Programme	B. Sc. Computer Science				
Course Title	Foundations of C Programming				
Type of Course	**Minor**				
Semester	II				
Academic Level	100-199				
Course Details	Credit	Lecture per week	Tutorial per week	Practical per week	Total Hours
	4	3	-	2	75
Pre-requisites	1. Basic Computer Literacy 2. Basic Problem-Solving Skills				
Course Summary	This course teaches the basics of programming using the C language. C is a powerful and widely used programming language known for its efficiency and flexibility. Through this course, students will learn how to write, understand, and debug C code to solve various problems and build simple applications.				

Course Outcomes (CO):

CO	CO Statement	Cognitive Level*	Knowledge Category#	Evaluation Tools used
CO1	Demonstrate a solid understanding of fundamental programming concepts	An	P	Instructor-created lab exams / Quiz
CO2	Develop effective problem-solving skills by applying algorithmic thinking and logical reasoning.	An	P	Problem-solving assessments
CO3	Gain proficiency in writing, compiling, debugging, and executing C programs to implement algorithms, solve	Ap	P	Modeling Assignments

	problems, and create applications.			
CO4	Learn techniques to write efficient and optimized C code, including memory management, algorithm design, and performance tuning, to produce high-quality and scalable software solutions.	Ap	P	Modeling Assignments/ / Case studies
CO5	Understand and apply software development practices such as modular programming, code documentation, and debugging techniques to write maintainable and robust C programs.	Ap	P	Modeling Assignments/ / Case studies

* - Remember (R), Understand (U), Apply (Ap), Analyse (An), Evaluate (E), Create (C)

- Factual Knowledge(F) Conceptual Knowledge (C) Procedural Knowledge (P) Metacognitive Knowledge (M)

Detailed Syllabus:

Module	Unit	Content	Hrs	Marks
I	**Problem-solving and logical Thinking**		**10**	**15**
	1	Overview of computational thinking concepts. Definition of algorithm and its characteristics.Understanding the importance of algorithms in problem-solving	2	
	2	Algorithm Development: Steps involved in designing algorithms	2	
	3	Pseudocode as an intermediate step in algorithm development.	1	
	4	Understanding flowchart symbols and their meanings.Learning to represent algorithms using flowcharts.	2	
	5	Raptor as a precursor to text-based programming languages	2	
	6	Drawing simple flowcharts	1	
II	**Introduction to C**		**10**	**20**

	7	Structure of C program	2	
	8	C Character Set, Keywords, Identifiers	1	
	9	Data Types, Variables, Declarations, Symbolic Constants	2	
	10	Operators:Arithmetic, Logical, Relational & Equality, and Unary, Operator Precedence and Associativity	2	
	11	Library Functions, Comments	1	
	12	I/O functions- Formatted scanf() & printf().	2	
III	**Control Statements, Arrays & Strings**		**14**	**20**
	13	Selection Statements:if, if-else, switch	3	
	14	iteration: while, do while, for	4	
	15	Arrays: One dimensional and Two Dimensional(introduction only)	3	
	16	Strings: Basic string handling functions	2	
	17	Structure:Definition, Processing-period Operator, Union(Concepts only)	2	
1V	**User-defined Functions**		**11**	**15**
	18	Definition of function, Advantages, Understanding function prototypes, and declarations	3	
	19	Introduction to function definitions and function calls	3	
	20	Exploring function parameters: actual and formal parameters	2	
	21	Recursion	2	
	22	Pointers-declarations(Basic concept only)	1	
V	**Hands-on C:** **Practical Applications, Case Study and Course Project**		**30**	

	1	Write a C program using Variables and Data Types Write a C program using Arithmetic Operations Write a C program using Loops Write a C program using Arrays Write a C program using Functions Write a C program using Strings	20	
	2	Case study: 1. Library Management System: Develop a program to manage a library's collection of books. Implement functions for adding, removing, and searching for books. 2. Ticket Booking System: Design a program to manage ticket bookings for a cinema or theater.	5	
	3	Capstone/Course Project: Design a real-time project in C	5	

Reference:

1. Balagurusamy, E. Programming in ANSI C. Tata McGraw-Hill Education, 2019.
2. King, K. N. C Programming: A Modern Approach. 2nd ed., W. W. Norton & Company, 2008.
3. Kernighan, Brian W., and Dennis M. Ritchie. The C Programming Language. 2nd ed., Prentice Hall, 1988.
4. Prata, Stephen. C Primer Plus. 6th ed., Addison-Wesley, 2013.
5. Perry, Greg. Absolute Beginner's Guide to C. 3rd ed., Que Publishing, 2014.
6. Oualline, Steve. Practical C Programming. 3rd ed., O'Reilly Media, 1997.
7. Hanly, Jeri R., and Elliot B. Koffman. Problem-Solving and Program Design in C. 8th ed., Pearson, 2016.
8. Gottfried, Byron S. Programming with C. 2nd ed., McGraw-Hill, 1996.
9. Holmes, Dan. C in a Nutshell. 2nd ed., O'Reilly Media, 2015.

Mapping of COs with PSOs and POs :

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	-	-	-	-	-	-	3	0	3	2	3	0	1
CO2	-	-	-	-	-	-	3	0	3	2	3	0	1
CO3	-	-	-	-	-	-	3	0	3	2	3	0	1
CO4	-	-	-	-	-	-	3	0	3	2	2	0	1
CO5	-	-	-	-	-	-	3	0	3	3	2	0	1

Correlation Levels:

Level	Correlation
-	Nil
1	Slightly / Low
2	Moderate / Medium
3	Substantial / High

Assessment Rubrics:

- Quiz / Assignment/ Quiz/ Discussion / Seminar
- Midterm Exam
- Programming Assignments (20%)
- Final Exam (70%)

Mapping of COs to Assessment Rubrics :

	Internal Exam	Assignment	Project Evaluation	End Semester Examinations
CO 1		✓		✓
CO 2	✓	✓		✓
CO 3		✓		✓
CO 4	✓			✓
CO 5	✓		✓	✓