Republic of Rwanda
Ministry of Education

RTB | RWANDA TVET BOARD

DATA STRUCTURES AND ALGORITHMS USING C

GENDS501

Apply Data Structures and Algorithms using C

Competence

RQF Level: 5

Learning Hours
100

Credits: 10

Sector: ICT and Multimedia

Trade: Computer System and Architecture

Module Type: General

Curriculum: ICTCSA5001-TVET Certificate V in Computer System and Architecture

2024/25

Issue Date: May 2024

Purpose statement	This general module describes the knowledge, skills and attitude required to apply Data Structure and Algorithm using C programming. This module is intended to prepare learners pursuing TVET Level 5 in computer system and architecture. At the end of this module, the learners will be able to identify Data Structures and Algorithms, apply Linear Data Structure and Apply Non-linear Data Structure.					
Learning assumed to be in place	N/A					
Delivery modality	**Training delivery**		**100%**	**Assessment**		**Total 100%**
	Theoretical content		**30%**	Formative assessment	**30%**	**50%**
	Practical work:		**70%**		70%	
	Group project and presentation	**30%**				
	Individual project /Work	**40%**				
				Summative Assessment		**50%**

Elements of Competence and Performance Criteria

Elements of competence	Performance criteria
1. **Identify Data Structures and Algorithms**	1.1. Data structures concepts are clearly identified based on intended use.
	1.2. Algorithms concepts are properly analysed based on system specification and intended use.
	1.3. Algorithms are properly written based on problem to be solved and their classifications.
	1.4. C programming environment is properly prepared based on language standards.
2. **Apply Linear Data Structure**	2.1. Linear Data Structures concepts are clearly identified based on intended use.
	2.2. Arrays are properly applied based on their use cases.
	2.3. Linked List is properly applied based on their uses cases.
	2.4. Queue is properly applied based on algorithm
	2.5. Stack is properly applied based on algorithm
3. **Apply Non-Linear Data Structure**	3.1 Non-Linear Data structures concepts are clearly identified based on intended use.
	3.2. Tree is properly applied based on its intended uses
	3.3. Graph is properly applied based on its intended uses
	3.4. Tables are properly applied based on their intended use

Knowledge	Skills	Attitude
• Description of Data structures concepts and Algorithms • Classification of sorting algorithms • Identification of Linear Data	• Analysing algorithms • Prepare C programming environment. • Writing algorithm • Testing C programming environment.	• Being Innovative • Having Time management • Having Self Confidence • Having Teamwork speed

Knowledge	Skills	Attitude
structure concepts • Identification of Non-Linear Data structure concepts	• Applying arrays. • Applying stacks. • Applying queue. • Applying linked list. • Applying tree. • Applying graph. • Applying tables.	• Being Flexible • Being Self-learner • Making Decision Maker • Having Critical Thinker • Being Persistence • Having Self-motivated • Having Reasoning • Being Open minded

Course content	
Learning outcomes	**At the end of the module, the learner will be able to:** 1. Identify Data Structures and Algorithms 2. Apply Linear Data Structures 3. Apply Non-Linear Data Structures
Learning outcome 1: Identify Data Structures and Algorithms	**Learning hours: 20 hours**

Indicative content

- **Identification of data structures concepts**
  - ✓ Description of key terms
    - ✚ Data
    - ✚ Structure
    - ✚ Data structures

    - List
    - Searching
    - Sorting
    - Keys
    - Index
  - Classifications of data structures
    - Linear
    - Non-linear
  - Data types
    - Built-in
    - Derived
    - User defined
    - Abstract Data Type(ADT)
  - List representation
  - Data structure operations
    - Traversing
    - Searching
    - Inserting
    - Deleting
    - Sorting
    - Merging
- **Description of algorithms**
  - Asymptotic notation
  - Searching algorithms
    - Binary search
    - Linear search
    - Depth-First Search (DFS)
    - Breadth-First Search (BFS)

- ✓ Sorting algorithms
  - Selection sort
  - Bubble sort
  - Insertion sort
  - Merge sort
  - Quick sort
  - Shell sort
  - Heap sort
  - Radix sort
  - Counting sort
  - Bucket sort
- ✓ Classification of sorting algorithms by:
  - Number of comparisons
  - Number of swaps
  - Memory usage
  - Recursion
  - Stability
  - Adaptability
  - Internal sorting
  - External sorting

- **Analysing algorithmic concepts**
  - ✓ Algorithmic cases
    - Worst-case
    - Average-case
    - Best-case
  - ✓ Complexity
    - Time complexity
    - Space complexity

        - Test time and space complexity

- **Writing algorithms**
    - ✓ Develop Data structures
        - Using structured English
        - Using structured pseudocodes
    - ✓ Perform sorting operations
    - ✓ Perform searching operations
- **Preparation of C programming environment**
    - ✓ Tools installation
        - IDE installation
        - Compiler installation
        - Setup environment variable path
    - ✓ Test development environment
        - Execute default program
        - Write samples for case scenarios(use C programming)

Resources required for the learning outcome	
Equipment	• Computer
Materials	• Internet • Training Manual
Tools	• C Compiler • Integrated Development Environment (IDE) • Text Editors • VisuAlgo
Facilitation techniques or Learning activity	• Demonstration • Individual practical exercises • Group discussion • Trainer guided

