NATIONAL UNIVERSITY OF TECHNOLOGY (NUTECH)

DR. SAMAN RIAZ LECTURE # 2

SOME BASIC QUESTIONS

- What is an integer?
 - What are the operations that can be performed on Integers?
- What is a character?
 - What are the operations that can be performed on characters?



SOME MORE BASIC QUESTIONS

- How is an integer represented?
 - How are the integer-based operations performed?
- How is a rational number represented?
 - How are different operations performed?

ABSTRACTION LEVELS

- What are different kinds of programming Languages?
 - Low-level/Assembly/High-level languages
- Programming in a certain type of language involves abstraction
 - In order to simplify the representation
 - In order to make it easier to program

ABSTRACTION LEVELS

• While programming in a high-level language we are not that bothered about low-level representation

Abstract data-types allow us to take our abstraction level, one-step further

 Instead of focusing on how a particular data-structure can be implemented, focus is on what should be chosen to solve the problem

- Abstract Data Type (ADT): a definition for a data type solely in terms of a set of values and a set of operations on that data type.
- Each ADT operation is defined by its inputs and outputs.
 - Hide Implementation details (Encapsulation!)

Def: A collection of related data items together with an associated set of operations

- e.g. whole numbers (integers)

 Basic Operations: apply arithmetic operations (addition, subtraction, multiplication and division)
- e.g. Flight reservation

 Basic Operations: find empty seat, reserve a seat,

 cancel a seat assignment

Why "abstract"?

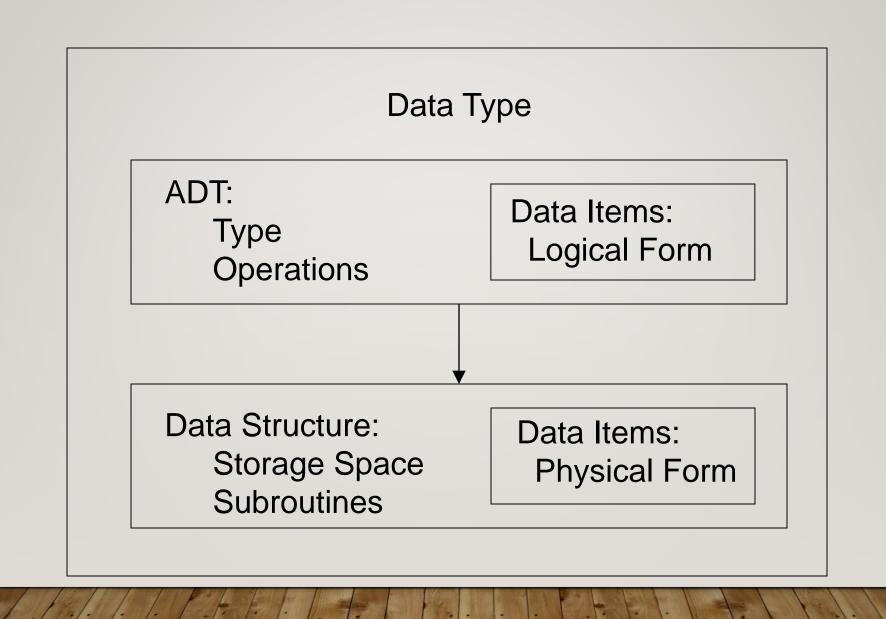
Data, operations, and relations are studied independent of implementation.

What not how is the focus.

DATA STRUCTURE

- A <u>data structure</u> is the physical implementation of an ADT.
 - Each operation associated with the ADT is implemented by one or more subroutines in the implementation.
- Data structure usually refers to an organization of data in main memory.

 <u>File structure</u> is an organization for data on peripheral storage, such as a disk drive.



DATA STRUCTURES, ABSTRACT DATA TYPES, AND IMPLEMENTATIONS

- Consider example of an airplane flight with 10 seats to be assigned
- Tasks
 - List available seats
 - Reserve a seat
- How to store, access data?

