20IT5352 Advanced Programming Lab - II

Course Category:	Program Core Lab	Credits:	2
Course Type:	Laboratory	Lecture-Tutorial-Practice:	0-0-2
Prerequisites:	20ES1152: Programming for problem solving using C Laboratory 20ES2152: Object Oriented programming using Python Laboratory 20IT3352: Data Structures Lab 20IT3353: Object Oriented Programming using C++ Lab	Continuous Evaluation:	30
		Semester end Evaluation:	70
		Total Marks:	100

Course Outcomes	Upon successful completion of the course, the student will be able to:	
	CO1	Combine fundamental data structures and algorithmic techniques in building a complete solution to a given problem
	CO2	Solve recurrences describing the performance of string algorithms.
	CO3	Develop combinatory solutions to the real-world problems
	CO4	Analyze dynamic programming strategies to solve a given problem.
	CO5	Derive solutions to the problems based on Computational Geometry
	CO6	Evaluate new techniques for solving specific problems inline with space and time requirements.

Contribution of Course Outcomes towards achievement of Program Outcomes (L-Low, M-Medium, H- High)		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
	CO1	2	2	3			2						3	2	3
	CO2	2	2	2			2						2	1	1
	CO3	3	2	3			2						2	2	3
	CO4	3	2	2			2						2	1	1
	CO5	3	2	3			3						2	3	2
	CO6	3	2	3			3						2	3	2

Course Content	**Week 1&2: Design advanced Solutions for Basic Data Structures** a. Fibonacci Heaps. Van Emde Boas Priority Queues. Dynamic Data Structures for Graph Connectivity/Reachability.
	Week 3&4: Understand and Identify String algorithms to solve real world problems a. Develop and use Rabin-Karp Fingerprinting algorithm for advanced problems. b. Using suffix trees solve programs from different coding platforms

	Week 5: Derive solutions for problems that make use of Dynamic Programming. a. Understanding the problem and identify the proper way of DP design using i) Trees ii) Bitmapping iii) Digit Dynamic Programming
	Week 6: Implement programs to solve problems using Tree algorithms a. Trie b. Fenwick Tree c. Segment Tree d. Sparse Table
	Week 7: Solve problems on programming platform using decomposition a. Identify solutions using Sqrt and Heavy Light decomposition
	Week 8: Solve programming problems based on Computational Geometry a. Line-segment Intersection b. Sweep Lines c. Range Trees d. Seidel's Low-dimensional LP Algorithm
	Week 9: Design efficient solutions using recursion a. Solve the problem on online coding platforms using recursion b. Identify the need of backtracking in solving the problems on online programming platforms.
	Week 10: Programs on Implementation of methods and operations on Maximum flows a. Augmenting Paths and Push-Relabel Methods. b. Minimum Cost Flows. c. Bipartite Matching.
	Week 11&12: Implement programs to solve real-world problems with NP-Completeness solutions a. Understand and analyze Polynomial time and polynomial time verification b. Using reducibility, design solutions for problems on various online coding platforms.

Text books and Reference books	**Text Book(s):** [1]. Halim, Steven and Halim, Felix, Competitive Programming 1, 2013 [2]. Reema Thareja,"Python Programming Using Problem Solving Approach", Oxford University Press, 2019. **Reference Books:** [1]. Antti Laaksonen, "Guide to Competitive Programming", 1st edition, Springer International Publishing, 2017 [2]. Ahmed Shamsul Arefin, Art of Programming Contest, ACM Solver, Second Edition, 2012 [3]. Zed Shah, "Learn Python The Hard Way", Third edition, Addison-Wesley, 2013. [4]. John V. Guttag, "Introduction to Computation and Programming Using Python", The MIT Press, 2013

E-resources and other digital material	[1]. Filipp Rukhovich, Competitive Programming for beginners, [COURSERA]. (11-12-2021), Available: https://www.coursera.org/learn/competitive-programming-for-beginners [2]. Prof Neeldhara, IIT Gandhinagar, Getting Started with Competitive Programming,[NPTEL],(11-12-2021),Available :https://onlinecourses.nptel.ac.in/noc21_cs99/preview [3]. Prof. Erik Demaine, Prof. Ronald Rivest,Prof. SriniDevadas MIT Open Courseware, Introduction to Algorithms, Getting Started with Competitive Programming,[MIT], (11-12-2021),Available:https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-spring-2008/index.htm [4]. Erik Demaine, Prof. Ronald Rivest, Prof. Srini Devadas, Lecture notes by EE & CSE of MIT https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-854j-advanced-algorithms-fall-2005/lecture-notes/ [5]. Hacker Rank, 11-12-2021 Available https://www.hackerrank.com/ [6]. Leet Code, 11-12-2021Availablehttps://leetcode.com/ [7]. Hacker Earth, 11-12-2021Available https://www.hackerearth.com/ [8]. Topcoder, 11-12-2021Available https://www.topcoder.com/challenges/ [9]. Coder Byte, 11-12-2021Available https://www.coderbyte.com/ [10]. Code wars, 11-12-2021Available https://www.codewars.com/ [11]. Code Signals, 11-12-2021Available https://codesignal.com/ [12]. Code Chef, 11-12-2021 Available https://www.codechef.com/