STRUCTURE

Thursday, 18. August 2022 14:25

INTRODUCTION TO STRUCTURE

① Structure can be collection of dissimilar elements.

② Structure is a way to group variable.

③ Defining structure is a creating custom datatype

PRIMITIVE AND NON PRIMITIVE DATA TYPES

int	book	custom data type
char	student	secondary data type
float	customer	user defined datatype
double	employee	

DEFINING STRUCTURE (creating data type)

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

- Global v/s local definition

DECLARING STRUCTURE VARIABLE

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

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

}
```

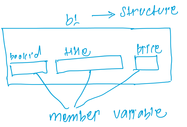
INITIALIZING STRUCTURE VARIABLE DURING DECLARATION

```
void f1()
{
    struct Book b1 = { 1, "CPP", 240 };
}
```

structure variable

Datatype

b1 → structure variable



member variable

INITIALIZING STRUCTURE MEMBER AFTER DECLARATION

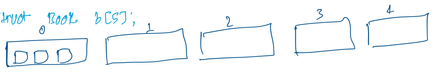
```
struct Book b2;
b2.bookid = 2;
strcpy ( b2.title , "Java");
b2.price = 425.0f ;
```

TAKING INPUT FROM USER

```
struct Book b3;
printf (" Enter book id , title and price");
scanf ( "%d" , &b3.bookid);
fflush ( stdin) ;
fgets ( b2.title , 20 , stdin) ;
scanf ( "%f" , &b2.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( )

S	A	V	A	\n	\0

S	A	V	A	\n

title [strlen (title) - 1] = '\0'

STRUCTURE POINTER

```
struct Book b1 ;
struct Book *ptr ;
ptr = &b1;
```

② b1.bookid

(*ptr) . bookid

ptr → bookid