Introduction to Course

Data Structures – CS2001 – Fall 2021

Dr. Syed Ali Raza
School of Computer Science
National University of Computing & Emerging Sciences
Karachi Campus

Agenda

- Course Description
- Course Objective
- Learning Outcomes
- Grading Scheme
- Textbook and Reference books
- Assignments
- Quizzes
- Exams
- Class Participation
- Class Project
- Weekly Class and Lab Plan
- Contact Information

Course Description

- Data Structures is a core course in Computer Science curriculum.
- It is an essential building block for solving applied problems with computers.
- The course will introduce the fundamentals of data structures and will provide thorough understanding of how to systematically organize data inside a computer system.

Course Description

- The course discusses basic memory management for efficiently solving problems on both time and space requirements.

- A variety of data structures will be discussed theoretically, their efficient implementations and application cases will also be discussed.

- The student will learn abstraction, encapsulation and structures for efficiently processing information in a variety of scenarios.

Course Objectives

- To understand the design of fundamental data structures and algorithms for problem solving through computer system.
- To study the tradeoff choices in the design and implementation of data structures.
- To provide a rigorous “hands-on” experience with implementing different data structures in a high-level programming language.
- To analyze time/space tradeoff for different solutions to the same problem.

Learning Outcomes

- Student will be able to learn and understand basic/advanced data structures
- Student will be able to perform analysis of data structures choices for any real-world application.
- Student will learn the tradeoff with different choices of data structures
- Student will be able to write computer solutions to efficiently store, retrieve, manipulate and update the data stored inside computers.

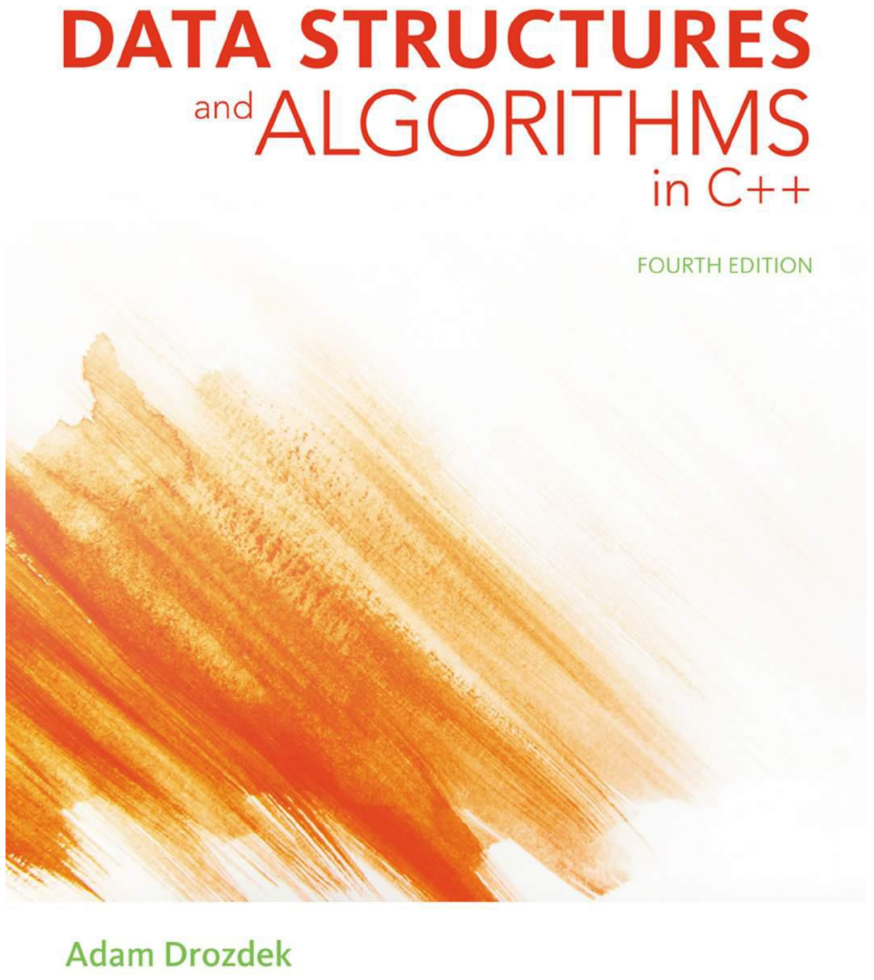
Grading Scheme

- Programming Assignments 15%
- Quizzes 05%
- Midterm Exam 20%
- Class Project 10%
- Final Exam 50%

