Department of Computer Science and Engineering
Institute of Technical Education & Research, SOA Deemed to be University

LESSON PLAN

Computer Science Workshop2 (CSE 3141)
Session: March' 2021 - July' 2021

1. **Course Number and Name:**

   CSE 3141, `Computer Science Workshop2`

2. **Credits and Course Format:**

   Grading Pattern = 5
   Credits = 4
   Course format: 8 hours/week (3 Labs/Week, 2∗3hr/Lab, 1∗2hr/Lab)

3. **Target Students:**

   Programme: B.Tech. (4th Semester)
   Branch: CSE/CSIT

4. **Instructor's Names:**

   (1) Rasmiranjan Mohakud, Assistant Professor (CSE), ITER

   (2) Samuka Mohanty, Assistant Professor (CSE), ITER

   (3) Smita Rath, Assistant Professor (CSE), ITER

   (4) Nikunja Bihari Kara, Assistant Professor (CSE), ITER

   (5) Smita Mohanty Assistant Professor (CSE), ITER

   (6) Rashmirekha Sahoo, Assistant Professor (CSE), ITER,

   (7) Bichitra Nanda Patra, Assistant Profesor (CSE), ITER

   (8) Bijendra Pratap, Assistant Professor (CSE), ITER

   (9) Sangram Panigrahi, Assistant Professor (CSIT), ITER

   (10) Santhosh Kumar, Assistant Professor (CSIT), ITER

(11) Rakesh Ranjan Swani, Assistant Professor (CSE), ITER

5. **Text Book(s):**

   (T1) Elements of Programming Interviews in Java by Aziz, Lee and Prakash

   (T2) Cracking the Coding Interview(Indian Edition), Mcdowell

   (T3) Coding Interview Questions By Karumanchi

6. **Specific Course Information:**

   (a) **Course Description:** The course topics will include study about how to design and implement different data structure using java. Data structure like Linked list, stack, queue, binary tree, heap are design, analyze and implemented using java. Different problems are solved, using dynamic programming, greedy algorithm and parallel computing.

   (b) **Prerequisites:**

   ☞ DSA, CSW1

7. **Course Outcomes (COs) :**
   By the end of course through lectures, laboratory works, and exams students will be able to:

   CO 1. Effective use of array, string, date of java and solving problems using it.

   CO 2. Structuring Data with Java, using linked list, stack, queue, binary tree, heap.

   CO 3. Understanding searching, hash table, sorting, and binary search tree.

   CO 4. Understanding recursion, Dynamic programming, Greedy Algorithms and its. application.

   CO 5. Design and implementing application based on graph, parallel Computing

   CO 6. Design and implementing some common problem. Understanding Object oriented design.

8. **Brief List of Topics to Be Covered: (L: Lecture, P: Laboratory)**

Contact hour	***Topics to be covered***	***Remarks (if any)***
Week#1:		
L-0	Introduction to CSW1, Explain Course objective , Course outcome, Program outcome.	

L-1	Computing the parity of a word, Swap bits, Reverse bits, Find a closest integer with the same weigh, Compute $X \times Y$ without arithmetical operators,Compute X/Y, Compute X^Y.	T1. C5
L-2	Reverse digits, Check if a decimal integer is a palindrome, Generate uniform random numbers, Rectangle intersection.	T1. C5
Week#2:		
L-3	The Dutch national flag problem, Increment an arbitrary-precision integer, Multiply two arbitrary-precision integers, Advancing through an array,Delete duplicates from a sorted array.	T1. Ch6
L-4	Buy and sell a stock once, Buy and sell a stock twice, Enumerate all primes to n, Permute the elements of an array, Compute the next permutation,	T1. Ch6
L-5	Sample offline data, Sample online data, Compute a random permutation, Compute a random subset, Generate nonuniform random numbers, The Sudoku checker problem, Compute the spiral ordering of a 2D array, Rotate a 2D array,Compute rows in Pascal's Triangle	T1. Ch6
Week#3:		
L-6	7.1 Interconvert strings and integers Base conversion, Compute the spreadsheet column encoding, Replace and remove, Test palindromicity, Reverse all the words in a sentence,Compute all mnemonicsfor a phone number	T1. Ch7
L-7	The look-and-say problem, Convert from Roman to decimal, Compute all valid IP addresses,Write a string sinusoidally, Implement run-length encoding, Find the first occurrence of a substring.	T1. Ch7
L-8	Merge two sorted lists, Reverse a single sublist, Test for cyclicity, Test for overlapping lists—lists are cycle-free, Test for overlapping lists—lists may have cycles.	T1. Ch8
Week#4:		
L-9	Delete a node from a singly linked list, Remove the kth last element from a list, Remove duplicates from a sorted list, Implement cyclic right shift for singly linked lists, Implement even-odd merge, Test whether a singly linked list is palindromic, Implement list pivoting, Add list-based integers	T1. Ch8

