Sl. no	Category of Courses	Continuous Assessment	Mid – Semester Assessment	End Semester
1.	Theory	40	20	40

CS6101 PROGRAMMING WITH C	L	T	P	EL	CREDITS
	2	1	4	3	6

Prerequisites for the course: None

OBJECTIVES:

- To learn programming using a structured programming language.
- To implement programs using basic features of C.
- To learn to use C pointers and dynamically allocated memory techniques.
- To learn advanced features of the C programming language
- To be able to use file operations in C

MODULE I :	L	T	P	EL
	2	1	4	3

Notion of memory, addresses, variables, instructions, execution of instructions. Operating system commands, file editing, compiling, linking, executing a program.

SUGGESTED ACTIVITIES :

- Practical - Use of operating system commands and file editing operations.

SUGGESTED EVALUATION METHODS:

- Exercises on the use of operating system commands and file editing operations.

MODULE II :	L	T	P	EL
	2	1	4	3

Data types - constants, variables - arithmetic operators - expressions - basic input/output. Relational, logical, increment, decrement operators. Bitwise, assignment, conditional operators.

SUGGESTED ACTIVITIES :

- EL - Programs using integer type, arithmetic operators and basic input/output.
- EL - Programs using other data types and operators.
- Practical - Demonstration of programs using integer type, arithmetic operators and basic input/output.
- Practical - Demonstration of programs using other data types and operators.

SUGGESTED EVALUATION METHODS:

- Programs on integer type, arithmetic operators, basic input output.

MODULE III :	L	T	P	EL
	2	1	4	3

Statements and blocks - Selection - if-else construct - iteration - while - for constructs. The constructs else-if, switch, do-while, break, continue, enum. Pseudocode, Programming style.

SUGGESTED ACTIVITIES :

- EL: Programs using if-else, while, for.
- EL: Programs using else-if, switch, do-while, break, continue, enum. Use of pseudocode, programming style.
- Practical: Demonstration of programs using if else, while, for.
- Practical: Use of pseudocode. Demonstration of programs using else-if, switch, do-while, break, continue, enum, programming style.

SUGGESTED EVALUATION METHODS:

- Programs using if else, while, for.

MODULE IV :	**L**	**T**	**P**	**EL**
	4	**2**	**8**	**6**

Array, declaration, initialization. Multi dimensional arrays. Strings and character arrays, string operations on arrays.

SUGGESTED ACTIVITIES :

- EL - Programs using arrays and operations on arrays.
- Practical - Demonstration of programs using arrays and operations on arrays.
- EL - Programs implementing string operations on arrays.
- Practical - Demonstration of programs implementing string operations on arrays.

SUGGESTED EVALUATION METHODS:

- Evaluation: Programs using arrays and operations on arrays.
- Evaluation: Programs using strings and use of string library functions.
- Evaluation: Programs implementing string operations on arrays.

MODULE V :	**L**	**T**	**P**	**EL**
	4	**2**	**8**	**6**

Functions, definition, call, arguments, call by value. Call by reference. Recursion. Call stack. Header files, static variables, external variables.

SUGGESTED ACTIVITIES :

- EL - Programs using functions.
- Practical - Demonstration of programs using functions.
- EL - Programs using recursion.
- Practical - Demonstration of programs using recursion.

SUGGESTED EVALUATION METHODS:

- Evaluation: Programs using functions.
- Evaluation: Programs using recursion.

MODULE VI:	**L**	**T**	**P**	**EL**
	6	**3**	**12**	**9**

Pointers and arrays - address arithmetic. Dynamic Memory Allocation - Two dimensional arrays and pointers. Pointers and strings, string library functions. Pointers to functions.

SUGGESTED ACTIVITIES :

- EL - Programs using pointers and arrays, address arithmetic.
- Practical - Demonstration of programs using pointers and arrays, address arithmetic..