Text Book

Leading Text in CS

- Strengthen your understanding of data structures and their algorithms for the foundation you need to successfully design, implement and maintain virtually any software system


DATA STRUCTURES and ALGORITHMS in C++

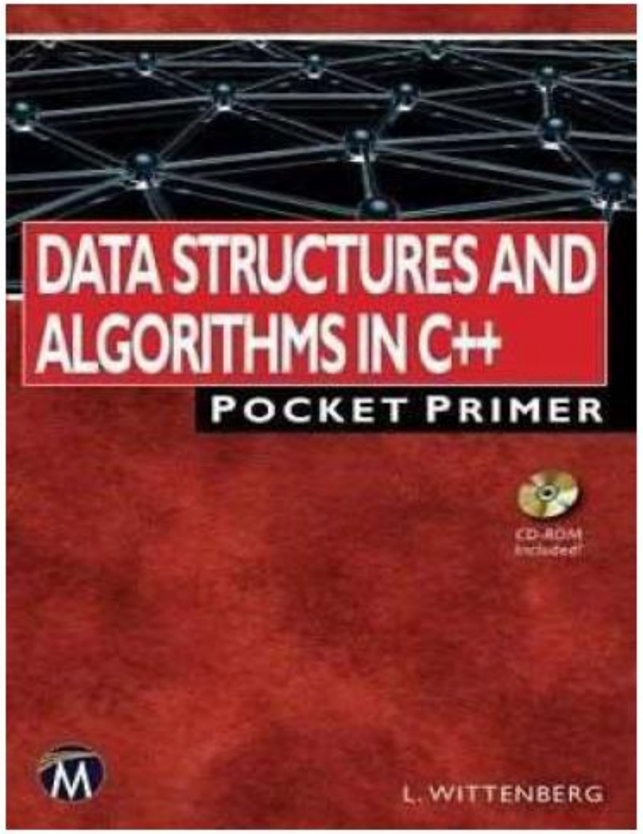
FOURTH EDITION

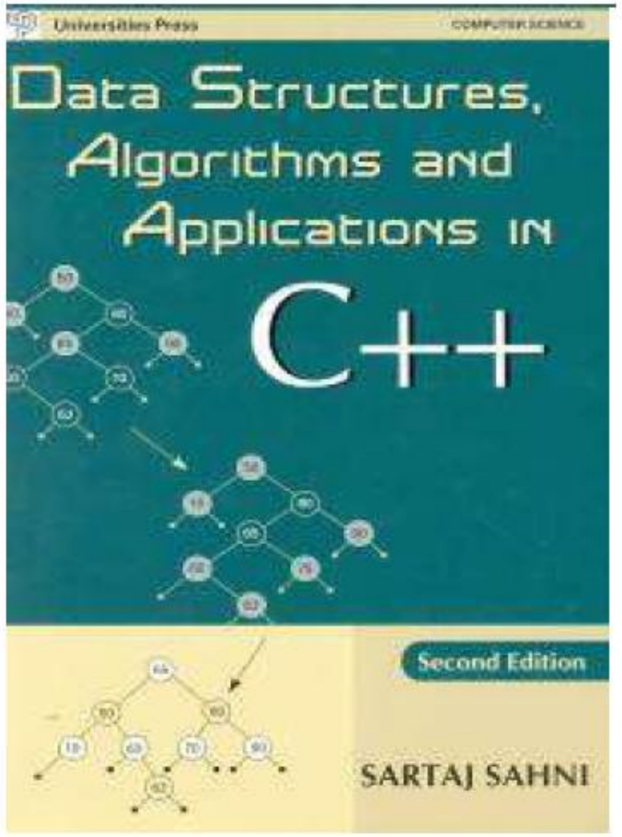
Adam Drozdek

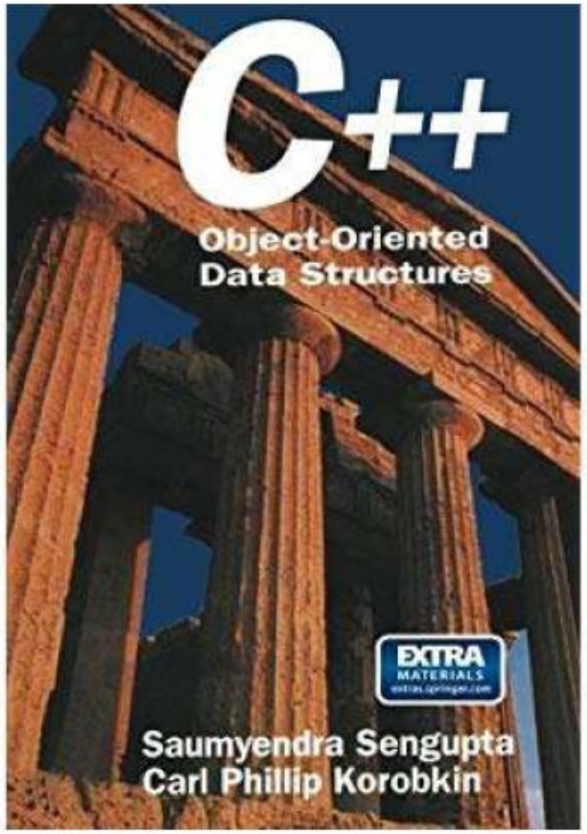
Reference Books

- Data Structure and Algorithms Using C++ -- A Practical Implementation by Sachi Nandan Mohanty and Pabitra Kumar Tripathy
- Data Structures Using C++ by VARSHA H. PATIL Oxford University Press
- Data Structures and Algorithm Analysis by Clifford A. Shaffer

