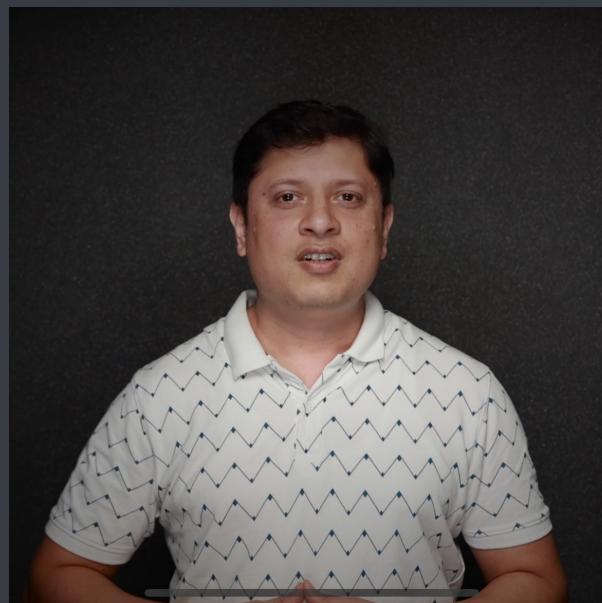


C Language

Structure



Saurabh Shukla (MySirG)

Agenda

- ① Introduction to structure
- ② Primitive and non-primitive data type
- ③ Defining structure
- ④ Declaring structure variable
- ⑤ Initializing structure variable during declaration
- ⑥ Initializing structure members after declaration
- ⑦ Taking input from user
- ⑧ Structure array
- ⑨ Function returning structure
- ⑩ Function call by passing structure
- ⑪ Structure Pointer

Introduction to Structure

- ① Structure can be a collection of dissimilar elements.
- ② Structure is a way to group variables.
- ③ Defining structure is creating custom data type

Primitive and Non- Primitive Data Types

int
char
float
double

Book ← Custom data type
Student ← Secondary data type
Customer ← user defined data type
Employee

- ① structure
- ② union
- ③ enumerator

Defining Structure (Creating Data Type)

Struct Book

{

int bookId;

char title[20];

float price;

}

Struct TypeName

{

// Declare variables

}

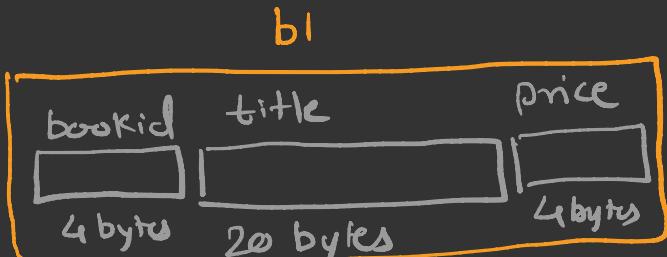
- Global vs local definition

Declaring Structure Variable

```
Struct Book
```

```
{  
    int bookId;  
    char title[20];  
    float price;
```

```
} b1, b2;
```

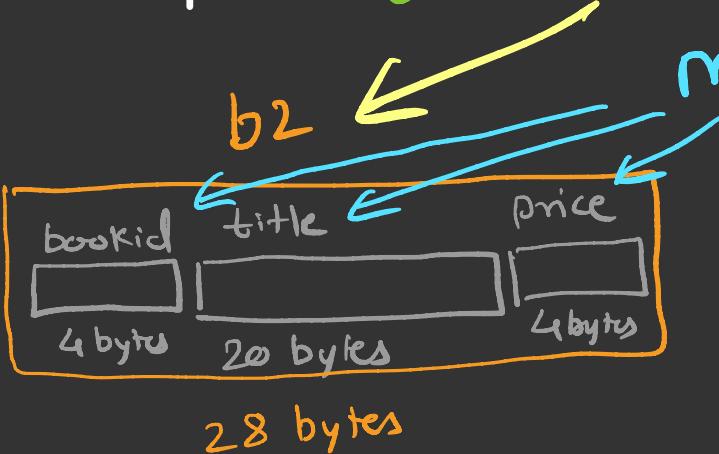


```
void f1()
```

```
{  
    struct Book b1, b2;
```

```
}
```

