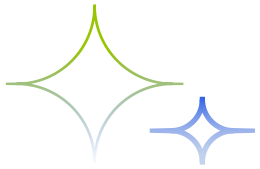


MONTANA
STATE UNIVERSITY

Programming with C I

Fangtian Zhong
CSCI 112

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

2024.03.22

Functions Whose Result Values are structured

- A function that computes a structured result can be modeled on a function computing a simple result.
- A local variable of the structure type can be allocated, fill 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.

Table Precedence and Associativity of Operators Seen So Far

Precedence	Symbols	Operator Names	Associativity
highest	a[j] f(...).	Subscripting, function calls, direct component selection	left
	++ --	Postfix increment and decrement	left
	++ -- ! - + & *	Prefix increment and decrement, logical not, unary negation and plus, address of, indirection	right
	(*type name*)	Casts	right
	* / %	Multiplication operators (multiplication, division, remainder)	left
	+ -	Binary additive operators (addition and subtraction)	left
	< > <= >=	Relational operators	left
	== !=	Equality/inequality operators	left
	&&	Logical and	left
	||	Logical or	left
lowest	= += -= *= /= %=	Assignment operators	right

Figure Function get_planet Returning a Structured Result Type

```
/*
 * Gets and returns a planet_t structure
 */
planet_t
get_planet (void)
{
        planet_t planet;

        scanf("%s%1f%d%1f%1f", planet.name,
                               &planet,diameter,
                               &planet,moons,
                               &planet.orbit_time,
                               &planet.rotation_time;
        return (planet);
}
```

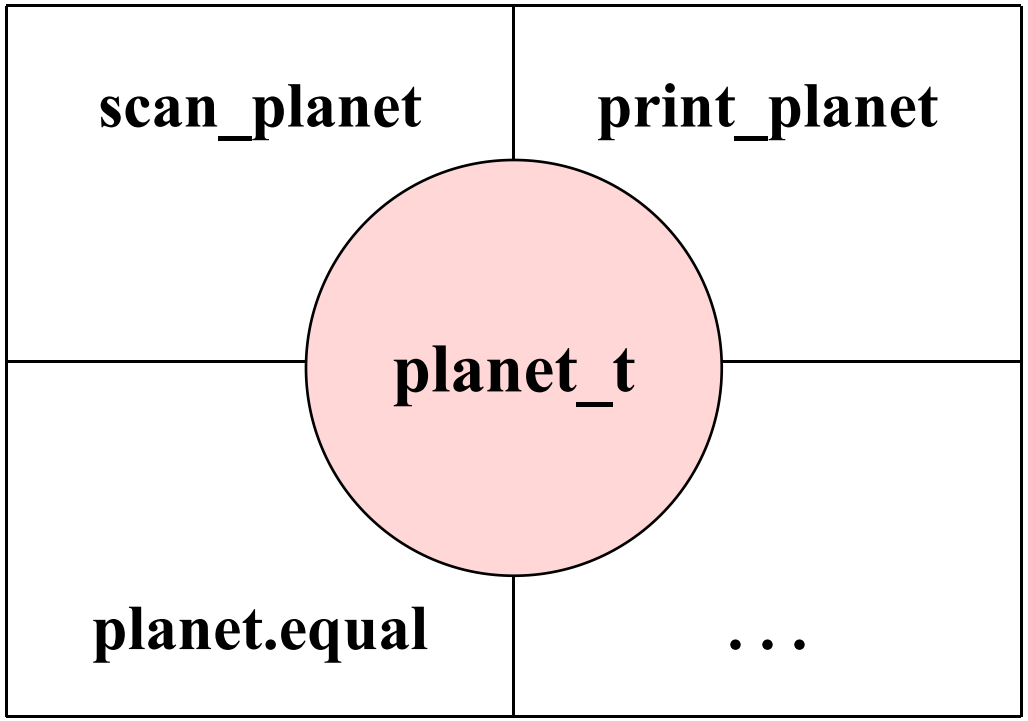
Figure Function to Compute an Updated Time Value

```
/*
 * Compute a new time represented as a time_t structure
 * and based on time of day and elapsed seconds.
 */
time_t
new_time(time_t time_of_day,        /* input - time to be
                                        updated                    */
         int      elapsed_secs)     /* input -seconds since last update  */
{
        int new_hr, new_min, new_sec;

        new_sec = time_of_day.second + elapsed_secs;
        time_of_day.second = new_sec % 60;
        new_min = time_of_day.minute + new-sec  /  60;
        time_of_day.minute = new_min  %  60;
        new_hr = time_of_day.hour + new_min  /  60;
        time_of_day.hour = new_hr  %  24;

        return (time_of_day);
}
```

Problem Solving with Structure Types

- **abstract data type (ADT**
  - a data type combined with a set of basic operations

Figure Data Type planet_t and Basic Operations



Header files: defining the interface

```
#include<stdio.h>
versus
#include"class.h"
```

- Angle brackets versus quotes tells compiler where to look for the file
- Gets copied in by preprocessor and then compiled in the .c file
- A .h file is never in the compile command

gcc –o exe –Wall program.c

.c files: the implementation

- Contain C code
- Do get compiled separately
- Are linked after compilation to form the executable

gcc –o exe –Wall program.c funcs.c

Header guards

- We don’t want to include headers multiple times, but they may reference one another
- Solution: header guards

```
#ifndef FILENAME_H
#define FILENAME_H
/* ... Declarations here ... */
#endif
```

THE END

Fangtian Zhong
CSCI 112

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

2024.03.22