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.

SUGGESTED ACTIVITIES :

- Decompose a complex problem into discrete steps,
- Design a simple algorithm for solving the problem.
- External learning: Study of different strategies for decomposition and algorithm design.
- Examine sample input and expected output and develop strategies to decompose the problem.
- Use decomposition to break the problem into smaller problems and algorithmic design to plan a solution strategy.
- External learning: Simple program implementations.

SUGGESTED EVALUATION METHODS:

- Whiteboard presentations of the decomposition and algorithm.
- Evaluation of the developed strategies.
- Demonstration of the implemented programs.

MODULE III :	**L**	**T**	**P**	**EL**
	0	**0**	**8**	**6**

Overall data representation, abstraction, analysis and algorithm design. Program implementations.

SUGGESTED ACTIVITIES :

- Examples of Data representation, abstraction, analysis and algorithm design.
- Programming implementation.

SUGGESTED EVALUATION METHODS:

- Whiteboard presentations of the Data analysis and Algorithm design.
- Demonstration of the programming implementations.

MODULE IV:	**L**	**T**	**P**	**EL**
	0	**0**	**8**	**6**

Measuring the complexity of an algorithm - sorting algorithms - the notion of unsolvable problems. Programming illustrations.

SUGGESTED ACTIVITIES :

- Develop algorithms for sorting and determine the complexity of the algorithm and how it scales as the number of items to sort increases.
- Implement the different algorithms and measure how they scale.
- Determine which algorithms are more efficient, whether or not all algorithms are calculable given enough time.

SUGGESTED EVALUATION METHODS:

- Determine complexity of algorithms and how they scale with number of items.
- Demonstration using appropriate programs.
- Determine which algorithms are computable given enough time.

MODULE V:	L	T	P	EL
	0	0	4	3
Enhancing the clarity of a program - documentation, style, idioms.				
SUGGESTED ACTIVITIES : • External Learning: Study the best practices of documentation, style, idioms, etc that are used to ensure the code can be understood and maintained over a long period. • Use these practices in the documentation of earlier programs.				
SUGGESTED EVALUATION METHODS: • Documentation of given programs.				

MODULE VI:	L	T	P	EL
	0	0	9	9
Application of computational thinking to simple real world problems - program implementation of decomposed modules.				
SUGGESTED ACTIVITIES : • Application to simple real world problems.				
SUGGESTED EVALUATION METHODS: • Evaluation of the solutions to the real world problems				

REFERENCES:

1. Exploring Computational Thinking.
https://edu.google.com/resources/programs/exploring-computational-thinking/

OUTCOMES:

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

- Abstract out details of data and represent them appropriately.
- Create appropriate algorithms to solve specified problems.
- Confidently deal with complexity and open-ended problems.
- Apply the computational thinking skills to real world problems.
- Use best practices for documentation that can ensure long term maintenance.

EVALUATION METHOD TO BE USED:

Continuous assessment	Mid term	End Semester
60	40	-