L-10	Implement a stack with max API, Evaluate RPN expressions, Test a string over for well-formedness, Normalize pathnames,	T1. Ch9
L-11	Search a postings list, Compute buildings with a sunset view, Compute binary tree nodes in order of increasing depth, Implement a circular queue,Implement a queue using stacks,Implement a queue with max API.	T1. Ch9
Week#5:		
L-12	Test if a binary tree is height-balanced, Test if a binary tree issymmetric, Compute the lowest common ancestor in a binary tree,Compute the LCA when nodes have parent pointers, Sum the root-to-leaf pathsin a binary tree.	T1. Ch10
L-13	Find a root to leaf path with specified sum, Implement an inorder traversal without recursion, Implement a preorder traversal without recursion, Compute the fcth node in an inorder traversal, Compute the successor, Implement an inorder traversal with 0(1)space, Reconstruct a binary tree from traversal data.	T1. Ch10
L-14	Reconstruct a binary tree from a preorder traversal with markers .Form a linked list from the leaves of a binary tree, Compute the exterior of a binary tree, Compute the rightsibling tree ,Implement locking in a binary tree.	T1. Ch10
Week#6:		
L-15	Merge sorted files, Sort an increasing-decreasing array, Sort an almost-sorted array,	T1. Ch11
L-16	Compute the k closeststars, Compute the median of online data,Compute the k largest elementsin a max-heap, Implement a stack API using a heap.	T1. Ch12
L-17	Search a sorted array for first occurrence of k, Search a sorted array for entry equal to itsindex, Search a cyclically sorted array	T1. Ch13
Week#7:		
L-18	Compute the integersquare root, Compute the real-square root, Search in a 2D sorted array, Find the min and max simultaneously	T1. Ch13
L-19	,Find the fcth largest element, Find the missing IP address,Find the duplicate and missing elements	T1. Ch23

L-20	Hash Tables, Test for palindromic permutations, Is an anonymousletter constructible?, Implement an ISBN cache, Compute the LCA, optimizing for close ancestors, Compute the k most frequent queries.	T1. Ch13
Week#8:		
L-21	Find the nearest repeated entriesin an array, Find the smallestsubarray covering all values, Find smallestsubarray sequentially covering all values, Find the longestsubarray with distinct entries,	T1. Ch13
L-22	Find the length of a longest contained interval, Compute the average of the top three scores, Compute allstring decompositions, Test the Collatz conjecture, Implement a hash function for chess.	T1. Ch13
L-23	Compute the intersection of two sorted arrays, Merge two sorted arrays, Remove first-name duplicates, Render a calendar, Merging intervals	T1. Ch14
Week#9:		
L-24	Compute the union of intervals, Partitioning and sorting an array with many repeated entries, Team photo day—1 , Implement a fastsorting algorithm for lists,Compute a salary threshold	T1. Ch14
L-25	Find the k largest elementsin a BST, Compute the LCA in a BST, Reconstruct a BST from traversal data,	T1. Ch15
L-26	Test if a binary tree satisfies the BST property, Find the first key greater than a given value in a BST	T1. Ch15
Week#10:		
L-27	15.6 Find the closest entriesin three sorted arrays, Enumerate numbers of the form $a + \sqrt{2}$, The most visited pages problem, Build a minimum height BST from a sorted array, Insertion and deletion in a BST, Test if three BST nodes are totally ordered, The range lookup problem, Add credits	T1. Ch15
L-28	The Towers of Hanoi problem, Generate all nonattacking placements of n-Queens, Generate permutations, Generate the powerset,	T1. Ch16
L-29	Generate allsubsets of size k, Generate strings of matched parens ,Generate palindromic decompositions, Generate binary trees ,Implement a Sudoku solver, Compute a Gray code, Compute the diameter of a tree	T1. Ch16

