CSE330

COMPETITIVE CODING APPROACHES-TECHNIQUES

Lecture #0

Course Details

- Course Code:- CSE330
- LTP:- 2 0 1 [2 lectures, 1 practical's /week]
- Credits:- 3

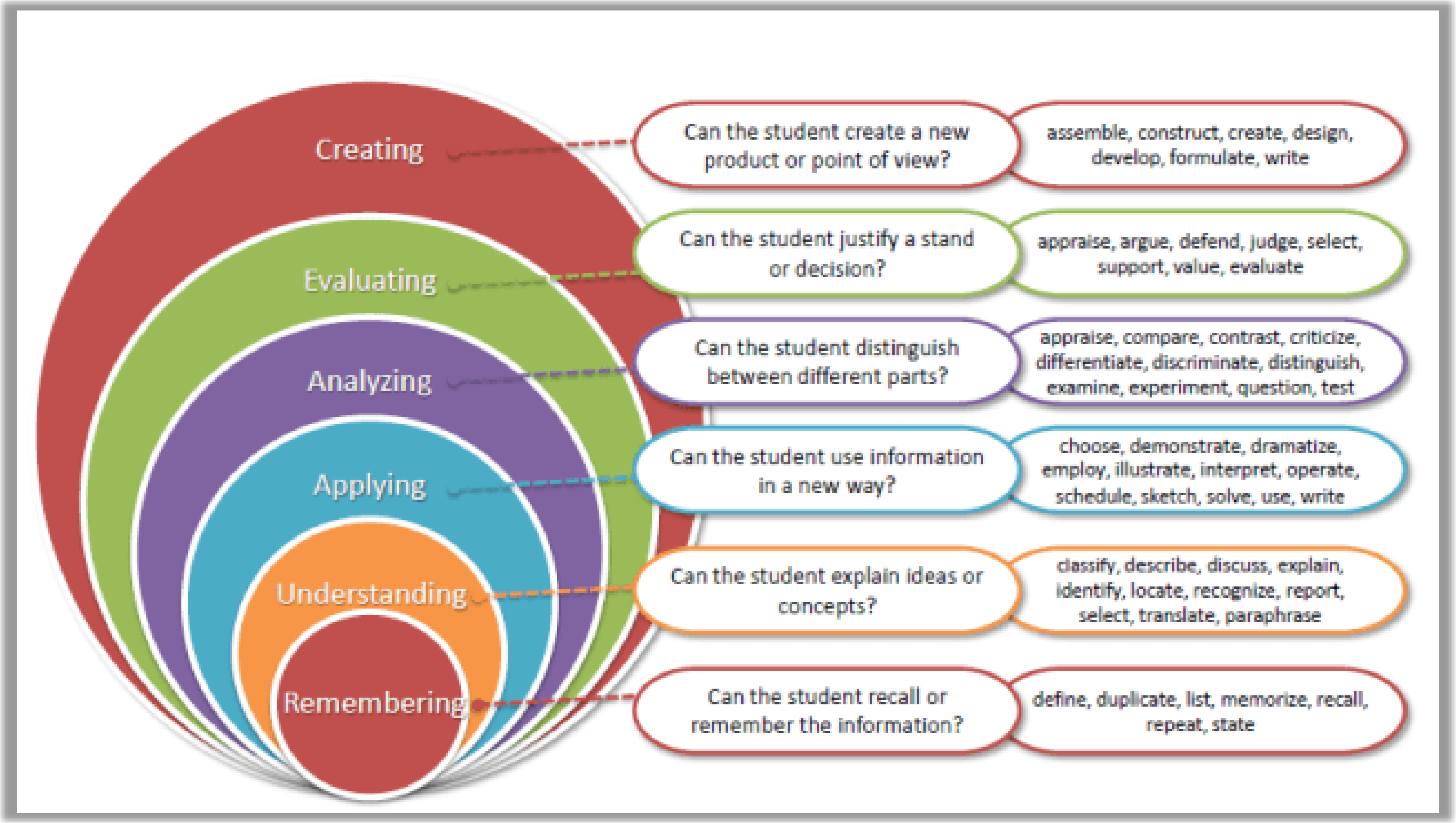
Vision

To be a globally recognized school through excellence in teaching, learning and research for creating Computer Science professionals, leaders and entrepreneurs of future contributing to society and industry for sustainable growth.

