

INTRODUCTION TO DATA STRUCTURE AND ALGORITHM

	Programme: Diploma in Computer Technology	Course Code: CPT 223	Credit Hours: 4
	Subject/Course: Introduction to Data Structure and Algorithm		Theoretical: 2 hours/week
	Year: Two Semester: Two	Pre-requisite:	Practical: 2 hours /week

General Objectives:

This course focuses on how to store data in your computer (data structure) and how to efficiently manipulate these data (algorithm analysis). Students will learn basic concepts and principles of various abstract data types, file structures, and algorithm analysis techniques, and will gain practical experience and programming skills through course projects

On completion of this course the student should be able to:

- 1.0 Understand data structure fundamentals
- 2.0 Review C++/JAVA and introduce Algorithm analysis
- 3.0 Know tools for studying data: symbols, relations, and graph
- 4.0 Know sets relations and string structure
- 5.0 Know the properties of order and linear list.
- 6.0 Know simple linear data Structure
- 7.0 Know non linear data structure
- 8.0 Understand Sorting Techniques
- 9.0 Understand Searching Techniques

General Objective 1: Understand concepts of data structure and tools for studying						
Theoretical Content				Practical Content		
Week/s	Specific Learning Outcomes	Teacher's activities	Resources	Specific Learning Outcomes	Teacher's activities	Resources
1	Be able to: <ul style="list-style-type: none"> • Define data structure • Define data attributes; name, value range, data types. • Define units for identify data character, fields, sub fields, records, files. 	Discuss concept of data structure Explain data attributes, name value range and data types Explain concepts of character, fields, sub field, records and files	White Board. PC loaded with PowerPoint and connected to an OHP	Be able to use Data attributes, fields, sub fields, records and files.	Demonstrate using relevant examples Concepts of attributes, name, value range and data types. Concept of character, fields, sub field, records and files.	Networked PC's loaded with relevant software
General Objective 2: Review a high level language such as C, C++, or Java and introducing Algorithm analysis						
Week	Specific Learning Outcomes	Teacher's activities	Resources	Specific Learning Outcomes	Teacher's activities	Resources
2	Understand a chosen programming language Code To be able to describe : <ul style="list-style-type: none"> - Algorithm analysis - Running time calculations 	To review a chosen programming language Features To explain : <ul style="list-style-type: none"> - Algorithm analysis 	Capability to project programming language Features , Algorithm	- The ability to write a simple program that	- Supervise the laboratory and support students in writing a simple	Personal Computer loaded with Programming

		- Explaining Running time calculations	analysis Explaining Running time calculations A comprehensive workbook of data structure fundamentals with some specific programming language such as C, C++, or Java.	includes and covers : a chosen programming language Features . - The ability to analyse Algorithm - The ability to calculate Running time.	programs that includes and covers chosen programming language Features - Support students to analyse Algorithm - Support students to calculate Running time	Language compiler such as C++ or Java compiler
Week/s	General Objective 3: Know tools for studying data: symbols, relations, and graph					
	Theoretical Content			Practical Content		
	Be able to:	Explain the meaning of	White	Be able to	Demonstrate	Networke

3	<ul style="list-style-type: none"> Define symbols, relations and graph. Explain the symbols for expressing relations among data. Position relation cell contents, record location, transfer key. Order relation; record rank, cell rank. State properties of graph: routes, edge, sequences, directed and non-directed. Describe operations such as precede, less than points to, move to, search, change, entry. 	<p>data structure. Discuss symbols, relations and graph</p> <p>Discuss the symbols for expressing relations among data, position relation cell contents, record location, and transfer key.</p> <p>Explain the properties of graph: routes, edge, sequences, directed and non-directed.</p> <p>Describe operations such as precede, less than points to, move to, search, change, entry.</p>	Board. PC loaded with PowerPoint and connected to OHP	use symbols, relations and graph.	using relevant examples on how to use symbols, relations and graph	d PC's loaded with relevant software.
	General Objective 4: Know sets relations and string structure					
	Theoretical Content			Practical Content		
Week	Specific Learning Outcomes	Teacher's activities	Resources	Specific Learning Outcomes	Teacher's activities	Resources
4	Be able to: <ul style="list-style-type: none"> Define sets and relation Define the elements of set, subsets, super sets, 	Discuss Sets and relations Concepts of subsets,	Board & Markers; Networked PC's loaded	Be able to write simple programs to carry out set	Demonstrate giving real life example.	Board & Markers; Networked PC's

