

# A job ready bootcamp in C++, DSA and IOT

## 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

# Defining Structure (Creating Data Type)

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

- Global vs local definition

# Declaring Structure Variable

```
Struct Book
```

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

```
} b1, b2;
```

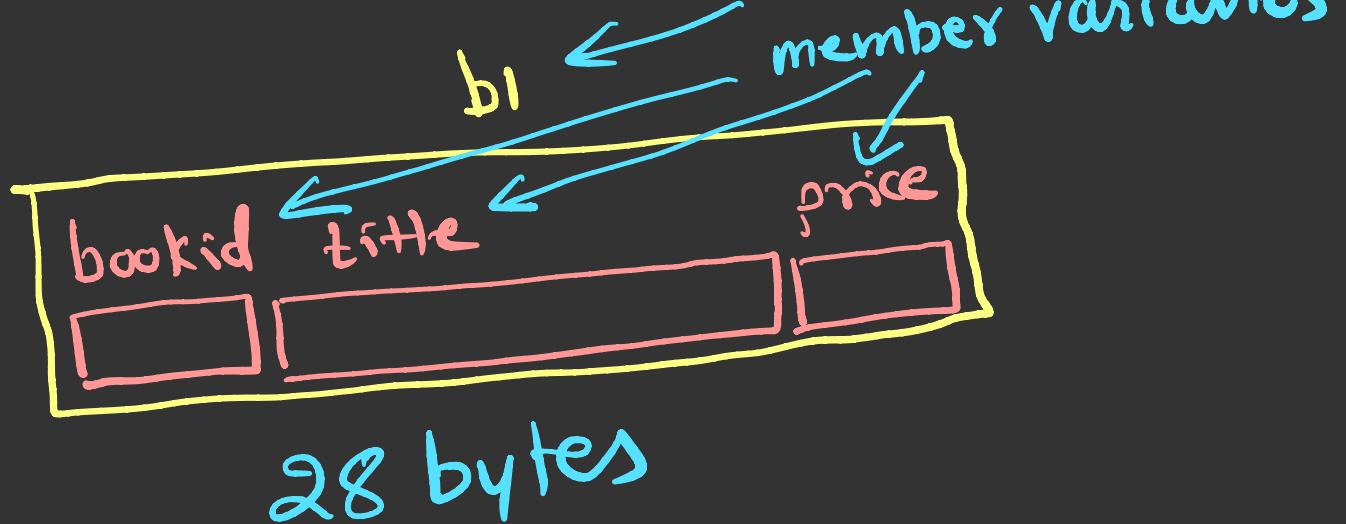
```
void f1()  
{  
    struct Book b1, b2;  
}
```

# Initializing structure variable during declaration

```
void f1()
{
    ...
}
```

datatype  
struct Book b1 = { 1, "CPP", 340 };

Structure Variable



## Initializing Structure member variables after declaration

struct Book b2;

b2.bookid = 2;

strcpy(b2.title, "Java");

b2.price = 425.0f;

StructureVariable.memberVariable

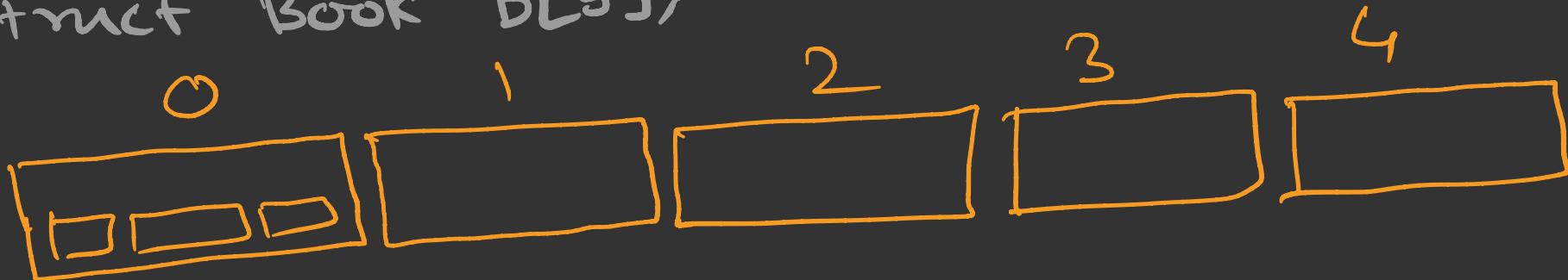


## Taking Input from user

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

# Structure Array

struct Book b[5];



`b[0].bookid = 5;`

`b[1].bookid = 6;`

# Function Returning Structure

```
struct Book input()
```

```
{
```

```
    struct Book b;  
    printf("Enter id, title & price");  
    scanf("%d", &b.bookid);  
    fflush(stdin);  
    fgets(b.title, 20, stdin);  
    scanf("%f", &b.price);  
    return b;
```

```
}
```

# Function call by passing Structure

```
void display(struct Book b)
```

```
{
```

```
    printf("%d %s %f", b.bookid, b.title, b.price);
```

```
}
```

display(b1);

fgets( )

0 1 2 3 4 5

title

[JavaIn\0]

[JavaIn]

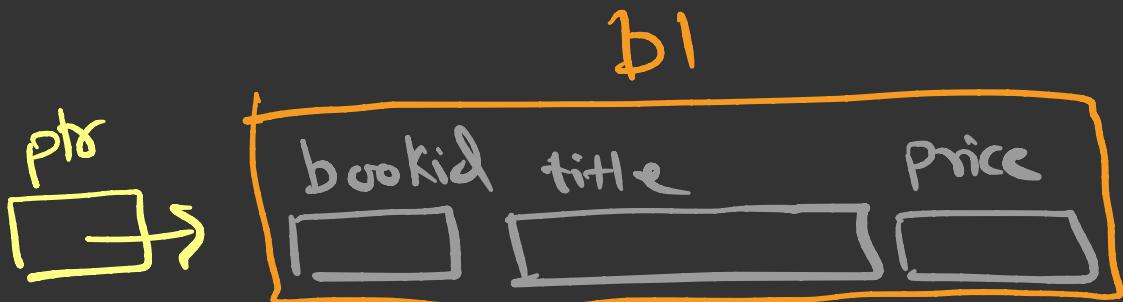
title[std::len(title) - 1] = '\0'

# Structure Pointer

```
struct Book bl;
```

```
struct Book *ptr;
```

```
ptr = &bl;
```



bl.bookid

(\*ptr).bookid

ptr->bookid