Mission

- To build computational skills through hands-on and practice-based learning with measurable outcomes.
- To establish a strong connect with industry for in-demand technology driven curriculum.
- To build the infrastructure for meaningful research around societal problems.
- To nurture future leaders through research-infused education and lifelong learning.
- To create smart and ethical professionals and entrepreneurs who are recognized globally

Revised Bloom's Taxonomy



Creating	Can the student create a new product or point of view?	assemble, construct, create, design, develop, formulate, write
Evaluating	Can the student justify a stand or decision?	appraise, argue, defend, judge, select, support, value, evaluate
Analyzing	Can the student distinguish between different parts?	appraise, compare, contrast, criticize, differentiate, discriminate, distinguish, examine, experiment, question, test
Applying	Can the student use information in a new way?	choose, demonstrate, dramatize, employ, illustrate, interpret, operate, schedule, sketch, solve, use, write
Understanding	Can the student explain ideas or concepts?	classify, describe, discuss, explain, identify, locate, recognize, report, select, translate, paraphrase
Remembering	Can the student recall or remember the information?	define, duplicate, list, memorize, recall, repeat, state

Course Outcomes

CO1 :: Analyse the space and time complexity of an algorithms

CO2 :: Determine primality testing on different algorithm

CO3 :: Apply recursion techniques on various dynamic programming problems

CO4 :: Apply tabulation and memorization in dynamic programming problem.

CO5 :: Apply various sorting techniques by computing O(n log n) complexity

CO6 :: Identify the sorting algorithm in real world problem solving

POs Specific to the course

- Apply the acknowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
- 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.
- Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PSOs specific to the course

Apply acquired skills in software engineering, networking, security, databases, intelligent systems, cloud computing and operating systems to adapt and deploy innovative software solutions for diverse applications.

Course Content

- Behaviour Analysis
- Primality Testing
- Recursion Techniques
- Advanced Techniques
- Basic Dynamic Programming
- Dynamic Programming Problems
- Efficient Sorting Algorithms & Analysis

Unit-wise course content

Unit I

Behaviour Analysis: Introduction to limits and behaviour of logic, understanding taxonomy in worst case, analysing the effectiveness and efficiency of algorithms, measuring time and space complexity of algorithm, trade-off concept.

Unit II

Primality Testing : Introduction to Primality Testing, O(sqrt(n)) Algorithm for Primality Testing, Factorization of a number, Finding prime factors by taking the square root, Fermat method, Sieve of Eratosthenes, Segmented Sieve, Sieve of Atkins , Mansi and her series, Collections of Pens, next prime palindrome.

Unit III

Recursion and Advanced Techniques : Introduction to recursion: base condition, solving problems using recursion: Classic and Modern Approaches, Direct vs. Indirect Recursion, Tailed vs. Non-Tailed Recursion, Memory Allocation in Recursion, Advantages & disadvantages of recursive programming, Backtracking, Memoization, recursive problems- next happy number, sum string, water overflow.

Unit-wise course content

Unit IV

Basic Dynamic Programming : Introduction to Dynamic Programming, Dynamic Programming Process and Techniques, Tiling problem, Tabulation vs Memoization, Optimal Substructure Property, Overlapping Subproblems Property, Formulating Dynamic Programming Problems.

Unit V

Dynamic Programming Problems : Longest Increasing Subsequence(LIS), Longest Common Subsequence (LCS), Binomial coefficient, Box Stacking, Integer Knapsack Problem (Duplicate Items Forbidden), Edit Distance, Balanced Partition Problem,

Unit VI

Efficient Sorting Algorithms & Analysis: Introduction to O(n log n) Sorting Algorithms, Iterative & Recursive Merge Sort, Quick Sort, Sorting Elements by Frequency, Finding Minimum Length Sorted Sub-array to Sort an Array, Sorting Strings, case-specific sorting of strings, Count Distinct Pairs with Difference of K.

Blended/Practical base Learning

HackerRank

Course Assessment Model

CSE330	Marks break up*
Attendance	05
CA (Two best out of Three CBTs)	50
ETP	45
Total	**100**

Note: No MTE

Detail of Academic Tasks

AT 1: BYOD PRACTICAL	Week 5
AT 2: BYOD PRACTICAL	Week 9 (After MTT)
AT 3: BYOD PRACTICAL	Week 13 (After MTT)

Note : CA is based on Mix of MCQs(10 Marks) + Coding Problems(20 Marks)

Evaluation and ETP exam pattern

ETP components	Marks Distribution
Written	20
Viva	30
Execution	50
Total	**100**

Details of Academic Task(s)

Academic Task	Objective	Detail of Academic Task	Nature of Academic Task (group/individuals)	Academic Task Mode	Marks	Allottment / submission Week
BYOD-Practical 1	To evaluate the understanding of student related to Primality testing and Recursion.	Unit1, Unit 2 syllabus will be considered for evaluation.	Individual	Online	30	4 / 5
BYOD-Practical 2	To evaluate the understanding of student related to Dynamic programming concepts to solve the problem.	Unit 3 and unit 4 syllabus will be considered for the evaluation	Individual	Online	30	8 / 9
BYOD-Practical 3	To evaluate the understanding of student related to nlogn sorting techniques and dynamic programming.	Unit 5 and Unit 6 will be considered for the evaluation	Individual	Online	30	12 / 13

Why Star Course?

