DATA STRUCTURE

Dr. Rania Baashirah

CHAPTER 1: INTRODUTION

What is Data?

- $x^2 + 2y - 2 = 1$
- x, y variables represent data

Data Types

- Value type name used in computer programs
- Set of data with predefined values. Ex. Int, float, char, string
- Computer knows only 0,1

Why do we use data type?

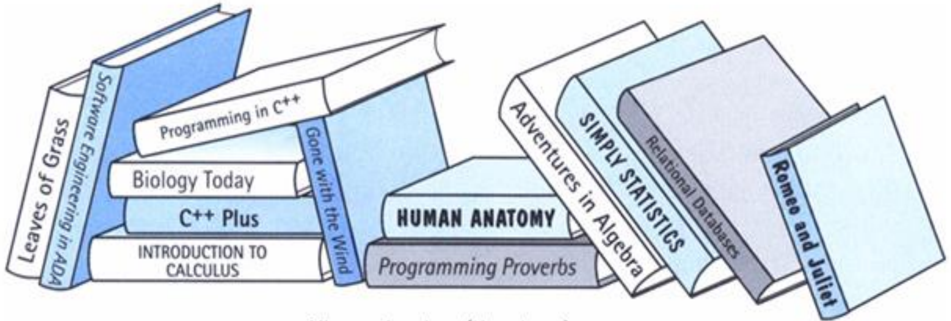
- Reduce coding efforts, errors, and efficient

Data types categories

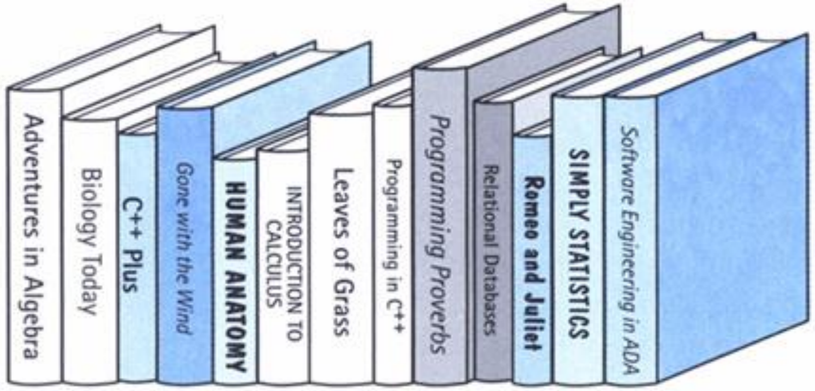
- Primitives data types (Pre-defined). Ex. Int, char, float, double
- User defined. Ex. Classes, objects

What is Data Structures?

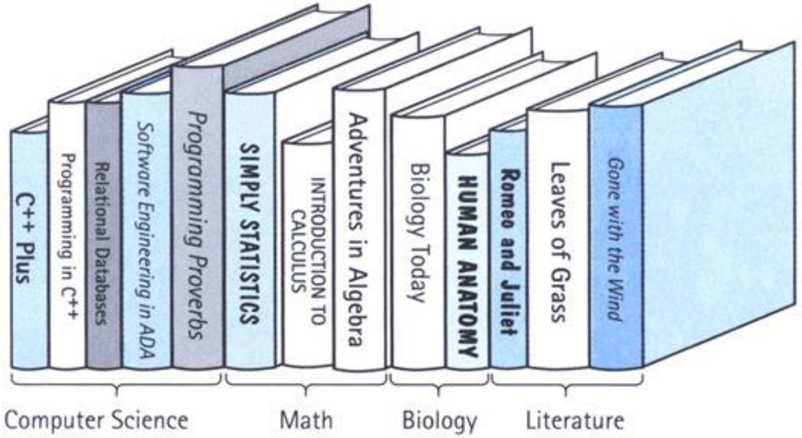
- Example: Library
  - is composed of elements (books)
  - Accessing a particular book requires knowledge of the arrangement of the books
  - Users access books only through the librarian



All over the place (Unordered)



Alphebetical order by title



Ordered by subject

What is Data Structure

- How we use data types and variables in our problems; store, organize and call data.
- A data structure is defined by :
  - The logical arrangement of data elements, combined with the set of operations we need to access the elements.

