Program Content

Semester	**III**	
Course Code:	**IT3206**	
Course Name:	**Data Structures and Algorithms**	
Credit Value:	**3**	
Core/Optional	**Core**	
Hourly Breakdown	Theory	Independent Learning
	45 hrs.	105 hrs.

Course Aim:

This course module provides fundamental knowledge in the application of different data structures and algorithmic processes. Further, this provides knowledge in the sorting and searching algorithms. Finally provides skills to use data structures and algorithms for problem solving activities.

Intended Learning Outcomes:

After successfully completing this course, students should be able to

- Explain common data structures and their applications
- Explain common searching and sorting algorithms and their applications
- Apply data structures and algorithms in appropriate real-world problem-solving activities.

Course Content: (Main Topics, Sub topics)

Topic	Theory (Hrs)
1. Introduction to Data Structures	03
2. Arrays and Linked Lists	05*
3. Stacks and Queues	05*
4. Recursion	04*
5. Trees	08*
6. Graphs	08*
7. Analysis of Algorithms	05*
8. Sorting and Searching Algorithms.	07*
Total	**45**

* Students are expected to do practical and tutorials to strengthen their knowledge of these sections

Section 1: Introduction to Data Structures (03 hours)

Material /Sub Topics

1.1. Introduction to data structures [Ref 3:pg.9, Ref 1:pg.230]
1.2. Use of Data Structures [Ref 3:pg.10-11]
- **1.2.1.** Real- world data storage [Ref 3:pg.10-11]
- **1.2.2.** Programmer's Tools [Ref 3:pg.11]
- **1.2.3.** Real-world Modeling [Ref 3:pg.11]

1.3. Overview of Data Structures [Ref 3:pg.11-12]
1.4. Classification of Data Structures [Ref 4:pg.18]

Section 2: Arrays and Linked Lists (05 hours)

Material /Sub Topics

2.1. Introductuion to Arrays [Ref 1:pg.37-45, Ref 4:pg.75-77]
- **2.1.1.** One dimensional arrays[Ref 1:pg.37-42, Ref 3:pg.39-42]
- **2.1.2.** Multi dimensional arrays [Ref 1:pg.45]
- **2.1.3.** Basic operations on arrays [Ref 3:pg.43-46]

2.2. Comparison of Arrays and Linked Lists [Ref 4:pg.76-78, pg.174]
2.3. Singly Linked Lists [Ref 1:pg.620-630, Ref 4:pg.78-88,]
2.4. Doubly Linked Lists [Ref 1:pg.630-633, Ref 4:pg.88-96, Ref 2:pg.236-239]
2.5. Circular Linked Lists [Ref 4:pg.96-107]
2.6. Skip Lists [Ref 4:pg.118-121]

Section 3: Stacks and Queues (05 hours)

Material /Sub Topics

3.1. Stacks
- **3.1.1.** Intoduction to Stacks [Ref 4:pg.163-165,Ref 2:pg.232]
  - **3.1.1.1.** Array based Stack implementation [Ref 4:pg.165-167, Ref 3:pg. 116-123, Ref 1:pg.596-599, Ref 2:pg.233-234]
  - **3.1.1.2.** Linked List based Stack implementation [Ref 4:pg.171-175, Ref1:pg.606-609]
- **3.1.2.** Applications of Stacks [Ref 4:pg.165, Ref 4:pg.174-185]

3.2. Queues
- **3.2.1.** Introduction to queues [Ref 4:pg.205-206, Ref 2:pg.234]
  - **3.2.1.1.** Array based Queue implementation [Ref 1:pg.260-261, pg.600-604, Ref 3:pg.132-142, Ref 2:pg.235]
  - **3.2.1.2.** Linked List based Queue implementation [Ref 4: pg. 212-213, Ref1:pg.609-612]
- **3.2.2.** Applications of queues [Ref 4:pg.207]
- **3.2.3.** Circular queue [Ref 4:pg.207-208, Ref 3:pg.36-137]
- **3.2.4.** Priority queue [Ref 4:pg.369-371, Ref 1:pg.274-276, Ref 3:pg.143-149]

Section 4: Recursion (4 hours)