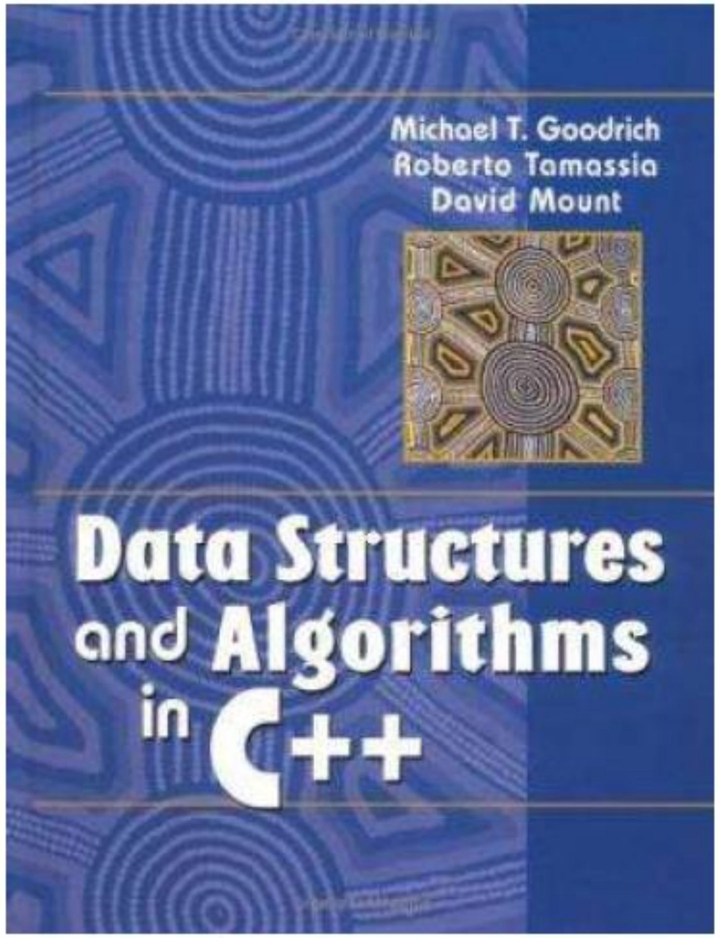
More Reference Books

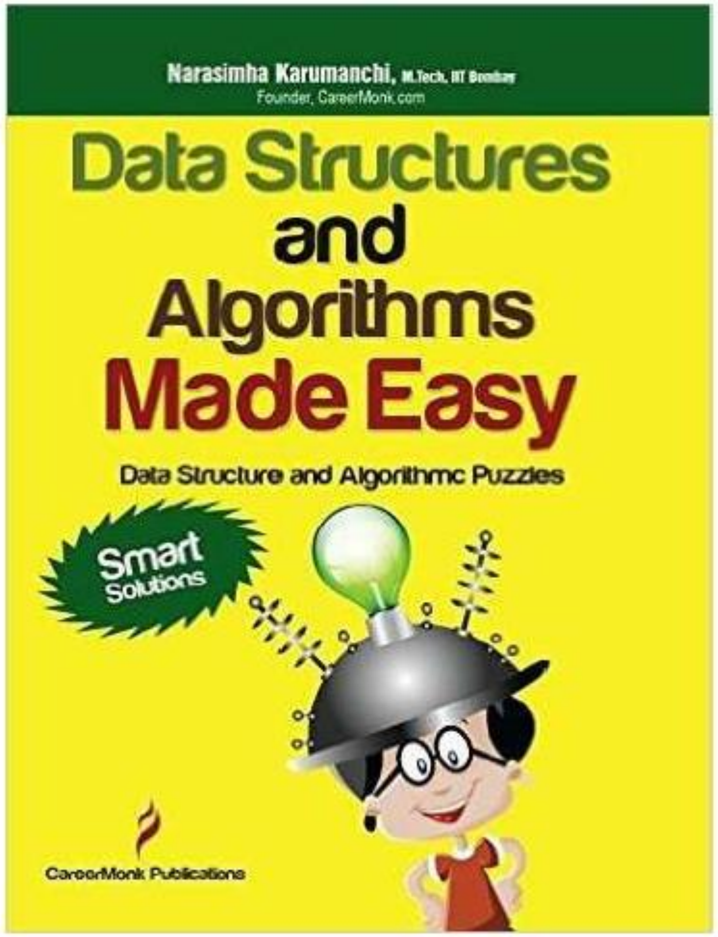


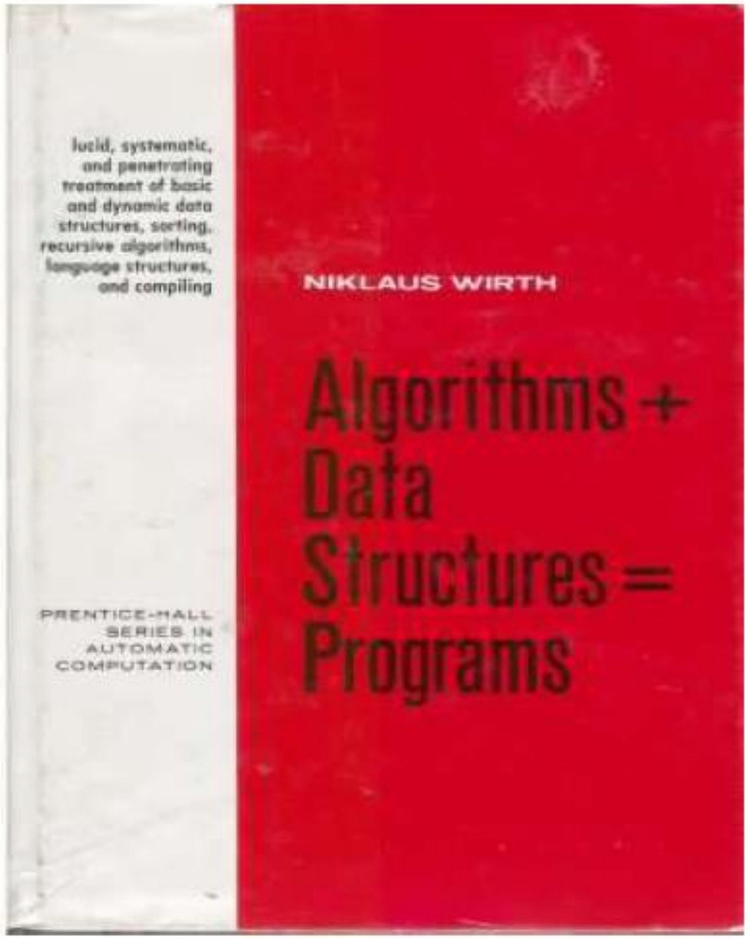




Some More Reference Books




Narasimha Karumanchi, M.Tech, IIT Bombay
Founder, CareerMonk.com
Data Structures and Algorithms Made Easy
Data Structure and Algorithmic Puzzles
Smart Solutions
CareerMonk Publications


lucid, systematic, and penetrating treatment of basic and dynamic data structures, sorting, recursive algorithms, language structures, and compiling
PRENTICE-HALL SERIES IN AUTOMATIC COMPUTATION
NIKLAUS WIRTH
Algorithms + Data Structures = Programs

Programming Assignments

- There will be 3 programming assignments, each assignment may contains 2-4 problems.
- These programming assignments will be on provided with sample input/output test cases.
- The assignments are for individual, plagiarism will not be tolerated.
- You need to access the platform with your nu mail IDs.
- There may be more hidden test-cases for the problems.