Week#11:		
L-30	Count the number of score combinations,Compute the Levenshtein distance, Count the number of waysto traverse a 2D array, Compute the binomial coefficients, Search for a sequence in a 2D array.	T1. Ch17
L-31	The knapsack problem, The bedbathandbeyond.com problem, Find the minimum weight path in a triangle, Pick up coinsfor maximum gain, Count the number of moves to climb stairs, The pretty printing problem, Find the longest nondecreasing subsequence.	T1. Ch17
L-32	Compute an optimum assignment of tasks, Schedule to minimize waiting time, The interval covering problem,	T1. Ch18
Week#12:		
L-33	The 3-sum problem, Find the majority element, The gasup problem, Compute the maximum water trapped by a pair of vertical lines, Compute the largest rectangle under the skyline	T1. Ch18
L-34	Search a maze, Paint a Boolean matrix, Compute enclosed regions, Deadlock detection,	T1. Ch19
L-35	Clone a graph, Making wired connections, Transform one string to another, Team photo day—2,Compute a shortest path with fewest edges	T1. Ch20
Week#13:		
L-36	Implement caching for a multithreaded dictionary, Analyze two unsynchronized interleaved threads, Implement Ssynchronization for two interleaving threads.	T1. Ch20
L-37	Implement a thread pool, Deadlock, The readers-writers problem, The readers-writers problem with write preference, Implement a Timer class,Test the Collatz conjecture in parallel.	T1. Ch20
L-38	Design a spell checker, Design a solution to the stemming problem, Plagiarism detector, Pair users by attributes, Design a system for detecting copyright infringement, Design, Design a search engine	T1. Ch21
Week#14:		
L-39	The JVM, throw vs. throws, final,finally, and finalizer, equals() vs. == ,equals() and hashCode(), List, ArrayList, and LinkedList, String vs. StringBuilder, Autoboxing, Static initialization	T1. Ch22

L-40	Template Method vs. Strategy, Observer pattern,Push vs. pull observer pattern ,Singletons and Flyweights, Adapters, Creational Patterns,Libraries and design patterns	T1 Ch23

9. **Evaluation scheme (under Grading Pattern-1) out of 100%:**

Attendance:	05%
Lab test:	20%
Mid-Term :	15%
End-Term Examination:	60%

10. **Program Outcomes & Program Specific Outcomes**
There are twelve program outcomes (1-12) and Two program specific outcomes for the Computer science & Engineering B. Tech program:
Program Outcomes (POs)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specifis Outcomes (PSOs)

PSO 1. The ability to understand, analyze and develop computer programs in the areas related to business intelligence, web design and networking for efficient design of computer-based systems of varying complexities.

PSO 2. The ability to apply standard practices and strategies in software development using open-ended programming environments to deliver a quality product for business success.

11. **Correlation between the Course Outcomes(COs), the Program Outcomes(POs) , and the Program Specific Outcomes(PSOs)**

Course Articulation Matrix:

Computer Science Workshop1 (CSE 2141)										Session: 2020-2021				
	POs												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 6	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Average	-	-	-	-	-	-	-	-	-	-	-	-	-	-