DATA STRUCTURES, ABSTRACT DATA TYPES, AND IMPLEMENTATIONS

- Consider example of an airplane flight with 10 seats to be assigned
- Tasks
 - List available seats
 - Reserve a seat
- How to store, access data?
 - 10 individual variables

USE 10 INDIVIDUAL VARIABLES

Algorithm to List available seats

Algorithm to Reserve a seat

```
I. Set DONE to false

2. If seat I ==' ':
    print "do you want seat #I??"
    Get answer
    if answer=='Y':
        set seat I to 'X'
        set Done to True

3. If seat 2 ==' ' and Done == false:
    print "do you want seat #2??"
    Get answer
    if answer=='Y':
        set seat 2 to 'X'
        set Done to True
```

DATA STRUCTURES, ABSTRACT DATA TYPES, AND IMPLEMENTATIONS

- Consider example of an airplane flight with 10 seats to be assigned
- Tasks
 - List available seats
 - Reserve a seat
- How to store, access data?
 - 10 individual variables
 - An array of variables

USE ARRAY

Algorithm to List available seats

```
For number ranging from 0 to max_seats-1, do:

If seat[number] == ' ':

Display number
```

Algorithm to Reserve a seat

```
Reading number of seat to be reserved

If seat[number] is equal to ' ':
    set seat[number] to 'X'

Else

Display a message that the seat having this number is occupied
```

- This simple example does illustrate the concept of an Abstract Data Type
- ADT consists of
 The <u>collection of data items</u>
 Basic <u>operation</u> that must be performed on them
- In the example, a collection of data is a list of seats
- The basic operations are
 - (I) Scan the list to determine which seats are occupied
 - (2) change seat's status

ADT'S FOR PRIMITIVE DATA TYPES

BOOLEAN DATA

Data values: {false, true}

Operations:

and &&

or

not

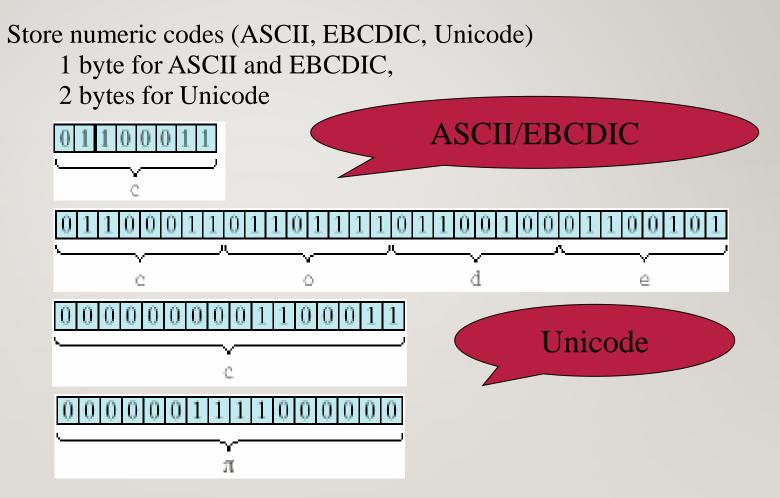
!

&&	0	1
0	0	0
1	0	1

П	0	1
0	0	1
1	1	1

<u>!x</u>
1
0

CHARACTER DATA



Basic operation: comparison to determine if Equal, Less than, Greater than, etc. use their numeric codes (i.e. use ordinal value)

INTEGER DATA

Non-negative (unsigned) integer:

Store its base-two representation in a fixed number w of bits (e.g., w = 16 or w = 32)

$$88 = 000000001011000_2$$

0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0

Signed integer:

Store in a fixed number w of bits using either of complement representations:

Operations: arithmetic operators for addition, subtraction, multiplication and division

REFERENCES

• Example taken from lecture slides by Dr. Bernard Chen Ph.D. at University of Central Arkansas