Material /Sub Topics

4.1. Introduction to Recursion [Ref 1:pg.294, Ref 4:pg. 62-65]
4.2. Recursion versus iteration [Ref 4:pg.65-66]
4.3. How recursion works [Ref 4:pg.66, Ref 1:pg.302-303]
4.4. Implementation in recursion [Ref 4:pg.63-68, Ref 1:pg.304-314, Ref 5:pg.184-197]
- **4.4.1.** Tail recursion [Ref 5:pg.184-185]
- **4.4.2.** Nontail recursion [Ref 5:pg.185-191]
- **4.4.3.** Indireect recursion [Ref 5:pg.191-193]
- **4.4.4.** Nested Recursion [Ref 5:pg.193]
- **4.4.5.** Excessive recursion [Ref 5:pg.194-197]

Section 5: Trees (08 hours)

Material /Sub Topics

5.1. Introduction to trees [Ref 1: pg.651-665, Ref 4: pg.224-231, Ref 3: pg.365-371]
- **5.1.1.** Introduction to general trees, properties of general trees [Ref 1: pg.651-658, Ref 4: pg.224-227]
- **5.1.2.** Introduction to binary trees, properties of binary trees [Ref 1: pg.658-665, Ref 4: pg.227-231, Ref 3: pg.365-371]

5.2. Tree traversal [Ref 1: pg.667-679, Ref 4: pg.231-239, Ref 3: pg.381-388]
- **5.2.1.** Depth First Traversal [Ref 1: pg.671-678, Ref 4: pg.231-238, Ref 3: pg.381-388]
  - **5.2.1.1.** Preorder Traversal
  - **5.2.1.2.** Inorder Traversal
  - **5.2.1.3.** Postorder Traversal
- **5.2.2.** Breadth First Traversal (Level Order Traversal) [Ref 1: pg.678-679, Ref 4: pg.238-239]

5.3. Binary search trees [Ref 1: pg.687-714, Ref 2: pg.286-289, Ref 4: pg.301-312, Ref 3: pg.371-376]
5.4. Tree Balancing [Ref 1: pg.706-714, Ref 4: pg.329-342, Ref 5: pg.255-266]
- **5.4.1.** Introduction to tree balancing [Ref 5: pg.255-258]
- **5.4.2.** Global tree balancing (The DSW algorithm) [Ref 5: pg.258-261]
- **5.4.3.** Local tree balancing (AVL Tree) [Ref 1: pg.706-714, Ref 4: pg.329-342, Ref 5: pg.258-261]

5.5. Heaps [Ref 1: pg.808-818, Ref 4: pg.372-386, Ref 3: pg.580-601]

Section 6: Graphs (08 hours)

Material / Sub Topics

6.1. Introduction to graphs [Ref 1: pg.528-529, Ref 4: pg.426-432, Ref 3: pg.615-619]

- 6.1.1. Directed graphs, undirected graphs, weighted graphs [Ref 1: pg.528-529, Ref 4: pg.426-431, Ref 3: pg.615-619]
- 6.1.2. Applications of graphs [Ref 4: pg.432]

6.2. Diferent types of Graph representations [Ref 1: pg.530-539, Ref 2: pg.589-592, Ref 4: pg.432-512, Ref 3: pg.619-622]

- 6.2.1. Array based implementation (Adjacency matrix, path matrix) [Ref 4: pg.432-434, Ref 3: pg.620-621]
- 6.2.2. Linked-list based implementation (Adjacency list) [Ref 1: pg.530-539, Ref 4: pg.434-437, Ref 3: pg.621-622]

6.3. Graph traversal [Ref 2: pg.594-611, Ref 4: pg.438-449, Ref 3: pg.623-642]

- 6.3.1. Depth first search (DFS) [Ref 2: pg.594-602, Ref 4: pg.438-445, Ref 3: pg.625-634]
- 6.3.2. Breadth first search (BFS) [Ref 2: pg.603-611, Ref 4: pg.445-449, Ref 3: pg.636-642]