Format of organizing and storing data

- Arrays, files, linked list, queue, graphs
- Linear data structure, Non linear data structure

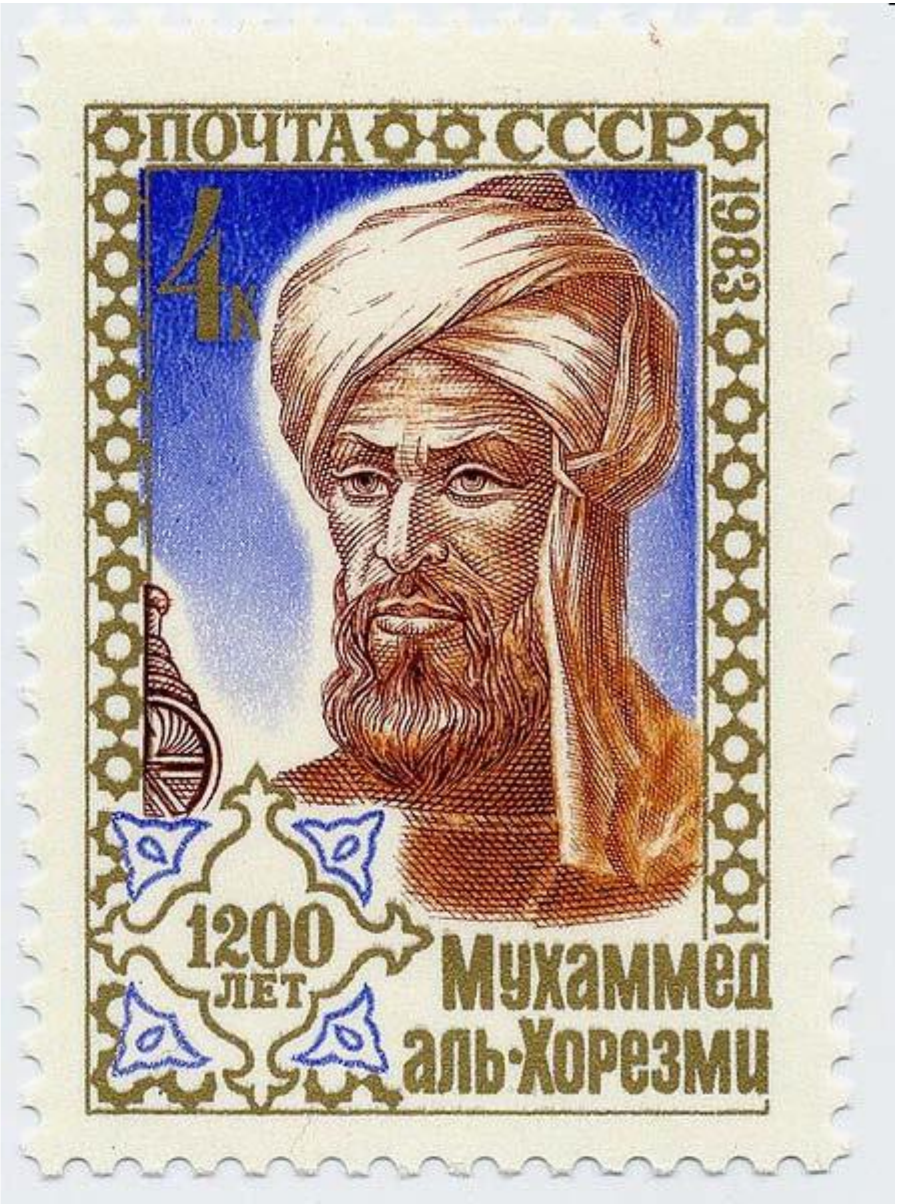
Abstract Data Structure (ADTs)

- By default primitive data types support basic operation +,-
- User defined needs operation

- A data structure is defined by :
  - The logical arrangement of data elements, combined with the set of operations we need to access the elements.