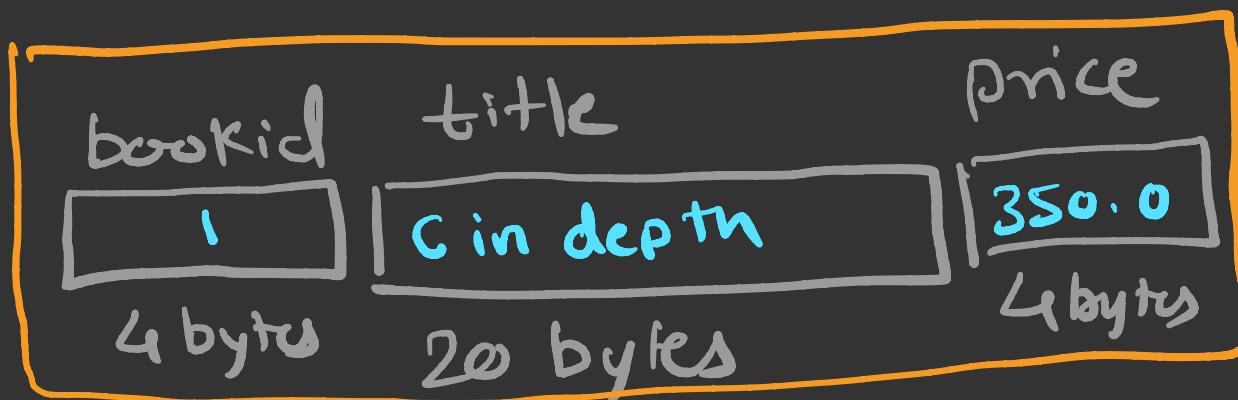
structure variable
member variable



Initializing structure variable during declaration

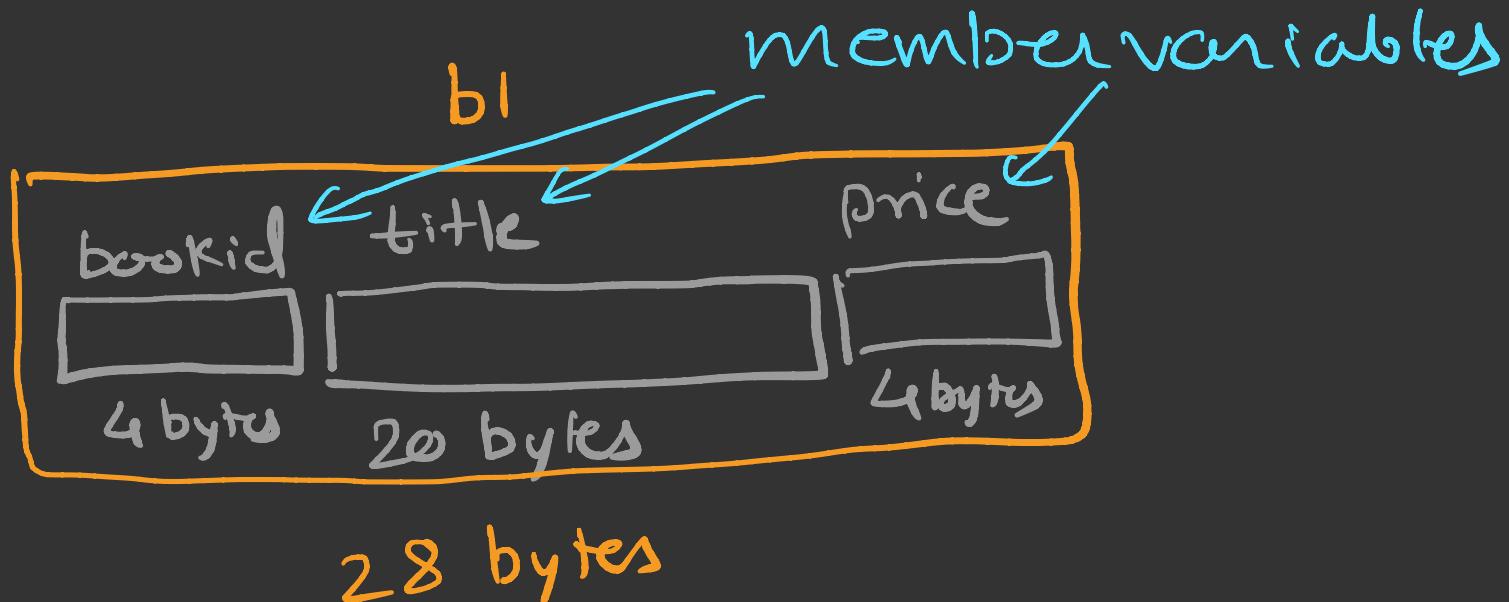
```
struct Book b1 = { 1, "C in depth", 350.0 };
```