- Weightage in Gate/Govt. exams
- Industry demand
  - Product Based
  - Service Based
- Higher Studies
- Government Jobs

Execution strategy for star courses

Topic: Activities	Details of the Activities Planned	Is this Activity a part of Evaluation (Yes/No): Which CA/MTE/ETE	Tentative week of conduct of activity	Responsibility: Who will ensure the conduct of the activity (specify the plan for the same)	Expected Outcome	References	Quantification	Start date	End date	Status	Proof resource
MCQ + Code Based Test	10 MCQ Question 1 Code Based Question.	Yes , (CA 1)	5th Week	Shruti (24096)	Complexity Analysis of the Problem.	Self-Prepared	1	12th September 2024	12th September 2024	Pending	
MCQ + Code Based Test	10 MCQ Question 1 Code Based Question.	Yes , (CA 2)	9th Week	Shruti (24096)	Understanding Recursive Concept.	Self-Prepared	1	10th October 2024	10th October 2024	Pending	
MCQ + Code Based Test	10 MCQ Question 1 Code Based Question.	Yes , (CA 3)	13th Week	Shruti (24096)	Understanding and Analysis the dynamic concepts.	Self-Prepared	1	8th November 2024	8th November 2024	Pending	

Details of Course Enrichment Activities

Details for Pedagogical Initiatives for the above-mentioned course

Lecture Number	Pedagogical Initiative	Expected Outcomes	References	Quantification
12,17 and 21	Students will receiving a unique problem from www.leetcode.com , problem description will be discussed and provided the link of question which is going to solve by each student	• Learn methods for the factorization of numbers. • Explore the Fermat method for primality testing.	• https://leetcode.com/problems/number-of-common-factors/ • https://leetcode.com/problems/number-of-subarrays-that-match-a-pattern-i/description/ • https://leetcode.com/problems/maximize-win-from-two-segments/	3
26 ,28 and 31	Students will receiving a unique problem from www.leetcode.com , problem description will be discussed and provided the link of question which is going to solve by each student .	• Understand memory allocation for different function calls in recursive programming. • Solve the tiling problem using dynamic programming techniques. • Apply dynamic programming techniques to solve problems	• https://leetcode.com/problems/delete-and-earn/ • https://leetcode.com/problems/rings-and-rods/ • https://leetcode.com/problems/encode-and-decode-tinyurl/	3
34, 36, 38	Students will receiving a unique problem from www.leetcode.com , problem description will be discussed and provided the link of question which is going to solve by each student	• Students will learn multiple approaches to solve a single problem and gain an understanding of how to implement different types of recursions.	• https://leetcode.com/problems/longest-increasing-subsequence/ • https://leetcode.com/problems/partition-labels/description/ • https://leetcode.com/problems/shortest-subarray-to-be-removed-to-make-array-sorted/description/	3

Evaluation strategy for all Components of the courses

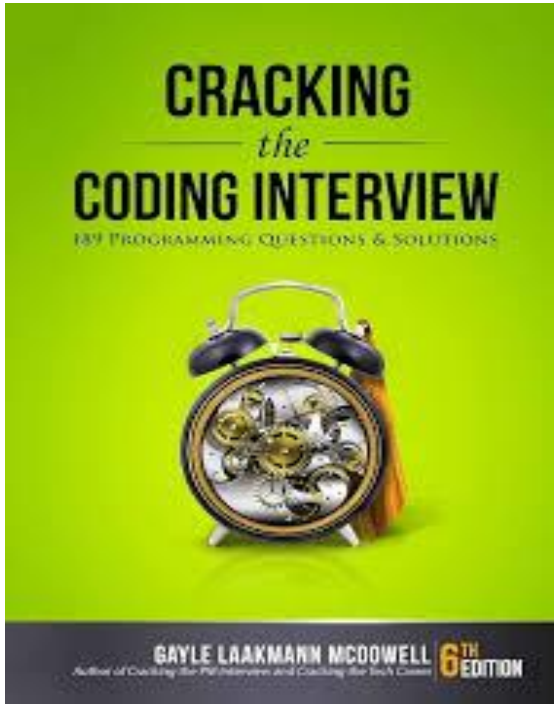
Type of Assessment (Add rows in case some other assessments are applicable)	Type of Assessment e.g. Test/Term Paper/project etc.	Details of academic task	Parameters/Rubric of Evaluation	External / Internal	Details of External Evaluations	References	Quantification
CA 1	MCQ + Code Based Test	To evaluate student's capability to solve competitive problems on prime numbers and primality testing on Online platform HackerRank	Code Structure and Organization: 5 Algorithmic Efficiency: 10 Documentation and Comments: 2 Error Handling and Exception Handling: 3 Coding Standards and Best Practices: 5 Performance Optimization: 5 Total: 30	Internal	NA	Self-Prepared	1
CA 2	MCQ + Code Based Test	To evaluate student's capability to solve competitive problems on Dynamic programming and recursion on Online platform HackerRank	Code Structure and Organization: 5 Algorithmic Efficiency: 10 Documentation and Comments: 2 Error Handling and Exception Handling: 3 Coding Standards and Best Practices: 5 Performance Optimization: 5 Total: 30	Internal	NA	Self-Prepared	1

