# CS1236: Advanced Data Structures

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Title and Code**: **Advanced Data Structures :CS1236** | | | |
| Hours per Week | | **L-T-P: 3-0-2** | |
| Credits | | **4** | |
| Students who can take | | B.Tech. Sem (VII) | |
| **Course Objective-**   * The course aims to develop a deeper understanding of algorithmic design paradigms * Design of advanced data structures for solving complex problems.   **Weightage : Theory 65%, Practical and assignments : 35%** | | | |
| **Course Outcome:**  On successful completion of this course, the students should be able to:  CS1236.1.Determine the correctness of algorithms using inductive proofs and loop invariants.  CS1236.2.Analyze algorithms using amortized analysis, including the accounting and potential methods, as required.  CS1236.3. Write programs to solve algorithmic problems using divide-and-conquer and dynamic-programming paradigms.  CS1236.4.Implement Linear Programming Techniques to solve specific problems.  CS1236.5.Analyze, implement and use heap structures and hashing techniques.  CS1236.6.Apply and implement the disjoint set data structures to solve problems modelled by the graph and application of computational Geometry.  CS1236.7.Evaluate and apply approximate algorithm design techniques for solving complex algorithmic problems. | | | |
| Prerequisites | | | **Programming language, DSA** |
| **Sr. No** | **Specifications** | | **Marks** |
| 01 | Attendance | | **5** |
| 02 | **Assignments** | | **10** |
| 03 | **Class Participation** | | **Nil** |
| 04 | **Quiz** | | **10** |
| 05 | **Theory Exam-1** | | **20** |
| 06 | Theory Exam-2 | | **Nil** |
| 07 | Theory Exam-3 | | **30** |
| 08 | Report-1 | | **Nil** |
| 09 | Report-2 | | **Nil** |
| 10 | Report-3 | | **Nil** |
| 11 | Project-1 | | **Nil** |
| 12 | Project-2 | | **Nil** |
| 13 | Viva | | **5** |
| 14 | **Lab Evaluation-1 (Continuous Assessment)** | | **10** |
| 15 | **Lab Evaluation-2 (Exam)** | | **10** |
| **16** | Course portfolio | | **Nil** |
|  | **Total (100)** | | **100** |

**Syllabus (Theory)**

**Unit 1: Amortized Analysis**: Aggregate, Accounting and Potential Method, Dynamic tables, **External Sorting**: Introduction to external sorting. Selection trees & k-way merging. Run generation—the optimal merging of runs.

**Unit 2: Approximation Algorithms:** One Way of Coping with NP-Hardness. Greedy Approximation Algorithms. Dynamic Programming and Weakly Polynomial-Time Algorithms. Linear Programming Relaxations. Randomized Rounding. Vertex Cover, Wiring, and TSP.

**Unit 3: String Matching Algorithms**: Knuth Morris Prat, and Boyer Moore. **String Processing Data Structures**: Tries, Suffix Tree, Rabin-Karp Fingerprinting Algorithm**. Disjoint Set Data Structures:** Disjoint-set operations, representation of disjoint sets, Disjoint-set forests

**Unit 4: Advance Data Structures:** Binomial Heap, Fibonacci Heap, Pairing heap, Van Emde Boas

Priority Queues. Dynamic Data Structures for Graph Connectivity/Reachability.

**Unit 5:** **Maximum Flows:**Augmenting Paths and Push-Relabel Methods. Minimum Cost Flows.

Bipartite Matching.

**Unit 6: Linear Programming:** Formulation of Problems as Linear Programs. Duality. Simple Interior

Point, and Ellipsoid Algorithms.

**Unit 7:** **Hashing**: Introduction, Perfect hash function - Cuckoo hashing, Coalesced hashing, Universal Hashing. **Applications**: Searching, Memory Indexing, Computer Graphics, Image Data Structures.

**Unit 8**: **Computational Geometry:** Convex Hull. Line-segment Intersection. Sweep Lines. Voronoi

Diagrams. Range Trees. Seidel’s Low-dimensional LP Algorithm.

**Syllabus (Practical)**

**Practical work will be based on programming exercises on topics covered in the theory syllabus.**

**Text Books:**

1. Samet, Hanan. Foundations of multidimensional and metric data structures. M. Kaufmann, 2006.
2. Mehta, Dinesh P., and Sartaj Sahni. Handbook of data structures and applications. Chapman and Hall/CRC, 2004.
3. Langsam, Yedidyah, Moshe Augenstein, and Aaron M. Tenenbaum. Data Structures using C and C++. Vol. 2. New Jersey: Prentice Hall, 2001.
4. Sartaj, Sahni. "Data Structures, Algorithms and Applications in C++." Computer Science, Singapore: McGraw-Hill (1998), reprint 2005.
5. Robert, L. Krune, Clovis L. Tondo, and Bruce P. Leung. "Data structures & Program Design in C." In O'Dougherty (production process staff workers)(second (hc) textbook ed.). Prentice-Hall, Inc. div. of Simon & Schuster, 2002.

