# 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
Student
Customer
Employee

Custom data type
Secondary data type
user defined data type

# Defining Structure (Creating Data Type)

```
struct Book
{
        int bookid;
        char title[20];
        float price;

};
```

```
struct TypeName
{
        //Declare variables

};
```

- Global vs local definition

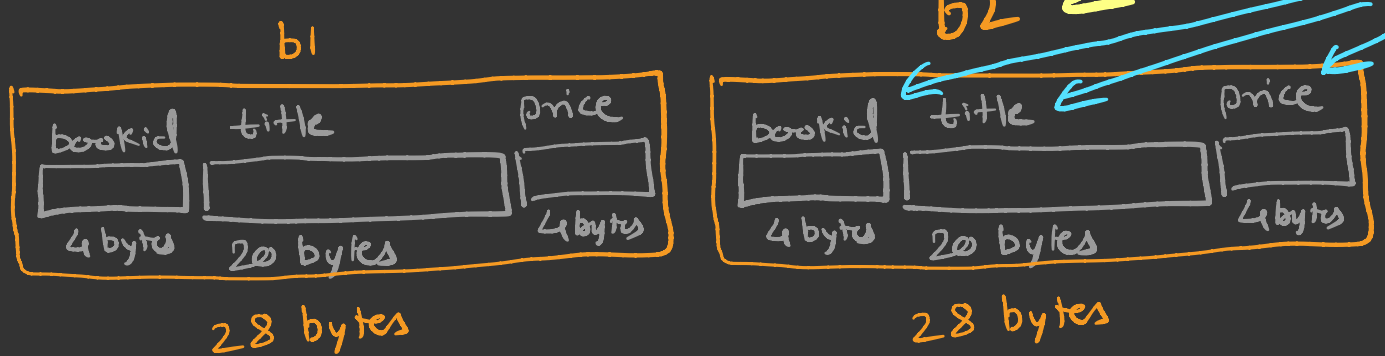
# Declaring Structure Variable

```
struct Book
{
    int bookid;
    char title[20];
    float price;
} b1, b2 ;
```

```
void f1()
{
    struct Book b1, b2;
}
```