6.4. Shortest path algorithms graph [Ref 1: pg.539-554, Ref 2: pg.651-654, pg.658-662]

- **6.4.1.** Shortest path in unweighted graph [Ref 1: pg.539-545]
- **6.4.2.** Shortest path in weighted graph (Dijkstra's algorithm) [Ref 1: pg.545-552, Ref 2: pg.658-662]
- **6.4.3.** Shortest path weighted graph with negative edges (Bellman-Ford Algorithm) [Ref 1: pg.552-554, Ref 2: pg.651-654]

Section 7: Analysis of Algorithms (05 hours)

Material / Sub Topics

7.1. Introduction to analysis of algorithms [Ref 1: pg.188-192, Ref 4: pg.19-23, Ref 2: pg.23-29]
7.2. Types of analysis [Ref 4: pg.23-24, Ref 2: pg.43-44]
7.3. Big-O notation [Ref 1: pg.201-205, Ref 4: pg.24-26, Ref 2: pg.47-48]

Section 8: Sorting and Searching Algorithms (07 hours)

Material / Sub Topics

8.1. Introduction to Iiterative and, Divide and Conquer Methodology [Ref 2: pg.29-34, pg.65-67]
8.2. Sorting

- **8.2.1.** Iterative Method [Ref 2: pg.16-22, Ref 3: pg.79-103]
  - **8.2.1.1.** Bubble sort [Ref 3: pg.79-89]
  - **8.2.1.2.** Selection sort [Ref 3: pg.89-95]
  - **8.2.1.3.** Insertion sort [Ref 2: pg.16-22, Ref 3: pg.95-103]
- **8.2.2.** Divide and Conquer Method [Ref 1: pg.361-369, pg.823-826, Ref 3: pg.279-294, pg.333-359, : pg.601-610, Ref 2: pg.151-169, pg.170-190]
  - **8.2.2.1.** Merge sort [Ref 1: pg.361-364, Ref 3: pg.279-294]

8.2.2.2. Quick sort [Ref 1: pg.364-369, Ref 3: pg.333-357, Ref 2: pg.170-190]
8.2.2.3. Radix Sort [Ref 3: pg.357-359]
8.2.2.4. Heap Sort [Ref 1: pg.823-826, Ref 3: pg.601-610, Ref 2: pg.151-169]

8.3. Searching algorithms
- **8.3.1.** Linear search [Ref 1: pg.207-208]
- **8.3.2.** Binary search [Ref 1: pg.208-210]
- **8.3.3.** Interpolation search [Ref 1: pg.211-212]

Teaching /Learning Methods:
You can access all learning materials and this syllabus in the VLE: http://vle.bit.lk/, if you are a registered student of the BIT degree program.

Assessment Strategy:

Continuous Assessments/Assignments:
The assignments consist of two quizzes, assignment quiz 1 (It covers the first half of the syllabus) and assignment quiz 2 (It covers the second half of the syllabus). The maximum mark for a question is 10 and the minimum mark for a question is 0 (irrespective of negative scores). Final assignment mark is calculated considering both assignments, and students will have to obtain at least 40% for each assignment. Students are advised to complete online assignments before the given deadline. It is compulsory to pass all online assignments to qualify to obtain the Level II, Higher Diploma in IT (HDIT), certificate.

In the course, case studies/Lab sheets will be introduced, and students have to participate in the learning activities.

Final Exam:
Final examination of the course will be held at the end of the semester. The course is evaluated using a two hour question paper which consists of 25 MCQ (1 hour) and 2 Structured Questions (1 hour).

References/ Reading Materials:

Main references

Ref 1: Data Structures and Problem solving using Java by Mark Allen Weiss, 4th Edition, ISBN 9780321541406

Ref 2: Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, The MIT Press, 3rd edition,ISBN 978-0262033848

Ref 3: Data Structures and Algorithms in Java by Robert Lafore, GC Join for Techmedia, 2nd edition, ISBN 978-0672324536

Supplimentry references

Ref 4: Data Structures and Algorithms Made Easy, by Narasimha Karumanchi, 5th Edition, [Online source : https://bit.ly/data-structures-and-algorithms-narasimha-karumanchi-pdf]

Ref 5: Data Structures and Algorithms in Java by Adam Drozdek, Cengage Learning Asia, 3rd edition. ISBN 978-9814239233

**Note : Ref 5 should be updated with 4th edition and page numbers should be updated accordingly.