Formative assessment methods /(CAT)	● Written assessment ● Performance assessment

Learning outcome 2: Apply Linear data structure	**Learning hours: 40**
Indicative content	

- **Identification of linear data structure concepts**
    - ✓ Description of linear data structure
        - Definition of key terms
        - Types
        - Characteristics
    - ✓ Operations performed on linear data structures.
- **Applying arrays data structure**
    - ✓ Declaring arrays
    - ✓ Initializing arrays
    - ✓ Accessing array Elements
    - ✓ Iterating over arrays
    - ✓ Array types
        - One dimensional array
        - Multidimensional arrays
    - ✓ Apply array basic operations in algorithms
    - ✓ Implementation of array basic operations in C
- **Applying linked list data structure**
    - ✓ Description of linked list data structure
        - Definition of key terms
        - Link

        - Next
        - Linked List
    - Representation of linked list
    - Types of linked list
        - Single
        - Doubly
        - Circular
    - Apply linked lists basic operations in algorithms.
    - Implementation of linked list basic operations in C
- **Applying queue data structure**
    - Description of queue data structure
        - Definition of key terms
        - Basic terminologies
    - Representation
    - Circular queue
    - Apply queue primary functions.
    - Apply queue basic operations in algorithms.
    - Implementation of queue basic operations in C
- **Applying stack data structure**
    - Description of stack data structure
        - Definition of key terms
        - Basic terminologies
    - Representation
    - Apply stack primary functions.
    - Apply stack basic operations in algorithms.
    - Implementation of stack basic operations in C

Resources required for the indicative content	
Equipment	• Computer

Materials	• Internet Connection • Training Manual
Tools	• C Compiler • Integrated Development Environment (IDE) • VisuAlgo • Text editors
Facilitation techniques or Learning activity	• Demonstration • Individual practical exercises • Group discussion • Trainer guided
Formative assessment methods /(CAT)	• Written assessment • Performance assessment

Learning outcome 3: Apply Non-linear Data Structure	**Learning hours: 40 hours**
Indicative content	
• **Identification of non-linear data structures concepts** ✓ Description of non-linear data structures ✚ Definition of key terms ✚ Basic terminologies ✓ Types of non-linear data structures ✓ Characteristics ✓ Operations performed on linear data structures ✓ Applications • **Applying tree data structure**	

  - Description of tree data structure
    - Definition of key terms
    - Basic terminologies
    - Types
  - Tree representation
  - Apply tree basic operations algorithms
  - Huffman’s tree
  - Implementation of tree operations in C
- **Applying graph data structure**
  - Description of graph data structure
    - Definition of key terms
    - Basic terminologies
    - Types
  - Graph representation
  - Apply graph basic operations algorithms
  - Apply Shortest path algorithms
    - Minimum spanning tree (Prim’s Algorithm, Kruskal’s Algorithm)
    - Dijkstra’s algorithm
    - Warshall’s algorithm
- **Applying hash table data structure**
  - Description of table data structure
    - Definition of key terms
    - Hash tables
    - Bucket
    - Hash functions
    - Load factor
    - Collision resolution
  - Apply hashing functions algorithms

✓ Apply collision resolution techniques ✓ Implementation of tables data structure in C	
Resources required for the indicative content	
Equipment	● Computer
Materials	● Internet Connection ● Training Manual
Tools	● C Compiler ● Integrated Development Environment (IDE) ● Text Editors ● VisuAlgo
Facilitation techniques or Learning activity	● Demonstration ● Individual practical exercises ● Group discussion ● Trainer guided
Formative assessment methods /(CAT)	● Oral Assessment ● Written assessment ● Performance assessment

References

geeksforgeeks.org. (n.d.). *Graph Data Structure and Algorithms*. Retrieved from geeksforgeeks.org: https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/

Karumanchi, N. (2011). *Data Structures and Algorithms.* CareerMonk Publications.

lucknow. (2007). *DATA STRUCTURES USING C.* NEW YORK: TATA McGraw-Hill.

Srivastava, S. K. (2004May 30, 2004). *Data Structures through C in Depth.* DELHII: Bpb Publications.

Thareja, R. (2014). *Data Structures using C.* DELHI: Oxford University Press.

Glossary

Data structure: is a way of organizing and storing data in a computer system so that it can be accessed and manipulated efficiently.

Algorithm: refers to a step-by-step procedure or a set of rules used to solve a problem or perform a specific task

Linear: is a type of data organization where elements are arranged in a sequential order.

Non-Linear: is a type of data organization where elements are not arranged in a sequential order.

Array: is a fundamental data structure used to store a collection of elements of the same data type

Stack: is a fundamental data structure that follows the Last In, First Out (LIFO) principle

It is used to store a collection of elements with two primary operations: push and pop.

Queue: is a fundamental data structure that follows the First In, First Out (FIFO) principle used to store a collection of elements where elements are inserted at the rear (enqueue) and removed from the front (dequeuer)

Tree: is a non-linear data structure that is widely used in computer science and programming for organizing hierarchical data

Graph: is a non-linear data structure used to represent relationships or connections between a set of objects

Abbreviation

DFS: Depth-First Search

BFS: Breadth-First Search

LIFO: Last In, First Out

FIFO: First In, First Out

ADT: Abstract Data Type