# C Language Concepts

CS 231

*ADTs and Multiple Files for Program*

# #include Preprocessor Directive

- *place a copy of specified file in place of directive*

  - *#include <filename> //look in standard library*

  - *#include "filename" //look in same directory as file being compiled*

- *.h extension is for header, file should contain declarations; for a program compiled from multiple .c files, .h files should contain common declarations.*

# C Structures

- *keyword struct - define a data type from other types*

- *syntax*

  struct *name* {

      *type varName; //as many as needed*

  *};*

- *Provides functionality of instance variables of objects.*

- *Above reserves no storage space.*

# Variables of struct type

- struct person{
  char * fName;
  char * lName;
  } pers1;


  struct person pers2;

- pers1 and pers2 are both variables of type struct person

# Accessing Members of struct

- *if pers1 is a variable of type struct person, use dot notation as in:*

  *pers1.fName*

- *if persPtr is a pointer to a struct person, use pointer notation as in:*

  *persPtr -> fName*

# typedef

- *define a synonym for a type*

- *Example: typedef struct person human;*

- *Now can use human instead of struct person in declaration*

# Memory Allocation

- *malloc takes as argument a number of bytes to allocate and returns a (void \*) pointer to memory (NULL on error)*

- *free deallocates memory, pass free pointer returned by malloc*

- *Use sizeof function to get number of bytes of data type.*

# Code Example

- *doublyLinkedList.c has code for a linked list of a data type defined elsewhere*

- *charMain.c has code to do work with list of char*

- *intMain.c has code to do work with list of int*

- *print char data or int data by using code in printCharData.c or printIntData.c*

- *header files contain declarations/definitions for doublyLinkedList, int data, and char data*