	<ul style="list-style-type: none"> Universal set and null set. Describe set operations Define relations. Explain equivalence relation. Explain composite relation 	<p>super set, Universal set and null set.</p> <p>Develop simple programs to carry out the operation.</p>	with relevant software.	operations	Guide the students on how to develop simple programs to carry out set operations.	loaded with relevant software.
5	<p>Be able to:</p> <ul style="list-style-type: none"> Define string Explain representation: character, string length and string values. Carry out basic operation on string assignment, sub string selection, insertion, sub string retrieval. Deletion concatenation and replacement. Carry out set representation. Describe storage mapping techniques for string variables. 	<p>Discuss String and its basic operations</p> <p>Set representation</p> <p>Storage mapping techniques for string variables.</p>	Ditto	Be able to solve problems requiring the application of sting length, assignment, selection, insertion	Introduce some problems and solve them with the students	Ditto
Week/s	General Objective 5: Know the properties of ordered and linear list					
Week	Specific Learning Outcomes	Teacher's activities	Resources	Specific Learning Outcomes	Teacher's activities	Resources
	<p>Be able to:</p> <ul style="list-style-type: none"> Explain the term occupancy leans, empty, loose. 	<p>Explain Different life cycle of data</p>	Ditto	Be able to use variable fixed length	Demonstrate concept of fixed and variable	Ditto

6	<ul style="list-style-type: none"> Distinguish and define birth, death and change of data. Define a sequential list, Explain the differences between fixed and variable length fields. Implement fixed and variable fields. 	<p>Discuss sequential list</p> <p>Record length outlining the fixed and variable length.</p>		record	length using appropriate examples.	
7	<p>Be able to:</p> <ul style="list-style-type: none"> Define ordered and linear list. Explain operations that can be performed on an ordered list: append, search (including delete, sort, selection and exchange, merge, including multiway merge and balance merge.) 	<p>Define ordered and linear list.</p> <p>Discuss various operations that can be performed on ordered list.</p>	Ditto	<p>Be able to</p> <p>Carry out ordered list operations</p>	<p>Demonstrate using appropriate examples concept of ordered and linear lists.</p> <p>Demonstrate how to perform ordered list operations</p>	Ditto
Week/s	General Objective 6: Know simple Linear Data Structure					
8	<p>Be able to:</p> <ul style="list-style-type: none"> Describe different types of linked list, array, double linked list, queues, stack. Explain the use of pointers. Describe storage mapping for linked lists. 	<p>Define linked list and compare it with linear list.</p> <p>Explain types of linked list.</p> <p>Discuss different types of trees.</p>	Ditto	<p>Be able to</p> <p>apply linked list.</p>	<p>Demonstrate the push and pop operation possibly with diagram.</p> <p>Carry out operations on linked lists e.g push and pop on</p>	Ditto

		Discuss the use of pointers			stacks and all operations on over list	
	General Objective 7: Know non – linear structures					
Week	Specific Learning Outcomes	Teacher’s activities	Resources	Specific Learning Outcomes	Teacher’s activities	Resources
9	<p>Be able to:</p> <ul style="list-style-type: none"> • Define a tree • State properties of tree • Describe different types of tree. (General tree, binary tree) • Explain binary tree representation. <p>Define graph, its types and properties</p>	<p>Discuss tree and its properties</p> <p>Explain binary tree representation</p> <p>Define graph State properties of graph: routes, queued and non-directed</p> <p>Describe different types of graphs: circle, loops, etc.</p> <p>Describe operations such as proceeds, less than etc.</p>	Ditto	<p>Be able to write simple program to implement trees</p> <p>write simple program to implement graphs</p>	<p>Demonstrate how to write simple program to illustrate trees</p> <p>Demonstrate how to write simple program to illustrate graphs</p>	Ditto
Week/s	General Objective 8: Understand different sorting and searching techniques					
	Be able to	Be able to: Define sorting	Ditto	Be able to implement different	Guide students on how to write programs to	Ditto

