


Programming Fundamentals

Course Code: CS-111

Course Instructor: Isra Naz

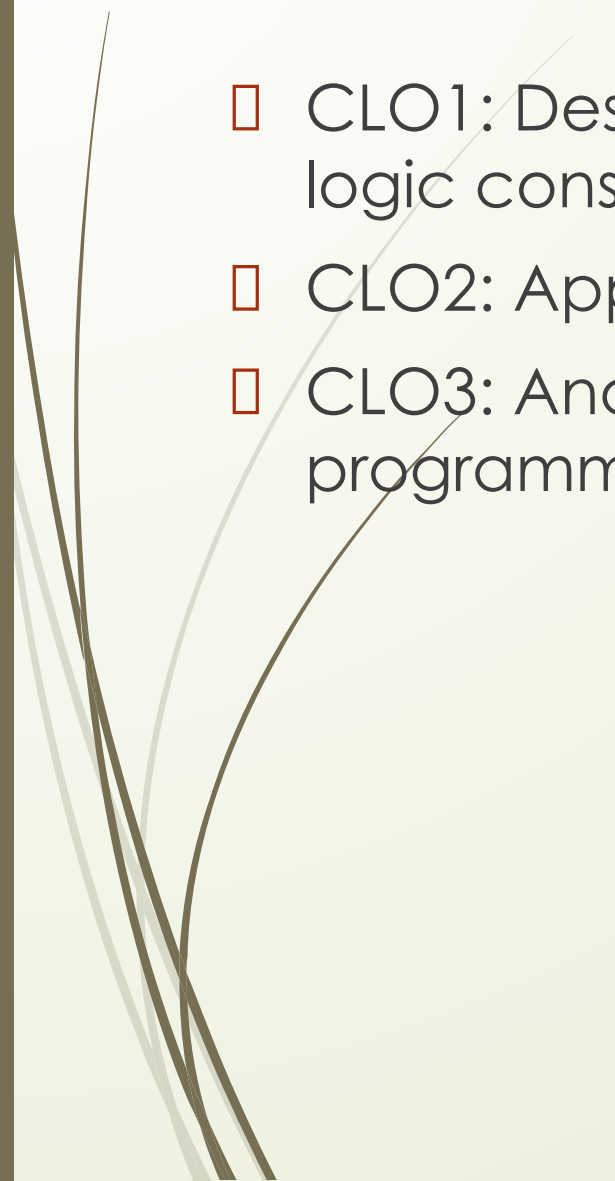


Goals for today:

- Structures
 - Declaring structure
 - Structure variables
 - Accessing structure members
 - Initializing value to structure members
 - Array type structure member
 - Array type structure variable
- 



CLO Covered

- CLO1: Describe fundamental problem-solving techniques and logic constructs. GA 1
 - CLO2: Apply basic programming concepts. GA2
 - CLO3: Analyze and solve the real-world problems by using programming constructs. GA3
- 



Structure

- A Structure is a collection of related data items, possibly of different types that can be referenced with single name.
- Structures are used to define new data types
 - **A new data type that may contain different types of data**
- A simple variable can store only one value at a time, whereas structure variable can store multiple values at the same time.



Structure


- ❑ Elements / data items in a structure are called structure elements, members or fields.
- ❑ A structure is **heterogeneous** in that it can be composed of data of different types.
- ❑ In contrast, array is **homogeneous** since it can contain only data of the same type



Declaring a Structure

Syntax:

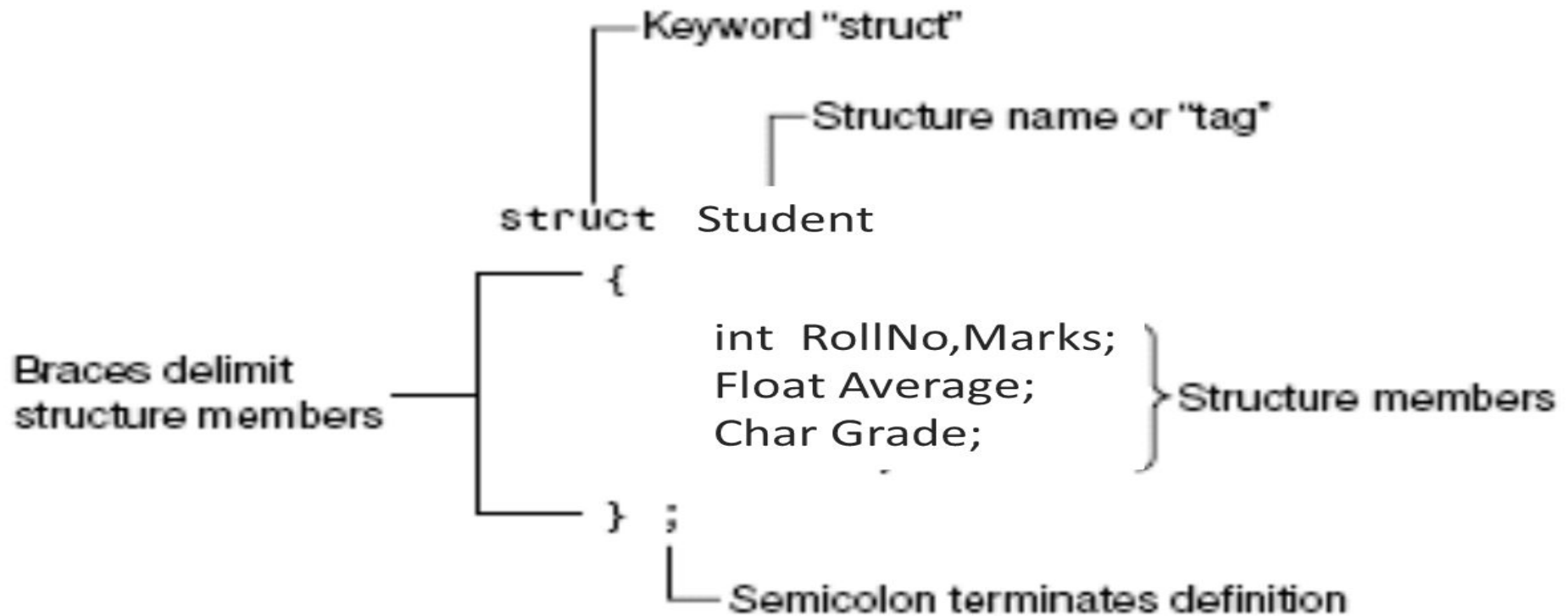
```
struct structure_name
{
    DataType Identifier_a;
    DataType Identifier_b;
    DataType Identifier_c;
};
```



