



University of Nicosia, Cyprus

Course Code COMP-211	Course Title Data Structures	ECTS Credits 6
Department Computer Science	Semester Fall, Spring	Prerequisites COMP-113, MATH-101
Type of Course Required	Field Computer Science	Language of Instruction English
Level of Course 1 st Cycle	Year of Study 2 nd	Lecturer(s) Dr Andreas Savva
Mode of Delivery Face-to-face	Work Placement N/A	Co-requisites None

Objectives of the Course:

The main objectives of the course are to:

- Introduce students to Abstract Data Types (ADT).
- Provide practical experience to advanced programming techniques and data structures including tables, linked lists, queues and stacks.
- Introduce students to advance recursion such as the divide-and-conquer and backtracking.
- Obtain a foundation that will allow students to use storage media; methods of representing structured data; and techniques for operating on data structures.
- Introduce students to searching and sorting algorithms.
- Introduce students to Binary Trees and graphs.

Learning Outcomes:

Upon completion of the course students should be able to:

- Discuss the use of primitive data types and build-in data structures.
- Describe common applications for different data structures.
- Implement user-defined data structures in a high-level language.
- Compare alternative implementations of data structures with respect to performance.
- Recognize when and how to use the following data structures: Arrays, Linked lists, Stacks, Queues and Binary trees.
- Compare and contrast the costs and benefits of dynamic and static data structure implementation.
- Choose the appropriate data structure for modeling a given problem.
- Describe the concept of recursion and give examples of its use.
- Describe the divide-and-conquer and backtracking approaches.
- Compare iterative and recursive solutions and determine when a recursive solution is appropriate for a problem.
- Apply various sorting and searching algorithms.

Course Contents:

1. Programming Principles

- Programming style
 - Coding, testing and further refinement
 - Program maintenance
 - Abstract Data Types (ADT)
2. Stacks - Arrays
 - Stack specifications
 - Implementation of Stacks
 - Application of Stacks
 - The Standard Template Library (STL)
 3. Queues – Arrays
 - Specification of Queues
 - Implementation of Queues
 - Circular Queues
 - Application of Queues
 4. Linked Stacks and Queues
 - Pointers
 - Linked Stacks
 - Safeguards
 - Linked queues
 5. Recursion
 - Principles of recursion
 - Tree of subprogram calls
 - Divide-and-Conquer
 - Backtracking
 - Tree structure programs: Look-Ahead in Games
 6. Linked Lists
 - List Definition
 - Implementation of Lists
 - Double Linked lists
 - Circular Linked lists
 7. Binary trees
 - Definition of Binary Trees
 - Traversal of Binary Trees
 - Linked implementation of Binary Trees
 - Binary Search Trees – Insertion, Removal, Treesort
 8. Sorting
 - Insertion sort
 - Bubble sort
 - Quick sort
 9. Searching
 - Sequential search
 - Binary search
 - Comparison Trees
 10. Introduction to Graphs

Learning Activities and Teaching Methods:

Lectures, In-class Exercises

Assessment Methods:

Homework, Assignments, Mid-Term, Final Exam.
--

Required Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Nell Dale	C++ Plus Data Structures, 4 th Edition	Jones and Barlett	2006	0-7637-4158-2

Recommended Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Larry Nyhoff	ADTs, Data Structures and Problem Solving with C++, 2nd Ed.	Prentice Hall	2004	0-13-140909-3
Robert L. Kruse, Alexander J. Ryba	Data Structures and Program Design in C++	Prentice Hall	1999	0-13-768995-0