Programming Assignments

Deadline and Penalty

- Deadline is mentioned for each Programming Assignment (Usually 2 weeks time)
- After the deadline – there will be a penalty of 25% if the assignment submitted after the deadline, then, for each 48 hours it is reduced 25% more.
- Marks are only given if the submission passed plagiarism check and manual check.
- Plagiarism cases may be awarded an F grade.

Quizzes

- There will be 4-5 quizzes – all surprise quizzes
- Best n-1 will be counted
- Weightage 05 %

Midterm Exam

- Two midterm exams – one hour each as per policy.
- Weight 10 % each

Final Exam

- There will be a 3 hours exam as per policy.
- Weightage 50%

Class Participation

- Very important instrument for creating an impact.
- I appreciate questioning…
- There will be no marks for it but make sure everyone participate

Class Project

- There will be a class project -You can have 2- 3 members- members allow within sections.
- Weightage 10%
- The “Theme” for CS201- Data Structures class project is “Data Structures for Large Datasets”
- Class project call and schedule will be announced

Class Project

Call for class project

- You need to submit a proposal mentioning team members name, project title and a brief description about the project. (1 mark for each submitted proposal)
- Class Project Demos – 2 marks
- Code review – 2 marks
- Idea and completion – 4 marks
- Project report – 1 mark

Weekly Plan

Session	Topics	Chapters
1	Course Overview, Introduction to Course & Conduct, Grading Scheme, Text Book, Quizzes, Assignments	
2	C++ Language Specification & OOP	Chapter 1
3	C++ Language Specification & OOP	
4	C++ Dynamic Memory Management	
5	Arrays (1D) – Dynamic Safe Arrays	
6	Arrays (2D)	
7	Different Type of Arrays	
8	Recursion - 1	Chapter 5
9	Recursion - 2	
10	Recursion - 3	
11	Node and List (Singly Linked List)	Chapter 3
12	Some utility functions of List (Singly Linked List)	
13	List (Circular Linked List)	
14	List (Doubly Linked List)	
15	Elementary Sorting Techniques -1	Chapter 9
16	**Midterm I – Exam**	

Weekly Plan

16	**Midterm I – Exam**	
17	Elementary Sorting Techniques -2	Chapter 9
18	Advanced Sorting	
19	Searching	
20	Stack -1	Chapter 4
21	Stack – 2	
22	Stack – application	
23	Queues + Priority Queues	
24	Queue Application	
25	Heap	
26	Trees – BT, BST, MWT– 1	Chapter 6-7
27	Trees – BT, BST, MWT– 2	
28	Trees – BT, BST, MWT– 3	
29	Trees – BT, BST, MWT– 4	
30	Hashing - 1	Chapter 10
31	Hashing - 2	
32	Graphs - 1	Chapter 8
33	Graphs Traversals- 1	
34	Graphs Traversals- 2	
35	Weighted Graphs	
36	Graph Algorithms	
37	Revision & Application	
38	Revision & Application	
39	Revision & Application	

Lab Weekly Plan

Lab sessions	Lab Content
LAB 1	**IDE (Planning a solution to a problem, coding and debugging)** Discuss 2-3 problems and their complete solutions. Along with the debugging and testing several test cases to these problems.
LAB 2	**Implementing a DynamicSafeArray with one/two dimensional pointers**. This lab covers how dynamic memory is used to implement 1D and 2D arrays which are more powerful as compared to default array mechanism of the programming language C/C++ supports. .
LAB 3	**Implementing Recursion, solving a simple maze**. Implement and test different types of recursion, call order, back tracking, and task completion. Use a simple single path maze problem to implement a complete solution.
LAB 4	**Elementary sorting** Implement and test elementary sorting algorithms.
LAB 5	**Linked List (Singly linked list)** Implement a SinglyLinkedList class with handful of helper functions.
LAB 6	**Linked List (Doubly and circular linked list)** Implement a DoublyLinkedList and CirculerLinkedList class with handful of helper functions.