b1



28 bytes

Initializing Structure member variables after declaration



You cannot access member variables directly.

`bookid = 2;` Error

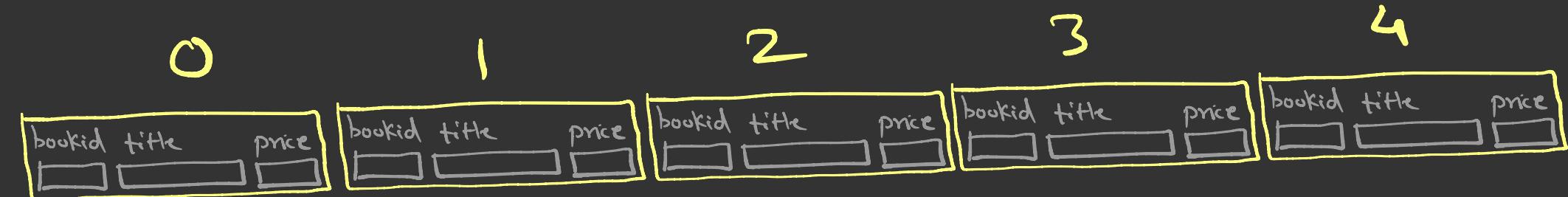
Structure Variable . memberVariable

Taking Input from user

```
printf("Enter bookid ,title and price");
scanf( ".d" , &b3.bookid );
fgets( b3.title , 20 , stdin );
scanf( ".f" , &b3.price );
```

Structure Array

struct Book b[5];



Function Returning Structure

```
struct Book FunctionName()
```

```
{
```

```
    struct Book b;
```

```
[  
=]  
—
```

```
return b;
```

```
}
```

Function call by passing Structure

— function (b)

— function (Struct Book b)

{
—
—
—
y

Structure Pointer

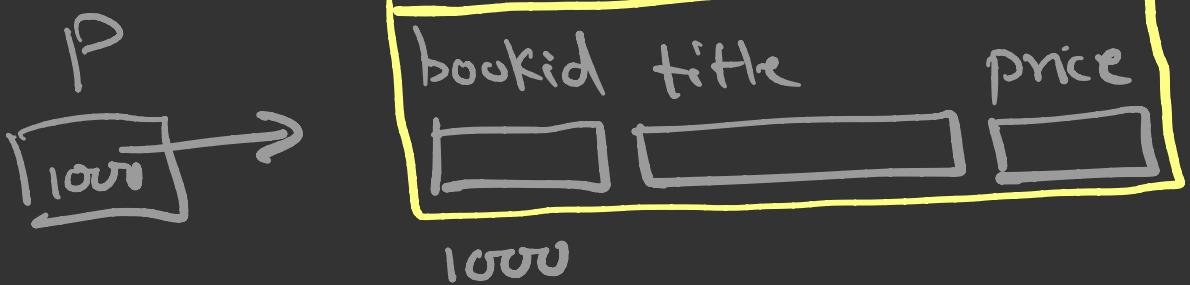
struct Book b1;

struct Book *p;