**Reeference Books:**

1. Allen, Weiss Mark. Data structures and algorithm analysis in C++. Pearson Education India, 2007.
2. Cormen, T. H., Charles E. Leiserson, R. L. Rivest, and C. Stein. "Introduction to algorithms 2nd edition. chpater 9: Medians and order statistics."
3. Hopcroft, John E., and Jeffrey D. Ullman. Data structures and algorithms. 1983 reprint 2001.
4. Standish, Thomas A. Data structures in Java. Addison-Wesley Longman Publishing Co., Inc., 1997. Reprint Pearson Education Asia (Adisson Wesley), New Delhi, 2000
5. Knuth, Donald E. "The art of computer programming. Vol. 1: Fundamental algorithms." Atmospheric Chemistry & Physics (1978).
6. Heileman, Gregory L. "Data Structures, Algorithms, and Object-Oriented Programming. 1996.", Tata Mc-Graw Hill, 2002.

**Mapping of CO to PO**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course specific LO's contribution to PO/PSO** | **Rate the level of course specific LO's corelated with POs/PSOs ( 1: Low Corelation; 2: Moderate; 3: Substantial corelation) Leave Blank if Not Corelated** | | | | | | | | | | | | | | | | |
|
| PO1 | PO2a | PO2b | PO2c | PO3a | PO3b | PO3c | PO4a | PO4b | PO4c | PO5a | PO5b | PO6 | PO7a | PO7b | PSO1 | PSO2 |
| **CS1236.1** | 3 |  | 2 |  | 2 | 2 | 1 |  |  |  |  |  |  |  |  | 3 | 3 |
| **CS1236.2** | 2 |  | 2 |  | 2 | 2 | 1 |  | 1 |  |  | 1 |  |  |  | 3 | 3 |
| **CS1236.3** | 2 |  | 2 |  | 2 | 2 | 2 |  | 1 |  |  |  |  |  |  | 3 | 3 |
| **CS1236.4** | 2 |  | 2 |  | 2 | 2 | 2 |  | 1 |  |  |  |  | 1 | 1 | 2 | 2 |
| **CS1236.5** | 2 |  | 2 |  | 2 | 2 | 2 |  | 1 |  |  |  |  |  |  | 1 | 2 |
| **CS1236.6** | 2 |  | 1 |  | 1 |  |  |  |  |  |  |  |  |  |  | 2 | 2 |
| **CS1236.7** | 2 |  | 2 |  | 2 | 2 | 2 |  | 2 |  |  |  |  |  |  | 3 | 3 |

**Learning Activities (LA):**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **LA** | **Evaluation component used** | **CO** |
| LA.1 | Problem Solving and Programming practice | Assignment, Class participation | **CS1236.1** |
| LA.2 | Complexity analysis: Asymptotic vs Amortized | Assignment, Class participation,Theory Exam | **CS1236.1- CS1236.6** |
| LA.3 | divide-and-conquer and dynamic-programming paradigm of problem-solving | Lab evaluation of recursion, Class participation, Theory Exam. | **CS1236.3** |
| LA.4 | Implement Linear Programming Techniques to solve specific problems. | Assignment, Lab evaluation, Theory Exam | **CS1236.4** |
| LA.5 | Analyze, implement and use heap structures and hashing techniques. | Assignment, Lab evaluation,Theory exam. | **CS1236.5** |
| LA.6 | Apply and implement the disjoint set data structures to solve problems modelled by the graph and application of computational Geometry. | Assignment, Theory Exam-and Lab evaluation | **CS1236.6** |
| LA.7 | Evaluate and apply approximate algorithm design techniques for solving the complex algorithmic problem | Class participation, Lab evaluation and Theory Exam | **CS1236.7** |

**Learning Activity Articulation Matrix: (Mapping of LAs with COs)**

**Learning Activities (LA):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S. No.** | CS1236.1 | CS1236.2 | CS1236.3 | CS1236.4 | CS1236.5 | CS1236.6 | CS1236.7 |
| LA.1 | 2 |  |  |  |  |  |  |
| LA.2 | 1 | 2 | 1 | 1 | 1 | 1 |  |
| LA.3 |  |  | 3 |  |  |  |  |
| LA.4 |  |  |  | 3 |  |  |  |
| LA.5 |  |  |  |  | 3 |  |  |
| LA.6 |  |  |  |  |  | 3 |  |
| LA.7 |  |  |  |  |  |  | 3 |

**1- Low Correlation; 2- Moderate Correlation; 3- Substantial Correlation**