Lab Weekly Plan

LAB 7	**Stack and Queue (using both Arrays and Linked List)** Implement both Stack and Queue with underlying arrays and Linkedlist based implementation. Wrapper based implementation.
LAB 8	**Midterm Exam**
LAB 9	**Binary Search tree** Given code of BST and asked to understand the insertion in binary search tree and implement some recursive functions for BST
LAB 10	**Huffman Coding:** Application of BST
LAB 11	**AVL Tree:** Implementing an AVL Tree with based helper function. Compare it implementation with BST
LAB 12	**Hashing:** Implementing basic Hashing mechanism
LAB 13	**Graph Data Structures** Develop coding routine for holding graphs inside memory into its basic representation. Some helper functions on graphs.
LAB 14	**Applications of Graph Data Structures** Develop coding routine for application of graphs in problem solving. Shortest Path, Reachability, MST, All Pair Shortest Paths, Topological sort etc.
LAB 15	**Lab Final Exam**

Consultancy Hours

- Consultancy Hours:
  - Monday: 1-3 PM in my office – prior appointment required
  - Friday: 9-11 AM in my office – prior appointment required

- Contact me
  - email: syed<dot>aliraza@nu<dot>edu<dot>pk
  - Office: 111-128-128 ext: 164

Introduction to Data Structures

Agenda

- What is Data Structures?
- Abstract Data Types?
- Some example data structures
- Operations performed on data structures
- Data, information, knowledge, and wisdom

What is Data Structures?

- A data structure is a data organization, management, and storage format that enables efficient access and modification
- In simple words, a data structure is an arrangement of data in a computer's memory
- DS is an implementation that we provide, for example, how the data will be stored and accessed
- Using DS the same data can be organized in a variety of ways so that it fills our objectives

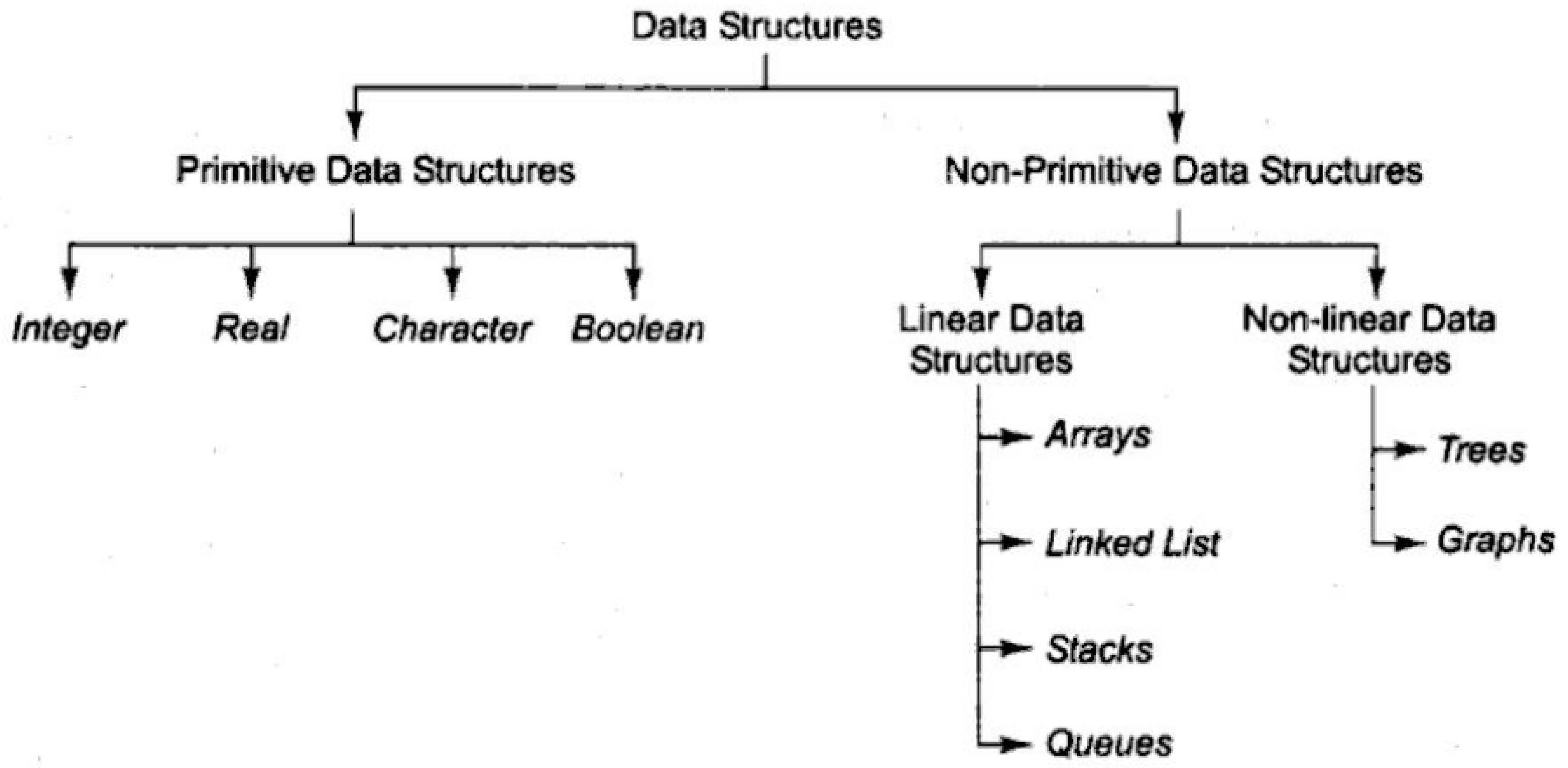
Why Data Structures?

