

Structures and Functions Examples

Last updated 6/4/24

These slides provide examples of structures in functions

Structures and Functions Examples

- Simple inventory example

```
/*
 * structs.c
 *
 * Created on: Jun 4, 2024
 * Author: johnsontimoi
 */
////////////////////////////////////
//
// program to model inventory
//
////////////////////////////////////
#include <stdio.h>
#include <stdint.h>

// item structure
typedef struct{
    uint8_t id_no;
    int quantity;
    char desc[20];
} inv_item;

void splash(void);
void add_item(int num, inv_item * the_item_ptr);
int remove_item(int num, inv_item * the_item_ptr);
void print_item(const inv_item the_item);
void print_inventory(int num_items, const inv_item[]);
```

```
int main(void){
    setbuf(stdout, NULL);

    splash();

    // create items in an array
    inv_item books[5] = {
        {.id_no = 0, .quantity = 0, .desc = "Dune"},
        {.id_no = 1, .quantity = 0, .desc = "The Hobbit"},
        {.id_no = 2, .quantity = 0, .desc = "The Hunger Games"},
        {.id_no = 3, .quantity = 0, .desc = "Jaws"},
        {.id_no = 4, .quantity = 0, .desc = "Twister"}
    };

    // add some items to the inventory
    add_item(3, &books[0]);
    add_item(5, &books[1]);
    add_item(7, &books[2]);
    add_item(9, &books[3]);
    add_item(11, &books[4]);

    print_inventory(5, books);
    printf("\n");

    // remove some items - no checking for now
    remove_item(2, &books[0]);
    remove_item(2, &books[1]);
    remove_item(2, &books[2]);
    remove_item(2, &books[3]);
    remove_item(2, &books[4]);

    print_inventory(5, books);

    return 0;
} // end main
```

Structures and Functions Examples

- Simple inventory example – cont'd

```
//////////////////////////////////
// add_item()
//
// function to add to the quantity of an item
//
// inputs: number of items to add, pointer to the item structure
// outputs: no return, modifies the items quantity
//
//////////////////////////////////
void add_item(int num, inv_item * the_item_ptr){
    the_item_ptr->quantity += num;

    return;
} // end add_item

//////////////////////////////////
// remove_item()
//
// function to remove a quantity of an item
//
// inputs: number of items to remove, pointer to the item structure
// outputs: returns 0 if OK, 1 if not enough items, modifies the items quantity
//
//////////////////////////////////
int remove_item(int num, inv_item * the_item_ptr){
    int tmp_rtn;
    if(the_item_ptr->quantity < num){
        tmp_rtn = 1;
    }else{
        the_item_ptr->quantity -= num;
        tmp_rtn = 0;
    }

    return tmp_rtn;
} // end remove_item
```

```
//////////////////////////////////
// print_item()
//
// function to print the quantity of an item
//
// inputs: item structure
// outputs: no return, prints the items quantity
//
//////////////////////////////////
void print_item(const inv_item the_item){
    printf("item %i : %s, has %i items in inventory\n", the_item.id_no, the_item.desc, the_item.quantity);

    return;
} // end print_item

//////////////////////////////////
// print_inventory()
//
// function to print the entire inventory
//
// inputs: number of items, an array (the full inventory)
// outputs: no return, prints the full inventory of items
//
//////////////////////////////////
void print_inventory(int num_items, const inv_item the_inventory[]){
    int i;
    for(i = 0; i < num_items; i++){
        print_item(the_inventory[i]);
    }
    return;
} // end print_inventory

//////////////////////////////////
// splash()
//
// function to print a splash screen
//
// inputs: none
// outputs: prints information about the program
//
//////////////////////////////////
void splash(void){
    printf("//////////////////////////////////\n");
    printf("//\n");
    printf("// Inventory structure program\n");
    printf("//\n");
    printf("//////////////////////////////////\n");

    return;
} // end splash
```

```
<terminated> (exit value: 0) Class_Notes_Project.exe [C/C++ A
//////////////////////////////////
//
// Inventory structure program
//
//////////////////////////////////
item 0 : Dune, has 3 items in inventory
item 1 : The Hobbit, has 5 items in inventory
item 2 : The Hunger Games, has 7 items in inventory
item 3 : Jaws, has 9 items in inventory
item 4 : Twister, has 11 items in inventory

item 0 : Dune, has 1 items in inventory
item 1 : The Hobbit, has 3 items in inventory
item 2 : The Hunger Games, has 5 items in inventory
item 3 : Jaws, has 7 items in inventory
item 4 : Twister, has 9 items in inventory
```

