UNDERSTAND ALGORITHMS AND DATA STRUCTURES

UNIT CODE: CT/OS/CS/CR/09/6/B

UNIT DESCRIPTION

This unit covers the competencies required to understand algorithms and data structure. It involves Understand fundamental principles of algorithms understanding fundamental concepts of data structures, linked lists, stacks and queues, search techniques and sorting techniques

ELEMENT These describe the **key outcomes** which make up **workplace function**.	**PERFORMANCE CRITERIA** These are **assessable** statements which specify the required level of performance for each of the elements. ***(Bold and italicized terms are elaborated in the range.)***
1. Understand fundamental principles of algorithms	1.1 Algorithm is defined 1.2 Characteristics of an Algorithm are explained 1.3 Algorithm writing is demonstrated 1.4 Algorithm Analysis is explained 1.5 ***Complexities*** of algorithms are explained 1.6 ***Greedy algorithms*** are outlined 1.7 Divide and conquer is demonstrated
1. Understand fundamental concepts of data structures	2.1 ***Key concepts in data structures*** are explained 2.2 Arrays are explained 2.3 Array insertion operations are explained 2.4 Array delete, search and update are explained 2.5 Array operations are demonstrated using C++
2. Understand Linked lists	3.1 Linked lists are explained 3.2 Doubly linked lists are explained. 3.3 Circular linked lists are explained. 3.4 ***Basic operations for the various linked lists*** are demonstrated using C++
3. Understand Stacks and Queues	4.1 Stacks and queues are defined 4.2 Stack and queue representation are outlined 4.3 ***Basic operations in stacks*** are explained 4.4 ***Basic operations in Queue*** are explained 4.5 Basic operations in stacks and queue are demonstrated using C++
4. Understand Search Techniques	5.1 Search is defined 5.2 Linear Search is explained 5.3 Binary Search is explained 5.4 Search techniques are demonstrated using C++

5. Understand Sorting Techniques	6.1 Sorting is defined 6.2 ***Categories of sorting techniques*** are outlined 6.3 ***Types of Sorting algorithms*** are explained 6.4 Sorting algorithms are demonstrated using C++

RANGE

This section provides work conditions to which the performance criteria apply. It allows for different work environments and situations that will affect performance.

Variable	**Range**
Complexities may include but is not limited to:	• Space • Time
Greedy algorithms may include but is not limited to:	• Counting coins
Key concepts in data structures may include but is not limited to:	• Data • Object • Type
Basic operations for various linked lists may include but is not limited to:	• Insertion • Deletion • Reverse • Display
Basic operations in stacks may include but is not limited to:	• Push • Pop
Basic operations in queues may include but is not limited to:	• Enqueue • Dequeue
Categories of sorting techniques may include but is not limited to:	• In place • Not in place • Stable • Not stable • Adaptive • Non-adaptive
Types of Sorting algorithms may include but is not limited to:	• Bubble sort • Insertion sort • Selection sort

REQUIRED SKILLS AND KNOWLEDGE

This section describes the skills and knowledge required for this unit of competency.

Required skills

The individual needs to demonstrate the following skills:

- Communications (verbal and written);
- Time management;
- Problem solving;
- Planning;
- Decision Making;
- Research;

Required knowledge

The individual needs to demonstrate knowledge of:

- Fundamental principles of algorithms
- Fundamental concepts of data structures
- Linked lists
- Stacks and queues
- Search techniques
- Sorting techniques

EVIDENCE GUIDE

This provides advice on assessment and must be read in conjunction with the performance criteria, required skills and understanding and range.

1. Critical Aspects of Competency	Assessment requires evidence that the candidate: 1.1 Wrote an algorithm 1.2 Demonstrated array operations 1.3 Demonstrated basic operations for the various linked lists 1.4 Demonstrated basic operations in stacks and queues 1.5 Demonstrated search techniques 1.6 Demonstrated sorting algorithms
2. Resource Implications	The following resources should be provided: 2.1 Access to relevant workplace where assessment can take place 2.2 Appropriately simulated environment where assessment can take place

3. Methods of Assessment	Competency may be assessed through: 3.1 Oral questioning 3.2 Practical tests 3.3 Observation 3.4 Written tests
4. Context of Assessment	Competency may be assessed 4.1 Off the job 4.2 on the job 4.3 During industrial attachment
5. Guidance information for assessment	5.1 Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended.