchie**,** *The C Programming Language*, Pearson Education, 1988.

Reference books

1.	Dromey R.G., How to solve it by computer, Prentice-Hall International, 1982.
2.	B.S. Gottfried, *Programming with C*, McGraw-Hill Education, 2018.
3.	E. Balagurusamy**,** *Programming in ANSI C*, McGraw Hill Education, 2019.
4.	P. Deitel and H. Deitel, *C How to Program*, Pearson Education, 2016.