10	<p>Be able to:</p> <p>Define sorting Explain the various sorting techniques</p>	<p>Explain Comparison based sorting</p> <p>Explain bubble sorting algorithm</p> <p>Explain selection sorting algorithm</p>		<p>sorting techniques in program</p>	<p>implement different sorting techniques</p> <p>Guide students on how to Perform different sorting and searching techniques</p> <p>Apply sorting algorithm to sort an array of objects.</p>	
11	<p>Be able to :</p> <p>Define sorting Explain the various sorting techniques</p>	<p>Be able to:</p> <p>Define sorting</p> <p>Explain selection sorting algorithm</p> <p>Explain insertion sorting algorithm</p> <p>Explain linear and binary search algorithm</p>	Ditto	<p>Be able to implement different sorting techniques in program</p>	<p>Guide students on how to write programs to implement different sorting techniques</p> <p>Guide students on how to Perform different sorting and searching techniques</p> <p>Apply sorting algorithm to sort</p>	Ditto

					an array of objects.	
General Objective 9: Understand Sorting Techniques						
Week/s	Specific Learning Outcomes	Teacher's activities	Resources	Specific Learning Outcomes	Teacher's activities	Resources
12	To be able to: Describe Elementary Sorting Methods : - Bubble Sort - Selection Sort - Insertion Sort - Quick sort	To explain Elementary Sorting Methods : - Bubble Sort - Selection Sort - Quick sort - Insertion sort	Capability to project and demonstrate Elementary Sorting Methods. A comprehensive workbook of data structures fundamentals with some specific programming language such as C, C++, or Java.	The ability to write a simple program that includes and covers Elementary Sorting Methods : - Quick sort	Supervise the laboratory and support students in writing a simple programs that includes and covers Elementary Sorting Methods : - Quick sort - Bubble Sort - Selection Sort - Insertion Sort	Personal Computer loaded with Programming Language complier such as C++ or Java complier
13	To be able to: Describe Elementary Sorting Methods :	To explain Elementary Sorting Methods : - Radix Sorting	Capability to project and	The ability to write a simple	Supervise the laboratory and support students	Personal Computer loaded

	<ul style="list-style-type: none"> - Radix Sorting - Priority Queues - Merge sort - External Sorting - 	<ul style="list-style-type: none"> - Priority Queues - Merge sort - External Sorting 	demonstrate Elementary Sorting Methods. A comprehensive workbook of data structure fundamentals with some specific programming language such as C, C++, or Java.	program that includes and covers Elementary Sorting Methods : <ul style="list-style-type: none"> - Radix Sorting - Priority Queues - Merge sort - External Sorting 	in writing a simple programs that includes and covers Elementary Sorting Methods : <ul style="list-style-type: none"> - Radix Sorting - Priority Queues - Merge sort - External Sorting 	with Programming Language complier such as C++ or Java complier
General Objective 10: Understand Searching Techniques						
Week/s	Specific Learning Outcomes	Teacher's activities	Resources	Specific Learning Outcomes	Teacher's activities	Resources
14	To be able to: Describe Elementary Searching Methods : <ul style="list-style-type: none"> - Binary Search - Sequential Search - Balanced Trees 	To explain Elementary Searching Methods : <ul style="list-style-type: none"> - Balanced Trees 	Capability to project and demonstrate Elementary Searching Methods. A comprehensive	The ability to write a simple program that includes and covers Elementary Searching Methods : <ul style="list-style-type: none"> - Balanced Trees - Sequential Search - Balanced Trees 	Supervise the laboratory and support students in writing a simple programs that includes and covers Elementary Searching	Personal Computer loaded with Programming Language complier such as

			ive workbook of data structure fundamental s with some specific programmin g language such as C, C++, or Java.		Methods : - Balanced Trees - Sequential Search - Balanced Trees	C++ or Java complier
15	To be able to: Describe Elementary Searching Methods : - Hashing - Radix Searching - External Searching	To explain Elementary Searching Methods : - Hashing - Radix Searching - External Searching	Capability to project and demonstrate Elementary Searching Methods. A comprehens ive workbook of data structure fundamental s with some specific programmin g language such as C,	The ability to write a simple program that includes and covers Elementary Searching Methods : - Hashing - Radix Searching - External Searching	Supervise the laboratory and support students in writing a simple programs that includes and covers Elementary Searching Methods : - Hashing - Radix Searching - External Searching	Personal Computer loaded with Programm ing Language complier such as C++ or Java complier

			C++, or Java.			
--	--	--	------------------	--	--	--

Assessment: Give details of assignments to be used:
Coursework/Assignments 20%; Course Test 10%; Practical 40%; Examination 30%