Introduction to Computers

 \bigcirc

Uploaded By: anonymous

 \bigcirc

& Programming

Comp 1330/ First Semester 2024/2025

Instructor: Saif Harbia

Faculty of Engineering and Technology Department of Computer Science STUDENTS-HUB.com

Chapter 10

Structure

STUDENTS-HUB.com

Uploaded By: anonymous .

0

Chapter Objectives:

1. Declare a **struct** data type which consists of several data fields, each with its own name and data type.

· · · · · · · · ·

.

 \bigcirc

.

Uploaded By: anonymous

- 2. Use a **struct** to store data for a structured object or record.
- 3. Use <u>dot notation</u> to process individual fields of a structured object
- 4. Use **structs** as function parameters and to return function results
- 5. Understand the relationship between <u>parallel arrays</u> and <u>arrays of structured objects</u>.

And More...

STRUCTU

- **RE** Unlike an array, a **structure** can have individual components that contain data of different types.
- A single variable of a composite type designed for a **textbook** can store a book's <u>author, title, publisher and year</u>

textbook	=> structure
author, title, publisher, year	=> components

Each of these data items is stored in a separate component of the structure and can be referenced by using the component name.

Uploaded By: anonymous

Structure Type Definition:

 \bigcirc

Before a structured data object can be created or saved, the format of its components must be defined. (1)



 \bigcirc

#define STRSIZ 10

EXAMPLE 10.1

typedef struct {
 char name[STRSIZ];
 double diameter;
 int moons;
 double orbit_time,
 rotation_time;

/* equatorial diameter in km */ /* number of moons */ /* years to orbit sun once */ /* hours to complete one revolution on axis */

Uploaded By: anonymous

} planet_t;

A single variable of a composite type designed for planets can store a planet's
 name
 diameter

number of moons

the number of years to complete one solar orbit

the number of hours to make one rotation on its axis.

STUDENTS-HUB.com

3. 0 4.

- This type definition is a <u>template</u> that describes the format of a planet structure and the name and type of each component.
- A name chosen for a component of one structure may be the same as the name of a component of another structure or the same as the name of a variable. (1)
- The **typedef** statement itself allocates no memory. A variable declaration is required to allocate storage space for a structured data object.

The structured variables **current_planet**, **previous_planet**, and **blank_planet** all have the format specified in the definition of type **planet_t**.

 \bigcirc

Uploaded By: anonymous

 \succ Thus, the memory allocated for each consists of storage space for five distinct values.

Variable blank_planet, a structure of type planet_t



STUDENTS-HUB.com

 \succ

 \bigcirc

- A user-defined type like **textbook_t** can be used to declare both simple and array variables and to declare components in other structure types.
- A structure containing components that are data structures (<u>arrays</u> or <u>structs</u>) is sometimes called a **hierarchical structure**.
- The following definition of a structure type includes a component that is an <u>array of textbooks</u>.



MANIPULATING INDIVIDUAL COMPONENTS OF A STRUCTURED DATA OBJECT

 \succ

 \bigcirc

- We can reference a component of a structure by using the **direct component selection operator** , which is a **period**.
- The **period** is preceded by the name of a **structure** type variable and is followed by the name of a component.
- Once data are stored in a record, they can be manipulated in the same way as other data in memory.

```
strcpy(book.name, "MyBook");
                                 book.year = 2000;
                                                                                             \bigcirc
     printf("%s was published in %d.\n",
                                                           MyBook was published in 2000
                 book.year);
     book.title,
STUDENTS-HUB.com
                                                                           Uploaded By: anonymous
```

MANIPULATING INDIVIDUAL COMPONENTS OF A STRUCTURED DATA OBJEC

• **EXAMPLE 10.2**

strcpy(current_planet.name, "Jupiter"); current_planet.diameter = 142800; current_planet.moons = 16; current_planet.orbit_time = 11.9; current_planet.rotation_time = 9.925;

Variable current_planet, a structure of type planet_t

Jupit	е	r '	0	?	?
142800.0					8
16					
11.9					
9.925					
	Jupit 142800.0 16 11.9 9.925	Jupite 142800.0 16 11.9 9.925	Jupiter 142800.0 16 11.9 9.925	Jupiter\0 142800.0 16 11.9 9.925	Jupiter\0? 142800.0 16 11.9 9.925

Uploaded By: anonymous

Ο

S Figure 10.1



REVIEW OF OPERATOR PRECEDENCE

> See Table 10.1 p.572

Manipulating Whole Structures

- The name of a structure type variable used with no component selection operator refers to the <u>entire structure</u>.
- A new copy of a structure's value can be made by simply assigning one structure to another as in the following statement:

previous_planet = current_planet;

• **Program Style:** (1)

To help reduce confusion, in this chapter we choose user-defined type names that use lowercase letters and end in the suffix_t (2)

Uploaded By: anonymous

When a structured variable is passed as an **input argument** to a function, all of its component values are copied into the components of the function's corresponding formal parameter.