Algorithms – Origin of Name

- The word is derived from:
  - أبو عبد الله محمد بن موسى الخوارزمي
  - Muhammad Ibn Mūsā Al-Khwārizmī
  - Al-Khwārizmī → Algorithm
- Algorism refers to rules of performing arithmetic using Arabic numerals.
- Algorithm means a well-defined procedure for solving a problem.



Algorithms Example

- Algorithms are used in our everyday life.
- They are descriptions of procedures we perform to solve problems.
  - Making a cake using a recipe.
    - Problem: make a cake.
    - Procedure: Add egg, sugar, flour, mix, cook, refrigerate, Eat
  - Buying from supermarket.
    - Problem: get items you want.
    - Procedure: visit shop, look for good and cheap stuff, scan for offers, check expiry dates, etc.
- Computer algorithms are more formal than every day's life algorithms.

What is Algorithm ?

- There is no formal definition of what an algorithm is.
- However, an algorithm is characterized by:
  - **Definiteness:** Each step of the algorithm must be precisely defined (no ambiguity).
  - **Finiteness:** The algorithm must terminate after a finite number of steps.
  - **Effectiveness:** The algorithm must solve the problem within manageable time.

Algorithm Example

Algorithm: Greatest Common Divisor (GCD) of positive integers A and B.

```
If A < B swap A and B
while(B != 0)
    R = A mod B
    A = B
    B = R
End while
Print A
```

"Mod" is the reminder of A/B
(ex. 14 mod 3 = 2)
In python:

```
>>> 14%3
2
```

iteration	A	B	R
	~~15~~	~~24~~	
1st	~~24~~	~~15~~	~~9~~
2nd	~~15~~	~~9~~	~~6~~
3rd	~~9~~	~~6~~	~~3~~
4th	~~6~~	~~3~~	~~0~~
5th	3	0	

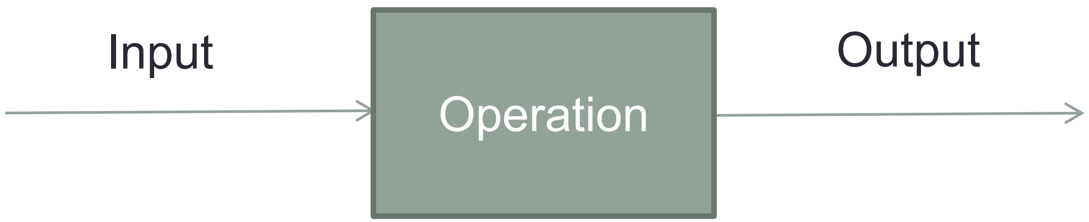
- Algorithm 4: Criteria Satisfaction
  - Definiteness: Yes
  - Finiteness: Yes
  - Effectiveness: Yes

Properties of Algorithms

- In the previous slide :
- The Algorithm is definite. Why?
- The Algorithm is finite. Why?
  - How long does it take to find the GCD of two numbers in the worst-case?
- The Algorithm is effective. Why?
  - do some Google search about the "greatest common divisor" algorithm by Euclid.

Properties of Algorithms

- ***Input***: An algorithm has a specified set of input values.
- ***Output***: The algorithm produces a specified set of the output values.
  - The output values are the solution.
- ***Correctness***: An algorithm should produce the correct output values **for each set of input values**.
- ***Finiteness***: get output after a finite number of steps for any input.
- ***Effectiveness***: Perform the algorithm correctly in a finite amount of time.
- ***Generality***: The algorithm should work for all problems of the desired form.



Input

Operation

Output

Exercises

- Determine which characteristics of an algorithm the following procedures have and which they lack.

```
1- function double(n: positive integer)
        while n > 0
                n := 2n
```

```
2- function divide(n: positive integer)
        while n ≥ 0
                m := 1/n
                n := n − 1
```

```
3- function sum(n: positive int)
        sum := 0
        while i < 10
                sum := sum + I
```

```
4- function choose(a, b: integers)
        x := either a or b
```