Intro. Computing with the C Programming Language

Structures

Shin Hong

30 October 2023

Compound Values

- An array is a compound variable as it comprises a set of primitive variables
    - also, a string is a compound value

- A compound value is useful for representing an objects with many parts

- Two kinds of compound variables
    - array
        - each component is identified by an index
        - every component belongs the same type
    - structure
        - each component is identified by its name
        - components may belong to different types

Example. Point

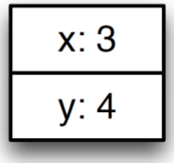
- A point in 2-D planar can be represented as a pair of two double numbers
  - (x, y)

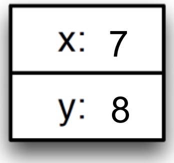
- Structure definition
  - ex. point.c

```
typedef struct {
   double x ;        //member or field
   double y ;
} Point_t ;

Point_t p1 ;
p1.x = 3.0 ;
p1.y = 4.0 ;

Point_t p2 = {6.0, 8.0} ;
```





Accessing Member Variable

- A member variable (or field) can be referred by the field-selection operator (i.e., dot op.)
  - ex.

```
printf(“%lf, %lf\n”, p1.x, p2.y) ;
double distance =
    sqrt((p1.x - p2.x) * (p1.x - p2.x) +
            (p1.y - p2.y) * (p1.y - p2.y)) ;
```

  - ex. `rectangles.c`

Operations on Structure

- The assignment operator works for structure variables as primitive variables

- Yet, other arithmetic operators do not work for structure variables

Passing Structure as Argument

- When a function call passes a structure variable as an argument, the values are copied to the parameters
  - the argument and the parameter are not the same variable, but two variables having the same values

- When a function returns a structure variable, the values are copied to the caller

Passing Pointer as Argument

- We can pass the pointer of a structure to a function that the structure variable is updated directly by the function

- We can use the pointing-to operator (`->`) to access a field from a pointer of a structure variable
  - ex. `reflection.c`