Uploaded By: anonymous

- **EXAMPLE 10.3, Figure 10.2 (1)**
- To display the value of our structure **current** planet, we would use the call statement:

print_planet(current_planet);

- Another function that would help us think of a planet as a data object is a function that would perform an <u>equality comparison</u> of two planets.
- Although C permits copying of a structure using the assignment operator, the equality and inequality operators CANNOT BE applied to a structured type as a unit.
- **Figure 10.3** (1)

- When such a variable is used as an **output argument**, the **addressof** operator must be applied in the same way that we would pass output arguments of the standard types **char**, **int**, and **double**.
 STUDENTS-HUB.com

Figure 10.4 (1)

- In order to use scanf to store a value in one component of the structure whose address is in plnp , we must carry out the following steps (*in order*):
- Follow the pointer in **plnp** to the structure.
- Select the component of interest.
- Unless this component is an array (e.g., component name in **Fig. 10.4**), get its address to pass to **scanf**.
 - When we check our precedence chart (see **Table 10.1**), we find that this reference **&*plnp.diameter** would attempt step 2 before step 1.
 - For this reason, the function in **Fig. 10.4** overrides the default operator precedence by \bigcirc parenthesizing the application of the indirect referencing (pointer-following) operator, the unary *.
- STUDENTS-HUB.com

 \bigcirc

Uploaded By: anonymous



TABLE 10.2 Step-by-Step Analysis of Reference &(*plnp).diameter

Reference	Туре	Value
plnp	planet_t *	address of structure that main refers to as current_planet
*plnp	planet_t	structure that main refers to as current_planet
(*plnp).diameter	double	12713.5
&(*plnp).diameter	double *	address of colored component of structure that main refers to as current_planet

Uploaded By: anonymous

0



> indirect component selection operator (->) (1)

 \bigcirc

If we rewrite the scan_planet function of Fig. 10.4 using the -> operator, the assignment to result will be

 \bigcirc

ano

```
result = scanf("%s%lf%d%lf%lf", plnp->name,
&plnp->diameter,
&plnp->moons,
&plnp->orbit_time,
&plnp->rotation_time);
S-HUB.com
```

10.3 FUNCTIONS WHOSE RESULT VALUES ARE STRUCTURED

A function that computes a struc result.

t car

iting a simple

- A local variable of the structure type can be allocated, filled with the desired data, and returned as the function result.
- The function does not return the address of the structure as it would with an array result; rather it returns the values of all components.
 - EXAMPLE 10.4, Fig. 10.6

 current_planet = get_planet();
 (1)
 status = scan_planet(¤t_planet);

 stude = status = scan_planet(¤t_planet);
 (1)
 (1)

 Uploaded By: anonymous
 (1)

10.3 FUNCTIONS WHOSE RESULT VALUES ARE STRUCTURED



10.3 FUNCTIONS WHOSE RESULT VALUES ARE STRUCTURED



10.5 PARALLEL ARRAYS AND ARRAYS OF STRUCTURES

Parallel Arrays: (1)

int id[50];	/* id numbers and	*/
double gpa[50];	/* gpa's of up to 50 students	*/
double x[NUM_PTS],	<pre>/* (x,y) coordinates of</pre>	*/
y[NUM_PTS];	/* up to NUM_PTS points	*/

.

()

Uploaded By: anonymous

Arrays id and gpa are called <u>parallel arrays</u> because the data items with the same subscript (for example, i) pertain to the same student (the ith student).

- Similarly, the **ith** elements of arrays \mathbf{x} and \mathbf{y} are the coordinates of one point.
- \bigotimes A better way to organize data collections like these is <u>shown next</u>.

STUDENTS-HUB.com

• • • • • • •

10.5 PARALLEL ARRAYS AND ARRAYS OF STRUCTURES

Declaring an Array of Structures:

STUDENTS-HUB.com

A more natural and convenient
organization of student data or polygon
points is to group the information
pertaining to one student or to one point in
a structure whose type we define. (1)

```
#define MAX STU 50
#define NUM PTS 10
typedef struct {
      int
             id:
      double gpa;
  student t;
typedef struct {
      double x, y;
} point t;
```

student_t stulist[MAX_STU];
point_t polygon[NUM_PTS];

Uploaded By: anonymous

10.5 PARALLEL ARRAYS AND ARRAYS OF STRUCTURES

- The data for the first student are stored in the structure **stulist[0]**.
- The individual data items are stulist[0].id and stulist[0].gpa .
- ➤ stulist[0].gpa is 2.71.

 \succ



.

9.4 RECURSIVE FUNCTIONS WITH ARRAY AND STRING PARAMETERS

CASE STUDY (Homework) P. 594 – 602

Universal Measurement Conversion

STUDENTS-HUB.com

 \bigcirc

 \bigcirc

Uploaded By: anonymous

 \bigcirc

10.7 COMMON PROGRAMMING ERRORS

STUDENTS-HUB.com

- When using the direct selection operator (.), always be aware of the type of the component selected, and use the value in a manner consistent with its type. (1)
- For example, if the component selected is an array, passing it to a function as an output argument does not require application of the <u>address-of operator.</u>
- If a structure type output parameter is used in a function, one can avoid the operator precedence problems associated with combining the indirection (*) and direct component selection (.) operators by using the indirect component selection operator (->).

Uploaded By: anonymous

10.7 COMMON PROGRAMMING ERRORS

- C allows the use of structure type values in assignment statements as function arguments and as function results, so one can easily forget that expressions of these types cannot be operands of equality comparators or arguments of **printf** and **scanf**.
- You can select simple components from a structure to use in these contexts, or you can write your own type-specific equality and I/O functions.

Refernces

Problem Solving and Program Design in C, 7th Ed., by Jeri R. Hanly and Elliot B. Koffman

 \bigcirc

Uploaded By: anonymous