- EL - Programs using Dynamic Memory Allocation, two dimensional arrays and pointers.
- Practical - Demonstration of programs using Dynamic Memory Allocation, two dimensional arrays and pointers.
- EL - Programs using Pointers and strings..
- Practical - Demonstration of programs using pointers and strings.

SUGGESTED EVALUATION METHODS:

- Evaluation: Programs on pointers and arrays, address arithmetic..
- Evaluation: Programs using Dynamic Memory Allocation, two dimensional arrays and pointers.
- Evaluation: Programs using pointers and strings.

MODULE VII:	**L**	**T**	**P**	**EL**
	4	**2**	**8**	**6**

Structures, Structures and arrays. Pointers to structures, Self referential structures. Enumeration types, Unions, bit fields, typedefs.

SUGGESTED ACTIVITIES :

- EL - Programs using structures and arrays.
- Practical - Demonstration of programs using Structures and arrays.
- EL - Programs using Pointers to structures, Self referential structures.
- Practical - Demonstration of programs using pointers to structures, Self referential structures.

SUGGESTED EVALUATION METHODS:

- Evaluation: Programs using Structures and arrays.
- Evaluation: Programs using pointers to structures, self referential structures.

MODULE VIII:	**L**	**T**	**P**	**EL**
	2	**1**	**4**	**3**

Files - binary, text - open, read, write, random access, close. Preprocessor directives. Command line arguments.

SUGGESTED ACTIVITIES :

- EL - Programs using file operations in real-world applications.
- Practical - Demonstration of real-world application using file operations.

SUGGESTED EVALUATION METHODS:

- Evaluation: Demonstration of real-world application.

TEXT BOOKS:

1. Reema Thareja, “Programming in C”, 2nd ed., Oxford University Press, 2016.
2. Brian W. Kernighan and Dennis M. Ritchie, “The C Programming Language”, Pearson Education, 1988.
3. Brian W. Kernighan and Rob Pike, “The Practice of Programming” (Chap 1), Pearson Education, 1999.

REFERENCES:

1. Pradip Dey and Manas Ghosh, “Computer Fundamentals and Programming in C”, 2nd ed., Oxford University Press, 2013.
2. Yashavant Kanetkar, “Let us C”, 15th ed., BPB Publications, 2017.

3. Paul J. Deitel and Harvey Deitel, “C How to Program”, 7th ed., Pearson Education, 2013.

OUTCOMES:

Upon completion of the course, the students will be able to:

- Apply appropriate programming constructs to solve problems.
- Write C programs for simple applications.
- Use C pointers and dynamically allocated memory to solve complex problems.
- Know advanced features of the C programming language.
- Apply file operations to develop solutions for real-world problems.

EVALUATION METHOD TO BE USED:

Continuous assessment	Mid term	End Semester
40 (P)	20	40

	L	T	P	EL	CREDITS
CS6102 COMPUTATIONAL THINKING	**0**	**0**	**4**	**3**	**3**

Prerequisites for the course: None

OBJECTIVES:

- To formulate problems in a way that enables the use of a computer to solve them.
- To logically organize and analyze data.
- To automate solutions through algorithmic thinking.
- To identify, analyze and implement possible solutions with the goal of achieving the most efficient and effective combination of steps and resources.
- To generalize and transfer this problem solving process to wide variety of problems.

MODULE I :	L	T	P	EL
	0	0	4	3

Algorithmic thinking - creating oral algorithms for everyday tasks - Data abstraction and representation - Abstraction and translation of everyday data for use on a computer.

SUGGESTED ACTIVITIES :

- Explore algorithm design by creating oral algorithms.
- Abstract the essential details of everyday objects.
- Translate the description of everyday objects into data types and variables.

SUGGESTED EVALUATION METHODS:

- Evaluation of the oral algorithms and computer data.

MODULE II :	L	T	P	EL
	0	0	12	9

Decomposing a complex problem - Strategies for decomposition and algorithm design - Divide and conquer - Simple program implementations.