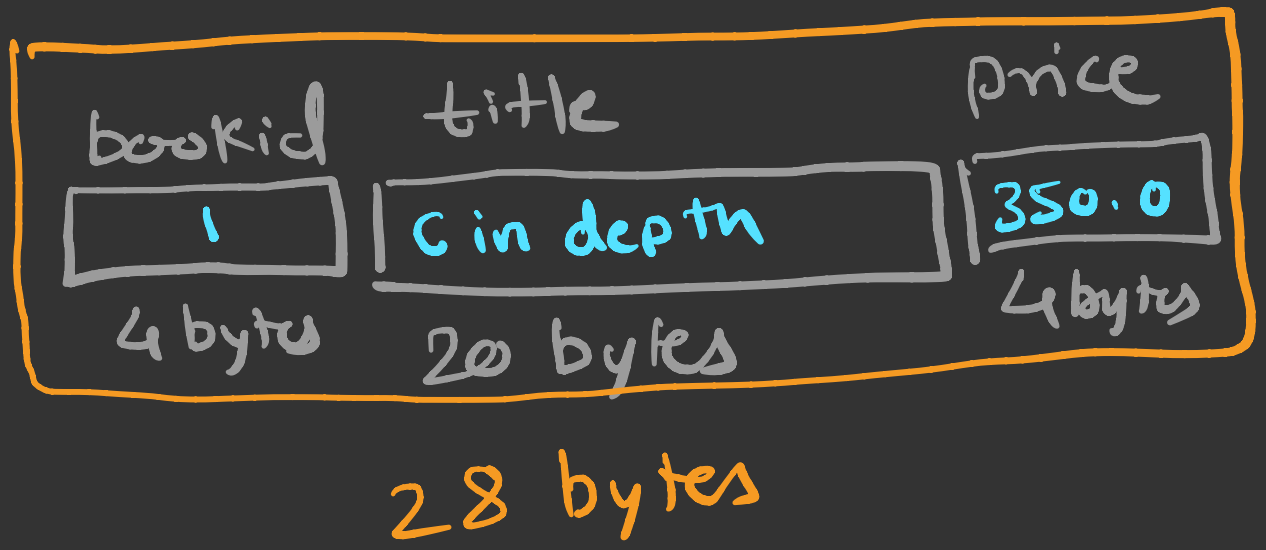
structure variable

member variable

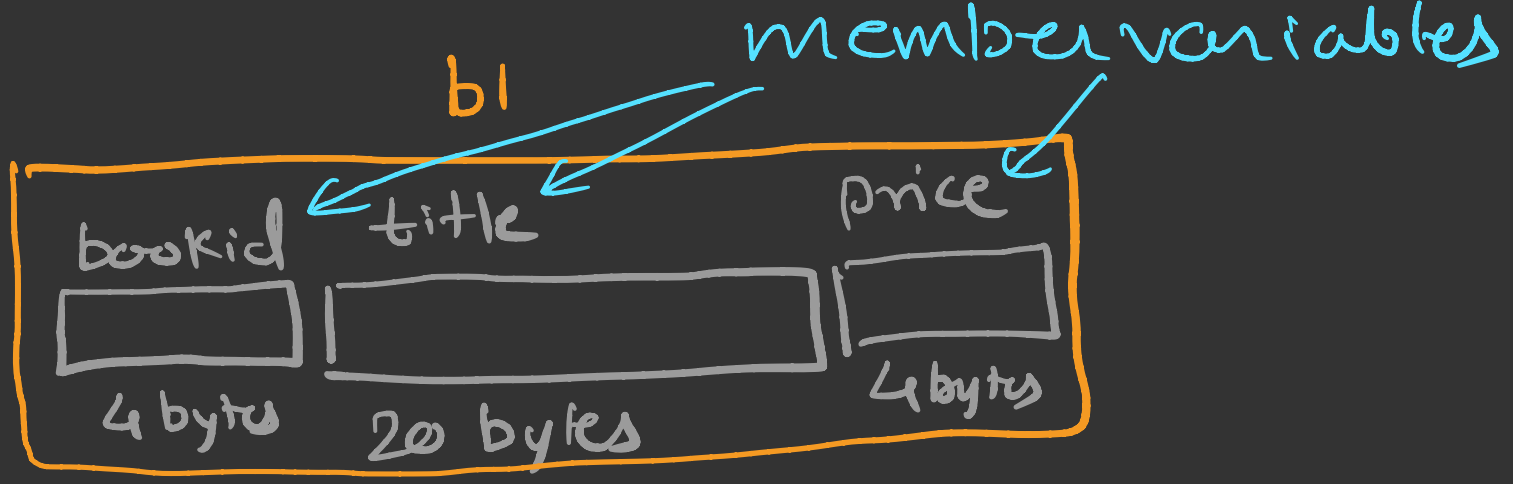


# Initializing structure variable during declaration

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



# Initializing Structure member variables after declaration




28 bytes

You cannot access member variables directly.

bookid = 2; Error

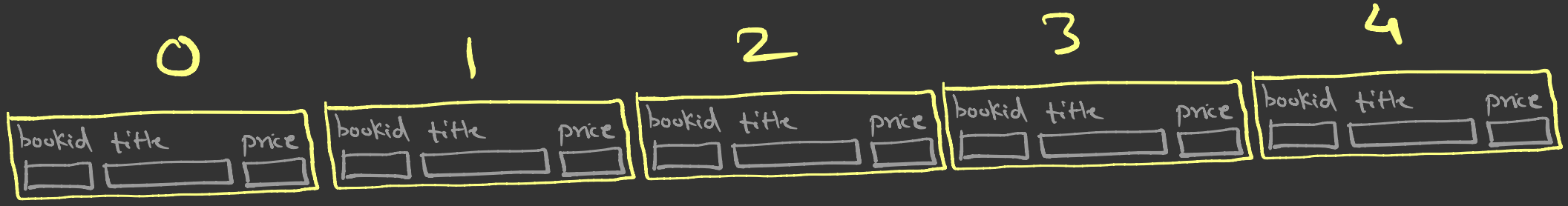
StructureVariable . memberVariable

# 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 ( b1 )

- function ( struct Book b )

{

=

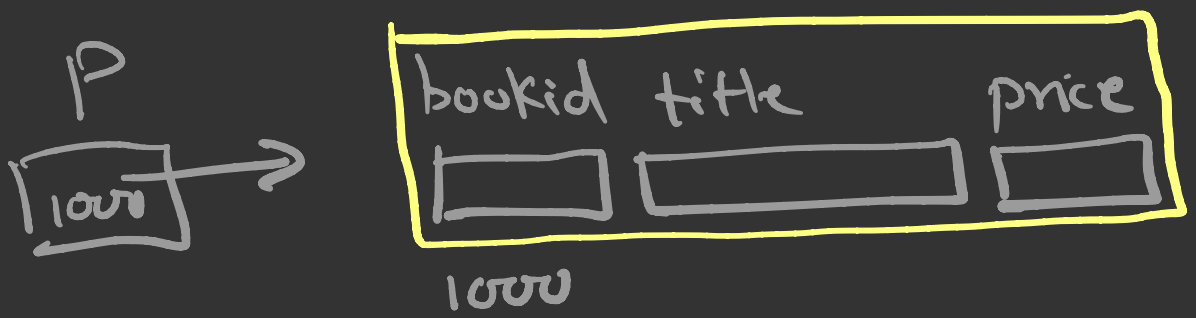
}

# Structure Pointer

```
struct Book b1;
struct Book *p;
```

structure Pointer

```
p = &b1;
```




*p ≒ b1

b1.bookid

structureVariable . memberVariable

(*p). bookid

structurePointer -> memberVariable

p -> bookid