CS300 Data Structures

- data – factual information (as measurements or statistics) used as a basis for reasoning, discussion, or calculation
- structure – arrangement or relationship of elements (as particles, parts, or organs) in a substance, body, or system

- The New Merriam-Webster Pocket Dictionary

Data Type

A data type is:

An allowed set of values called the **domain**

A specified set of **operations** on the domain

Simple(Atomic) C Data Types

- char
- int
- float
- double

C has qualifiers that can precede a data type such as:

- short int

Questions

- Q1: What determines the domain of allowed values?
- Q2: What are the domain of values for an int?
- Q3: What determines the operations for a given data type?

Data Structures

A data structure can be thought of as a data type with values that

- Can be broken up into a set of component elements where each element is either atomic or another data structure

- Include a set of relationships (structure) involving the component elements

Abstract Data Types (ADTs)

- An abstract data type has two qualities:

  a) Irrelevant details are suppressed

  b) The data type being abstracted is isolated

Integer ADT

- Let us consider the specification for the integer ADT as follows:

ADT: Integer

Domain: All whole numbers i where -maxint <= i <= maxint

maxint is the maximum integer value that can be represented

Integer ADT Specification

Operations: Given i is an integer and f & g are expressions whose result is an integer, we define the following operations for C:

Unary Operators:

Operator	Results
Unary +	+f is the same as f
Unary -	-f changes the sign of f
Assignment =	i = f assigns the integer value of f to i
Binary +	f + g is the addition of two integer values

You get the idea

String ADT

Integer is an atomic ADT.

How might we specify a structured data type such as a String?

Before specifying the String ADT, we need to answer certain questions.

String ADT Questions

- What type are the component elements?
- What structure does the type have?
- What are the domain of possible values
- What operations exist?

String ADT Specification

- **Elements**: Type char excluding the null terminating character.
- **Structure**: Characters are arranged linearly
- **Domain**: All combinations of strings of length 0 to 255 that can be formed from the character set

String ADT Specification

- Operations

1) function strLength (s)

   results: returns the number of characters in the string s

2) function strEqual (s1, s2)

   results: returns true iff strLength (s1) equals strLength (s2) and the i^{th} character of s1 and s2 are equal for all i where 1 <= i <= strLength (s1)

3) function strConcat (s1, s2)

   results: string s2 is concatenated on the end of string s1; if the result exceeds the max string length, the characters are dropped

String ADT Specification

- Operations Continued

4) function strAppend (s, ch)

requires: strLength (s) < max string length

results: ch is added to the end of s increasing the length by 1

5) function strReverse (s)

results: the characters of s are reversed “abc” is “cba”

6) function strClear (s)

results: the string s is made empty

String ADT Specification

- Operations Continued

  7)function strCopy (s1, s2)

    results: string s2 is copied into string s2

ADT Implementation

Now that the String ADT has been specified, we can focus on the best implementation choice.

Before writing the code for each of the functions, we need to decide how we are going to represent a string.

String Represention Possibilities

```
typedef char *String;


typedef struct
        {
          int length;
          char data[256];
        } String;
```

Problem

- For each String representation, implement each of the following functions in C:
  - strLength
  - strCopy
  - strConcat