

## STRUCTURES

A structure is a collection of variables that are referenced under one name. The collection of variables is referred to as the elements of the structure. The elements normally have a relation to one another, and the name of the structure normally is descriptive of the relationship. Structures are called compound data types because they consist of several different, yet logically connected, elements.

A *structure template* describes how a structure is put together. It does not actually create a structure; it merely defines the template. The user must explicitly create a structure variable (or variables) that conform to the structure template definition. A structure template is shown below:

```
struct Person
{
    char  firstName[30];
    char  middleName[30];
    char  lastName[30];
    char  ssn[12];
    int   heightInches;
    float weightPounds;
};
```

In this example, `struct` is called the keyword, and `Person` is called the structure tag. Enclosed within the curly braces are the member variables. A structure variable can be declared by providing a structure variable name between the closing curly brace and the semi-colon. Structure variables can also be created using the keyword and tag as shown below:

```
struct Person Man;
struct Person Woman;
struct Person Anybody;
struct Person Moe, Larry, Curley;
```

A programmer can also use the typedef facility in C Programming to type-define a structure. An example is shown below:

```
typedef struct Book
{
    char  title[100];
    char  author[100];
    float cost;
} BOOK;

BOOK MyBook;
BOOK MyBook, YourBook;
```

**UNIVERSITY OF MASSACHUSETTS LOWELL**  
**Department of Continuing Education**

**90.267**

**C Programming**

**Lecture 7, Rev. 1.0**

---

```
#include <stdio.h>
#include <string.h>

typedef struct Person
{
    char  firstName[31];
    char  lastName[31];
    char  ssn[12];
    int   heightInches;
    float weightPounds;
} PERSON;

/*-----
 * Function Name: displayPerson()
 *
 * Description:   This program displays a structure variable
 *
 * Inputs:       A - PERSON structure variable
 *
 * Output:       None
 *
 * Return:       None
 *-----*/
void displayPerson(PERSON A)
{
    printf("\n-----");
    printf("\nName: %s %s", A.firstName, A.lastName);
    printf("\nSSN: %s", A.ssn);
    printf("\nHeight: %d in., Weight: %.1f lbs.",
           A.heightInches, A.weightPounds);
    printf("\n-----\n");
}

/*-----
 * Function Name: main()
 *
 * Description:   This program demonstrates declaring, initializing,
 *               changing, and displaying a structure variable
 *
 * Inputs:       None
 *
 * Output:       None
 *
 * Return:       Status
 *-----*/
int main(void)
{
    PERSON Man;

    /* Fill in the information */
    strcpy(Man.firstName, "Joseph");
    strcpy(Man.lastName, "Russell");
}
```

**UNIVERSITY OF MASSACHUSETTS LOWELL**  
**Department of Continuing Education**

**90.267**

**C Programming**

**Lecture 7, Rev. 1.0**

```
strcpy(Man.ssn, "012-34-5678");
Man.heightInches = 72;
Man.weightPounds = 180;

/* Display the information */
displayPerson(Man);

/* Get new information */
printf("\nEnter Last Name: ");
scanf("%30s", Man.lastName);

printf("Enter First Name: ");
scanf("%30s", Man.firstName);

printf("Enter SSN: ");
scanf("%11s", Man.ssn);

printf("Enter height, weight: ");
scanf("%d,%f", &Man.heightInches, &Man.weightPounds);

/* Display the information */
displayPerson(Man);

return 0;
}
```

Sample Run,

```
Z:\apc-cc-conduit\Images>a
```

```
-----
Name: Joseph Russell
SSN: 012-34-5678
Height: 72 in., Weight: 180.0 lbs.
-----
```

```
Enter Last Name: Doe
Enter First Name: John
Enter SSN: 352-87-1029
Enter height, weight: 70, 175.5
```

```
-----
Name: John Doe
SSN: 352-87-1029
Height: 70 in., Weight: 175.5 lbs.
-----
```

```
Z:\apc-cc-conduit\Images>
```

## ARRAY OF STRUCTURES

An array of structures can be declared in a fashion similar to an array of variables. The dimension of the structure is determined at compile time explicitly. Because the dimension of the array is determined at compile time, any change in the dimension requires a recompile of the program.

```
struct Person People[3];      /* Array of 3 people */
struct Person Students[50];   /* Array of 50 students */

#define NUM_BOOKS 100
struct Book MyLibrary[NUM_BOOKS];

typedef struct Date
{
    unsigned int month;
    unsigned int day;
    unsigned int year;
} DATE;
DATE Commitments[25];
```

### **Example:**

```
#include <stdio.h>

typedef struct Date
{
    unsigned int month;
    unsigned int day;
    unsigned int year;
} DATE;

int main(void)
{
    DATE HourExamDates[] = { {10, 5, 2004}, {11, 9, 2004} };

    printf("Exam #1 Date: %02d/%02d/%04d.\n", HourExamDates[0].month,
        HourExamDates[0].day,
        HourExamDates[0].year);

    printf("Exam #2 Date: %02d/%02d/%04d.\n", HourExamDates[1].month,
        HourExamDates[1].day,
        HourExamDates[1].year);

    return 0;
}
```

