

Programming with C I

Fangtian Zhong
CSCI 112

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

Functions Whose Result Values are structured

- **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.**

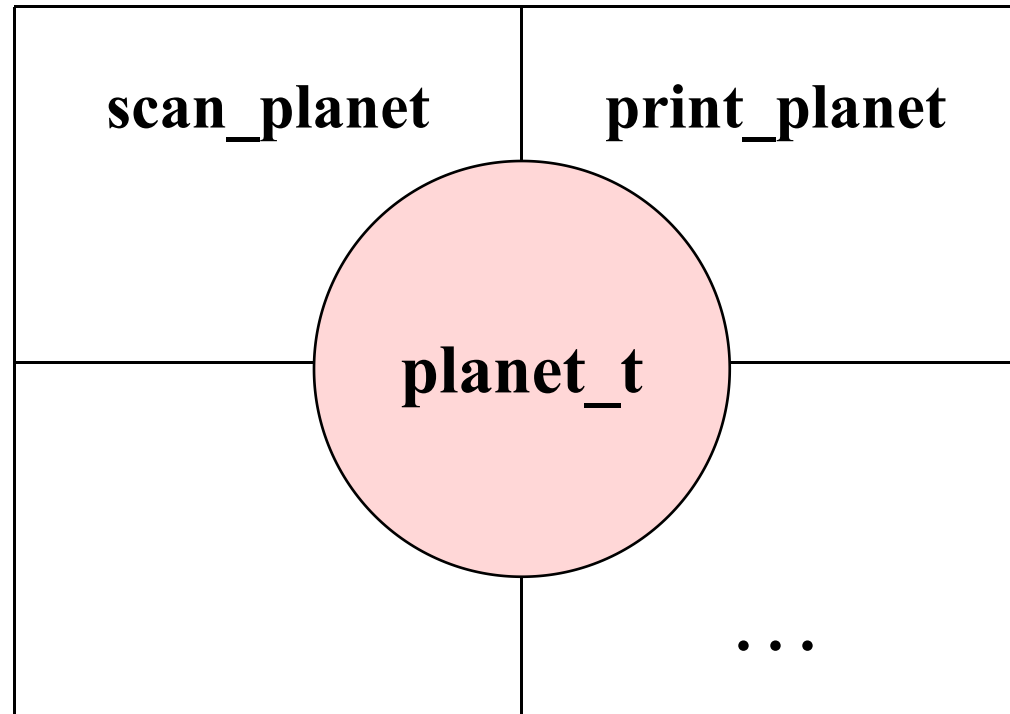
Figure Function get_planet Returning a Structured Result Type

```
/*  
 * Gets and returns a planet_t structure  
 */  
planet_t  
get_planet (void)  
{  
    planet_t planet;  
  
    scanf(“%s%lf%ld%lf%lf”, planet.name,  
        &planet.diameter,  
        &planet.moons,  
        &planet.orbit_time,  
        &planet.rotation_time;  
  
    return (planet);  
}
```

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"
```

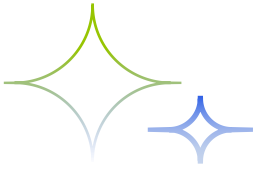
- **Angle brackets versus quotes tells compiler where to look for the file**
- **Gets copied in by preprocessor and then compiled in the .c file**
- **A .h file is never in the compile command**

```
gcc -o exe -Wall program.c
```

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