Course Title: Algorithms and Data Structures

Course Placement: Msc(IT) first semester

Credit Structure(L-T-P-Cr): 3-0-2-4

Course Code: IT623

Prerequisites (if any): Basic programming skills

Instructor’s Name with email: Priyanka Singh, Email: Priyanka_Singh@daiict.ac.in

Course Objectives:

• Learn different data structures such as arrays, linked list, queue, stack, trees, graph etc.
• Learn algorithms implementation to perform a specific task using these data structures.
• Learn how to measure the execution time of some of these algorithms using asymptotic notation.

Suggested Textbook/references:

• Data Abstraction and Problem Solving with JAVA: Walls and Mirrors, 1st edition, by Carrano and Prichard
• Introduction to Algorithms, 2nd edition, by Cormen, Leiserson, Rivest, Stein (CLRS)

Mode of Delivery

• Content delivery through 35-40 lectures.
• Q&A and problem solving through tutorials and quizzes.
• Labs session for algorithms and data structure implementation using languages such as Java.

Assessment method: Quizzes, tutorials, one in-semester examination, a final examination

Course Outcomes:

After completion of this course, students will be able to understand:

 The need of data structures.
 Basic data structures.
 Visualizing the problem and making an abstract model.
Lecture Plan:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Detailed Content</th>
<th>Lectures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asymptotic analysis</td>
<td>Big O, little o, omega, and theta notation, worst case and average case analysis, and solving recurrences</td>
<td>03</td>
</tr>
<tr>
<td>List data</td>
<td>Arrays, Stacks, Queues, Linked lists, Hashing</td>
<td>12</td>
</tr>
<tr>
<td>Sorting</td>
<td>Quicksort, Heap Sort, Insertion sort, Bubble sort etc.</td>
<td>03</td>
</tr>
<tr>
<td>Trees</td>
<td>Heaps, Binary search trees (BST), height of BST</td>
<td>04</td>
</tr>
<tr>
<td>Balanced BSTs</td>
<td>Red Black trees, AVL Trees</td>
<td>03</td>
</tr>
<tr>
<td>Graphs</td>
<td>Representation using adjacency matrices and adjacency lists, Graph searching algorithms BFS and DFS, Shortest-path (Dijkstra’s and Floyd’s algorithms), Minimum spanning tree (Prim’s and Kruskal’s algorithms) and Topological sort</td>
<td>06</td>
</tr>
<tr>
<td>Algorithm Design Techniques</td>
<td>Brute-force algorithms, Divide-and-conquer, Dynamic programming, Greedy algorithms, Backtracking and Heuristics</td>
<td>04</td>
</tr>
</tbody>
</table>