Evaluation strategy for all Components of the courses

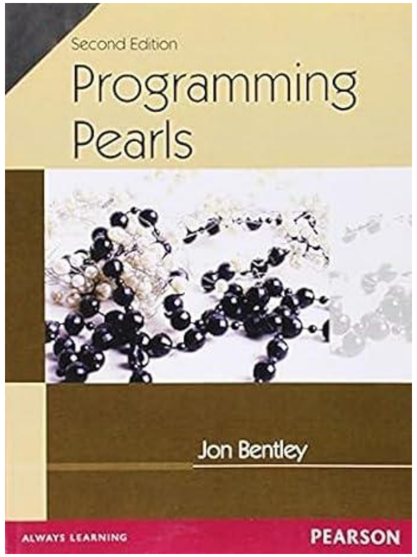
Type of Assessment (Add rows in case some other assessments are applicable)	**Type of Assessment e.g. Test/Term Paper/project etc.**	**Details of academic task**	**Parameters/Rubric of Evaluation**	**External/ Internal**	**Details of External Evaluations**	**References**	**Quantification**
CA 3	MCQ + Code Based Test	To evaluate student's capability to solve competitive problems on Dynamic programming and Sorting techniques on Online platform HackerRank	Code Structure and Organization: 5 Algorithmic Efficiency: 10 Documentation and Comments: 2 Error Handling and Exception Handling: 3 Coding Standards and Best Practices: 5 Performance Optimization: 5 Total: 30	Internal	NA	Self-Prepared	1
ETP	Code Based Question + Viva	Competitive Coding based challenges on Online platform HackerRank related to course syllabus	Code Structure and Organization: 5 Algorithmic Efficiency: 10 Documentation and Comments: 2 Error Handling and Exception Handling:3 Coding Standards and Best Practices: 5 Performance Optimization: 5 Total: 70 + 30=100	Internal	NA	self prepared	1

Text & References Books Details

Text Books: 1. CRACKING THE CODING INTERVIEW by GAYLE LAAKMANN MCDOWELL, CAREERCUP



References: 1. PROGRAMMING PEARLS by JOE BENTLEY, PEARSON



Course add On

Note: There is no course add on like MOOC/ certification applicable for this course.

Cohort

- **Software Development**
  - ✓ **Product based**
  - ✓ **Service based**

Open Educational Resources (OER)

Unit mapped	Broad topic	Sub Topic	Source Type	Source Title	*%age mapping (approx)	Source URL
Unit 1	Asymptotic Analysis (Big-O notation)	Introduction to asymptotic notations	Web Link	Types of Asymptotic Notations in Complexity Analysis of Algorithms	12%	https://www.geeksforgeeks.org/types-of-asymptotic-notations-in-complexity-analysis-of-algorithms/
Unit 2	Sqrt(n) Primality Testing	Fermat method	Web Link	Fermat’s Factorization Method	17%	https://www.naukri.com/code360/library/fermat-s-factorization-method
Unit 3	Basic Recursion		Web Link	Backtracking Algorithm	13%	https://www.geeksforgeeks.org/backtracking-algorithms/

Open Educational Resources (OER)

Unit mapped	Broad topic	Sub Topic	Source Type	Source Title	*%age mapping (approx)	Source URL
Unit 4	Basic Dynamic Programming	Optimal Substructure Property	Web Link	Optimal Substructure Property in Dynamic Programming	26%	https://www.geeksforgeeks.org/optimal-substructure-property-in-dynamic-programming-dp-2/
Unit 5	Dynamic Programming Problems	Longest increasing subsequence	Web Link	Longest Increasing Subsequence (LIS)	11%	https://www.geeksforgeeks.org/longest-increasing-subsequence-dp-3/#:~:text=Longest%20Increasing%20Subsequence%20using%20Recursion,all%20L%5Bi%5D%20values.
Unit 6	O(n logn) Sorting	Recursive Merge Sort	Web Link	merge sort using recursion in C++	13%	https://www.naukri.com/code360/campus/lovely-professional-university/discussions/exams-and-study-resources/4736

Skill Set

- Programming Skills
- Logic building
- Code analysis

Next Class :

Competitive Coding Approaches-techniques