Structures and Functions Examples

- Move a sprite around a screen

```
////////////////////////////////////
/*
 * structs.c
 *
 * Created on: Jun 4, 2024
 * Author: johnsontimoj
 */
////////////////////////////////////
//
// program to move a sprite around
//
// a sprite is any graphic element
// that moves around the screen
//
////////////////////////////////////
#include <stdio.h>
#include <stdint.h>

// sprite structure
typedef struct{
    uint8_t player_no;
    uint8_t xloc;
    uint8_t yloc;
    uint8_t visible;
} player;

void splash(void);
int move_player(int moveX, int moveY, player * the_player_ptr);
void print_player(const player the_player);

// define the game window limits
#define XMIN 0
#define YMIN 0
#define XMAX 200
#define YMAX 200
```

```
int main(void){
    setbuf(stdout, NULL);

    splash();

    // game flag
    uint8_t continue_game;
    continue_game = 1;
    char tmp_char;

    // increment variables - reused for all players
    // using int instead of int8_t due to scanf limitations
    int xinc;
    int yinc;

    // create players
    player P1 = {.player_no = 1, .visible = 1};
    player P2 = {.player_no = 2, .visible = 1};

    // start at center (side by side)
    P1.xloc = 100;
    P1.yloc = 100;
    P2.xloc = 110;
    P2.yloc = 100;

    while(continue_game){
        // update players
        printf("Enter x and y increments for Player 1: ");
        scanf("%i %i", &xinc, &yinc);
        move_player(xinc, yinc, &P1);

        printf("Enter x and y increments for Player 2: ");
        scanf("%i %i", &xinc, &yinc);
        move_player(xinc, yinc, &P2);

        // output players
        print_player(P1);
        print_player(P2);

        // continue??
        printf("enter any character except z to continue, enter z to stop: ");
        scanf(" %c", &tmp_char);
        if(tmp_char == 'z')
            continue_game = 0;
    } // end while

    return 0;
} // end main
```

Structures and Functions Examples

- Move a sprite around a screen – cont'd

```
//////////////////////////////////
// move_player()
//
// function to move a player (if allowed)
//
// inputs: x increment, y increment, pointer to the player structure
// outputs: returns 1 if the player can move, 0 if not, modifies the players position
//
//////////////////////////////////
int move_player(int moveX, int moveY, player * the_player_ptr){
    int rtn_val;
    rtn_val = 0;

    // can it move in x direction?
    if((the_player_ptr->xloc + moveX > XMIN) && (the_player_ptr->xloc + moveX < XMAX)){
        the_player_ptr->xloc += moveX;
        rtn_val = 1;
    }

    // can it move in y direction?
    if((the_player_ptr->yloc + moveY > YMIN) && (the_player_ptr->yloc + moveY < YMAX)){
        the_player_ptr->yloc += moveY;
        rtn_val = 1;
    }

    return rtn_val;
} // end move_player
```

```
//////////////////////////////////
// print_player()
//
// function to print a player to the screen (if visible)
// in this case we don't have graphics so we just print the location
//
// inputs: the player structure
// outputs: prints the players location
//
//////////////////////////////////
void print_player(const player the_player){
    if(the_player.visible == 1)
        printf("Player %hu is located at %hu-%hu\n", the_player.player_no, the_player.xloc, the_player.yloc);

    return;
} // end print_player
```

```
//////////////////////////////////
// splash()
//
// function to print a splash screen
//
// inputs: none
// outputs: prints information about the program
//
//////////////////////////////////
void splash(void){
    printf("//////////////////////////////////\n");
    printf("//\n");
    printf("// Move the sprite program\n");
    printf("//\n");
    printf("//////////////////////////////////\n");

    return;
} // end splash
```

```
<terminated> (exit value: 0) Class_Notes_Project.exe [C/C++ Application] Z:\msoc
//////////////////////////////////
//
// Move the sprite program
//
//////////////////////////////////
Enter x and y increments for Player 1: -5 7
Enter x and y increments for Player 2: 5 7
Player 1 is located at 95-107
Player 2 is located at 115-107
enter any character except z to continue, enter z to stop: f
Enter x and y increments for Player 1: 20 -10
Enter x and y increments for Player 2: -15 2
Player 1 is located at 115-97
Player 2 is located at 100-109
enter any character except z to continue, enter z to stop: z
```