Sample Run,

Exam #1 Date: 10/05/2004. Exam #2 Date: 11/09/2004. Press any key to continue
---

**UNIVERSITY OF MASSACHUSETTS LOWELL**  
**Department of Continuing Education**

**90.267**

**C Programming**

**Lecture 7, Rev. 1.0**

---

**Example:**

```
#include <stdio.h>
#include <string.h>

#define NUM_PEOPLE 3

typedef struct Book
{
    char title[100];
    char author[100];
    float cost;
} BOOK;

typedef struct Person
{
    char firstName[31];
    char lastName[31];
    char ssn[12];
    int heightInches;
    float weightPounds;
    BOOK favoriteBook;
} PERSON;

/*-----
 * Function Name: main()
 *
 * Description:   This program displays an array of structure variables
 *
 * Inputs:       A[] - Array of PERSON structure variables
 *
 * Output:       None
 *
 * Return:       None
 *-----*/
void displayPerson(const PERSON A[])
{
    int counter;

    for(counter = 0; counter < NUM_PEOPLE; counter++)
    {
        printf("\n-----");
        printf("\nName: %s %s",    A[counter].firstName,
                A[counter].lastName);
        printf("\nSSN: %s", A[counter].ssn);
        printf("\nHeight: %d in., Weight: %.1f lbs.",
                A[counter].heightInches,
                A[counter].weightPounds);
        printf("\nFavorite Book: %s", A[counter].favoriteBook.title);
        printf("\nAuthor: %s", A[counter].favoriteBook.author);
        printf("\nCost: $%.2f", A[counter].favoriteBook.cost);

        printf("\n-----\n");
    }
}
```

**UNIVERSITY OF MASSACHUSETTS LOWELL**  
**Department of Continuing Education**

**90.267**

**C Programming**

**Lecture 7, Rev. 1.0**

---

```
/*-----
 * Function Name: main()
 *
 * Description:   This program demonstrates declaring, initializing,
 *               changing, and displaying an array of structure
 *               variables.
 *
 * Inputs:       None
 *
 * Output:       None
 *
 * Return:       Status
 *-----*/
int main(void)
{
    PERSON People[NUM_PEOPLE];
    int counter;

    /* Get the information for each person */
    for(counter = 0; counter < NUM_PEOPLE; counter++)
    {
        printf("\n\nPerson #%d\n", counter+1);

        printf("Enter Last Name: ");
        scanf("%30s", People[counter].lastName);

        printf("Enter First Name: ");
        scanf("%30s", People[counter].firstName);

        printf("Enter SSN: ");
        scanf("%11s", People[counter].ssn);

        printf("Enter height, weight: ");
        scanf("%d,%f%c", &People[counter].heightInches,
                &People[counter].weightPounds);

        printf("Enter Favorite Book Title: ");
        gets(People[counter].favoriteBook.title);

        printf("Enter Favorite Book Author: ");
        gets(People[counter].favoriteBook.author);

        printf("Enter Favorite Book cost: ");
        scanf("%f", &People[counter].favoriteBook.cost);
    }

    /* Display the information for the people */
    displayPerson(People);

    return 0;
}
```

**UNIVERSITY OF MASSACHUSETTS LOWELL**  
**Department of Continuing Education**

**90.267**

**C Programming**

**Lecture 7, Rev. 1.0**

Sample Run,

```
Z:\apc-cc-conduit\Images>a

Person #1
Enter Last Name: Russell
Enter First Name: Joe
Enter SSN: 012-34-5678
Enter height, weight: 72, 180
Enter Favorite Book Title: New C Primer Plus
Enter Favorite Book Author: Mitchell Waite & Stephen Prata
Enter Favorite Book cost: 29.95

Person #2
Enter Last Name: Doe
Enter First Name: John
Enter SSN: 352-87-1029
Enter height, weight: 70, 175.5
Enter Favorite Book Title: Absolute C++
Enter Favorite Book Author: Walter Savitch
Enter Favorite Book cost: 69.95

Person #3
Enter Last Name: Smith
Enter First Name: Jane
Enter SSN: 222-33-4456
Enter height, weight: 68, 132
Enter Favorite Book Title: VxWorks Programmer's Guide
Enter Favorite Book Author: WindRiver
Enter Favorite Book cost: 20

-----
Name: Joe Russell
SSN: 012-34-5678
Height: 72 in., Weight: 180.0 lbs.
Favorite Book: New C Primer Plus
Author: Mitchell Waite & Stephen Prata
Cost: $29.95
-----

-----
Name: John Doe
SSN: 352-87-1029
Height: 70 in., Weight: 175.5 lbs.
Favorite Book: Absolute C++
Author: Walter Savitch
Cost: $69.95
-----

-----
Name: Jane Smith
SSN: 222-33-4456
Height: 68 in., Weight: 132.0 lbs.
Favorite Book: VxWorks Programmer's Guide
Author: WindRiver
Cost: $20.00
-----

Z:\apc-cc-conduit\Images>
```