- We use data structures to write *efficient* code: solve the problem within its resource constraints which are,
  - Time
  - Space
- To build a program that needs to deal with lots of data, for example a web app used by thousands of people simultaneously, the data structures you select may affect whether your software runs at all, or simply conks out because it can’t handle the load.
- Data Structure allows efficient data search and retrieval.
- It allows to manage large amount of data such as large databases and indexing services such as hash table.

How to Use Data Structures?

- Each data structure has its own pros and cons.
- No data structure is ideal for all problems.
- Specific Data structures are decided to work for specific problems.
- The use of data structures requires:
  - Memory to store data
  - Time to perform operation
  - Programming effort

Classification of Data Structures



Abstract Data Structure (ADT)

- Data structure can be seen in two ways,
    - Logical/Abstract view (Protocols or rules to follow)
    - and implementation view.
- ADT is a blueprint which tells how this data structure should behave.
- It tells how data can be saved, accessed, and manipulated.
- In the implementation part, the rules given by ADT are followed to implement the data structure using some programming language.
- Formally, an ADT is basically a logical description or a specification of components of the data and the operations that are allowed, that is independent of the implementation.

Abstract Data Structure (ADT)

- ADT is a theoretical concept which is used in the design and analysis of data structures, algorithms and software programs.
- An ADT can be implemented in many different ways, even using the same programming language. But these implementations must follow ADT descriptions.

Some Examples of Data Structures

- **Stack:** Stack is a linear list in which insertion and deletions are allowed only at one end, called **top**.
- A stack is an abstract data type (ADT), can be implemented in most of the programming languages. It is named as stack because it behaves like a real-world stack, for example: - piles of plates or deck of cards etc.
- **Queue:** Queue is a linear list in which elements can be inserted only at one end called **rear** and deleted only at the other end called **front**.
- It is an abstract data structure, similar to stack. Queue is opened at both end therefore it follows First-In-First-Out (FIFO) methodology for storing the data items.

Operations on Data Structures

- **Traversing:** visiting each element of a data structure to perform some specific operation like searching or sorting.
- **Insertion:** adding the element(s) to the data structure at any location.
- **Deletion:** removing an element at any random location from the data structure.
- **Searching:** finding the location of an element within the data structure.
- **Sorting:** arranging the data structure in a specific order.
- **Merging:** two lists List A and List B of size M and N respectively, of similar type of elements, clubbed or joined to produce the third list, List C of size (M+N).

Some Applications of DS

- Circular linked list is used in a slide show where a user wants to go back to the first slide after last slide is displayed.
- Stack is used in matching of parenthesis, string reversal, UNDO and REDO functions in an editor.
- Trees are used to implement the hierarchical structures like directory and file system.

Data ->information -> knowledge -> wisdom

DATA

- Data is a collection of facts in a raw or unorganized form such as numbers or characters.
- Data examples:
  - 12012021
  - Yes, Yes, No, Yes, No, Yes, No, Yes
  - 42, 63, 96, 74, 56, 86
- Without context, data can mean little.

Data ->information -> knowledge -> wisdom

INFORMATION

- Using the context we can process data and transform it into information
- For example, *12012021* is just a sequence of numbers without apparent importance. But in the context of ‘date’, we can recognize that it represents 12 January 2021.
- Information is the data which is easier to measure, visualize and analyze for a specific purpose.

Data ->information -> knowledge -> wisdom

INFORMATION

- Data examples:

Data	Context	Information
Yes, Yes, No, Yes, Yes, Yes, No, Yes	: Responses to a survey question: would you but product x at price y?	Yes=6, No=2
42, 63, 96, 74, 56, 86	Marks in different subjects	Average marks is 69.5

Data ->information -> knowledge -> wisdom

KNOWLEDGE

- “How” are the pieces of information connected to other pieces to add more meaning and value?
- Uncover relationships that are not explicitly stated as information.