P = &b1;

*P ≈ b1

structure Pointer
b1



b1.bookid

StructureVariable.memberVariable

(*P).bookid

structurePointer → memberVariable

P → bookid

Struct Employee

```
{  
    int empid;  
    char name[20];  
    float salary;  
};
```

```
int main()
```

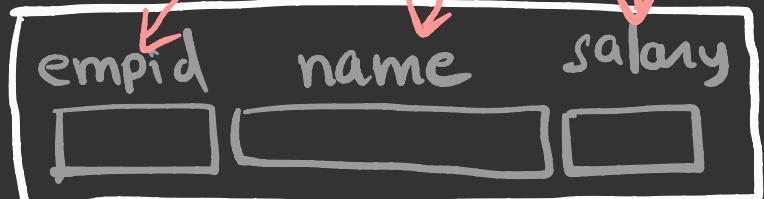
```
{
```

```
struct Employee e1,e2;
```

```
e1
```

structure
variable

structure
member variable



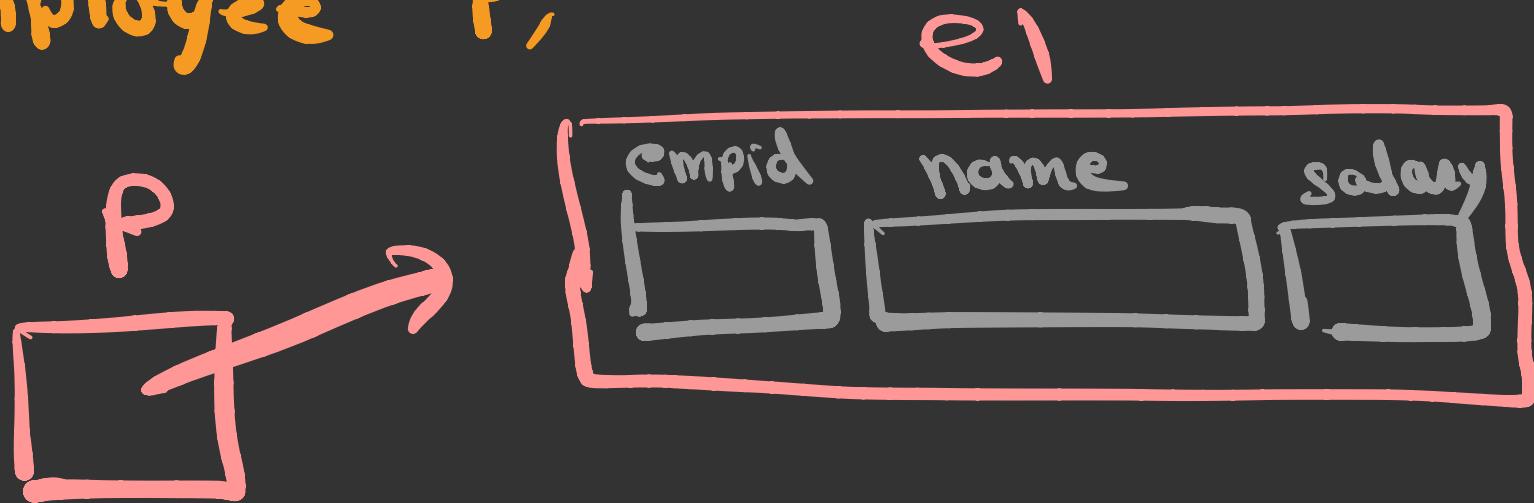
```
}
```

structureVariable . memberVariable

struct Employee e1;

struct Employee *p;

p=&e1;



*p ≈ e1

e1.empid

p->empid

(*p).empid