CS300 Data Structures

- data – factual information
- structure – arrangement or relationship of elements

- The New Merriam-Webster Pocket Dictionary

Data Type

A data type is: In C++?

- An allowed set of values called the **domain**
  - What determines the domain of allowed values?
  - What are the domain of values for an int?
- A specified set of **operations** on the domain
  - What determines the operations for a given data type?

Goal

- Build a useful data structure
- Test the data structure
- Hand that data structure off to a customer
  - future you
  - teammate
  - paying customer
  - your professor
- Profit / pass the class!

Primitive (Atomic) C Data Types

- char
- int
- float
- double

Defined by the C language specification.

sizeof(int)

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

- short int
- unsigned char

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 (hidden)

  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
  INT_MIN <= i <= INT_MAX   <limits.h>

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:

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

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 are the domain of possible values

- What operations exist?

Language independent

Language specific

- What type are the component elements?

- What structure does the type have?

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 the max string length 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 ith 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

*Why 1 and not 0?

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 s1

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.

Code defensively! Your Data Structure should never crash the user's program.

String Represention Possibilities

```
typedef char *String;


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

Problem

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