|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SN** | **Course Code** | **TITLE OF THE COURSE** | **L** | **T** | **P** | **S** | **C** | **CH** | **Course Type\*** |
|  | **Competitive Coding – II** | **0** | **0** | **2** | **0** | **1** | **2** | **EE** |
| **20CSP351** | |  | | | | **Course Code(s)**  **20CSP-351** | | | |
| **PRE-REQUISITE** | | **21CSP-314** | | | |  | | | |
| **CO-REQUISITE** | | **21CST-352,21CST-353,21CST-354,21CST-355,21CST-357,21CST-371,21CSP-356** | | | |  | | | |
| **ANTI-REQUISITE** | | **21CST-475** | | | |  | | | |

**a. Course Description**

Competitive programming is the course in which students will learn how to apply algorithms in order to solve complex problems. The goal of this course is to teach students how to apply familiar algorithms to non-intuitive problems.

**b. Course Objectives**

Competitive programming is the course in which students will learn how to apply algorithms in order to solve complex problems.

The goal of this course is to teach students how to apply familiar algorithms to non-intuitive problems.

**c. Course Outcomes**

|  |  |
| --- | --- |
| CO1 | Understand the problem and find out better approach to solve particular problem |
| CO2 | To gain critical understanding of problem solving on Leetcode platform |
| CO3 | Apply advance programming concepts for logic building |
| CO4 | To implement the logic and find out the solution of problem and achieve all test cases |
| CO5 | To acquire proficiency in developing and implementing efficient solutions of given problems by using different approaches and achieve desirable results. |

**d. Syllabus**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unit-1** | | **Data Structures** | | **Contact Hours:15** |
| **Arrays,**  **Stacks, Queues linked list** | | <https://leetcode.com/problems/3sum/>  <https://leetcode.com/problems/jump-game-ii/>  <https://leetcode.com/problems/simplify-path/>  <https://leetcode.com/problems/implement-queue-using-stacks/>  <https://leetcode.com/problems/merge-two-sorted-lists/>  <https://leetcode.com/problems/remove-duplicates-from-sorted-list-ii/> | | |
| **String Matching** | | https://leetcode.com/problems/rotate-string/  <https://leetcode.com/problems/find-the-index-of-the-first-occurrence-in-a-string/>  <https://leetcode.com/problems/camelcase-matching/>  <https://leetcode.com/problems/repeated-string-match/>  https://leetcode.com/problems/longest-happy-prefix/ | | |
| **Heap model** | | <https://leetcode.com/problems/kth-largest-element-in-a-stream/>  <https://leetcode.com/problems/last-stone-weight/>  <https://leetcode.com/problems/cheapest-flights-within-k-stops/>  <https://leetcode.com/problems/distant-barcodes/>  <https://leetcode.com/problems/furthest-building-you-can-reach/> | | |
| **Hashing** | | <https://leetcode.com/problems/missing-number/>  https://leetcode.com/problems/word-pattern/  https://leetcode.com/problems/longest-substring-without-repeating-characters/  <https://leetcode.com/problems/longest-duplicate-substring/>  https://leetcode.com/problems/shortest-palindrome/ | | |
| **Unit-2** | **Advanced Data Structures** | | **Contact Hours:15** | |
| **Trees** | | <https://leetcode.com/problems/same-tree/>  <https://leetcode.com/problems/symmetric-tree/>  <https://leetcode.com/problems/balanced-binary-tree/>  <https://leetcode.com/problems/path-sum/>  <https://leetcode.com/problems/count-complete-tree-nodes/>  <https://leetcode.com/problems/delete-node-in-a-bst/>  <https://leetcode.com/problems/diameter-of-binary-tree/> | | |
| **Graph** | | <https://leetcode.com/problems/is-graph-bipartite/>  <https://leetcode.com/problems/gray-code/>  <https://leetcode.com/problems/group-the-people-given-the-group-size-they-belong-to/>  <https://leetcode.com/problems/the-skyline-problem/>  <https://leetcode.com/problems/find-the-difference/>  <https://leetcode.com/problems/predict-the-winner/> | | |
| **Divide and conquer** | | <https://leetcode.com/problems/count-and-say/>  <https://leetcode.com/problems/1-bit-and-2-bit-characters/>  <https://leetcode.com/problems/jewels-and-stones/>  <https://leetcode.com/problems/snakes-and-ladders/>  <https://leetcode.com/problems/water-and-jug-problem/>  https://leetcode.com/problems/find-and-replace-in-string/ | | |
| **UNIT-3** | **Advanced Data Structures** | | **Contact Hours:15** | |
| **Greedy** | | <https://leetcode.com/problems/candy/>  <https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/>  <https://leetcode.com/problems/remove-duplicate-letters/>  <https://leetcode.com/problems/can-place-flowers/>  <https://leetcode.com/problems/assign-cookies/>  https://leetcode.com/problems/three-equal-parts/ | | |
| **Backtracking** | | <https://leetcode.com/problems/binary-watch/>  <https://leetcode.com/problems/stickers-to-spell-word/>  <https://leetcode.com/problems/all-paths-from-source-to-target/>  <https://leetcode.com/problems/word-ladder-ii/>  <https://leetcode.com/problems/subsets/>  <https://leetcode.com/problems/combinations/> | | |
| **Dynamic Programming** | | <https://leetcode.com/problems/best-time-to-buy-and-sell-stock/>  <https://leetcode.com/problems/decode-ways/>  <https://leetcode.com/problems/scramble-string/>  <https://leetcode.com/problems/climbing-stairs/>  <https://leetcode.com/problems/maximum-subarray/>  <https://leetcode.com/problems/longest-palindromic-substring/>  <https://leetcode.com/problems/house-robber-ii/> | | |

**e. Assessment Pattern - Internal and External**

The performance of students is evaluated as follows:

|  |  |  |
| --- | --- | --- |
|  | **Theory** | |
| **Components** | **Continuous Internal Assessment (CAE)** | **Semester End Examination (SEE)** |
| **Marks** | 60 | 40 |
| **Total Marks** | 100 | |

**f. Internal Evaluation Component**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. No. | Type of Assessment | Weightage of actual conduct | Frequency of Task | Final Weightage in Internal  Assessment | Remarks |
| 1 | Conduct | 10 Marks per Practical | 1 per practical | 60 Marks per course |  |
| 2 | Report | 10 Marks per Practical | 1 per practical |  |
| 3 | Viva- Voce | 20 Marks per Course | 1 per Course |  |

**g. CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Outcome** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** |
| CO1 | 3 | - | 3 | 2 | 3 | - | - | 1 | 1 | - | - | - | - | - |
| CO2 | 3 | - | - | 3 | 2 | 3 | - | - | - | - | - | - | - | - |
| CO3 | 3 | 2 | - | 3 | 2 | - | - | - | - | - | - | 3 | 3 | - |
| CO4 | 3 | 3 | - | - | 3 | 2 | - | - | 2 | - | - | 3 | 3 | - |
| CO5 | 3 | - | - | - | - | - | - | 1 | 2 | 3 | 2 | 3 | - | 2 |