Data	Context	Information	Knowledge
Yes, Yes, No, Yes, Yes, Yes, No, Yes	: Responses to a survey question: would you but product x at price y?	Yes=6, No=2	More people voted ‘Yes’ than ‘No’
42, 63, 96, 74, 56, 86	Marks in different subjects	Average mark is 69.5	Average mark is less the passing mark

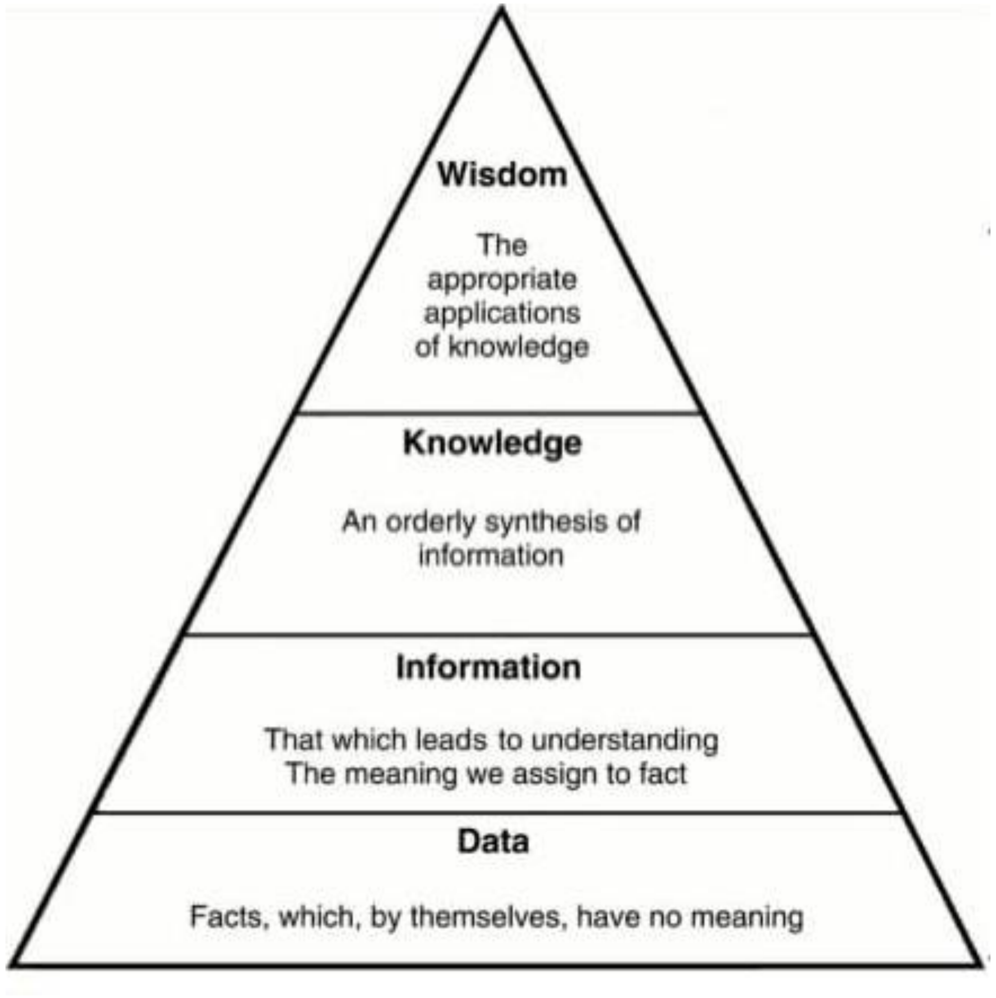
Data ->information -> knowledge -> wisdom

WISDOM

- Wisdom is knowledge applied in action
- Use the knowledge gained from the information to take proactive decisions

Data	Context	Information	Knowledge	Wisdom
Yes, Yes, No, Yes, Yes, Yes, No, Yes	: Responses to a survey question: would you but product x at price y?	Yes=6, No=2	More people voted 'Yes' than 'No'	Increase price of the product
42, 63, 96, 74, 56, 86	Marks in different subjects	Average mark is 69.5	Average mark is less the passing mark	Increase study time

Data, Information, Knowledge, Wisdom – DIKW Pyramid


Wisdom
The appropriate applications of knowledge
Knowledge
An orderly synthesis of information
Information
That which leads to understanding
The meaning we assign to fact
Data
Facts, which, by themselves, have no meaning