Declaring/ Defining a Structure

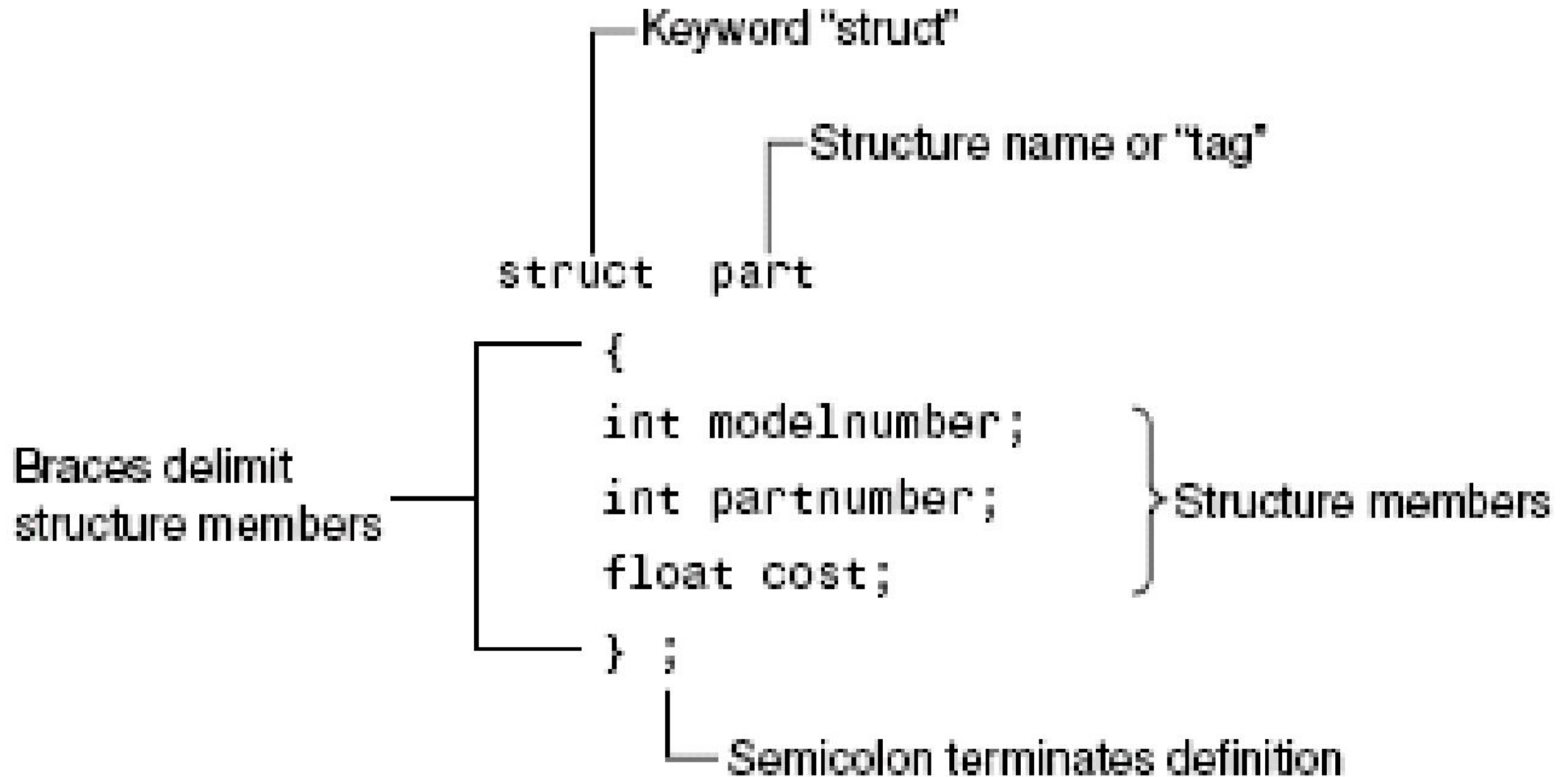
- ❑ The **structure definition** tells the compiler about the detail of the structure and how the structure is organized
 - ❑ It specifies what members the structure will have.
- ❑ A structure definition is merely a specification for how structure variables will look when they are defined
- ❑ A structure is declared using the keyword **struct**.
- ❑ It does not set aside any space in memory or even name any variables.
 - ❑ This is unlike the definition of a simple variable, which does set aside memory.
- ❑ The structure declaration is also called **structure specifier**.

Declaring a Structure



- The above structure contains three member variables.
- **A new data type student is created** in the above declaration.
- A variable of type student can store three variables at one time.

Declaring a Structure



Defining a Structure Variable

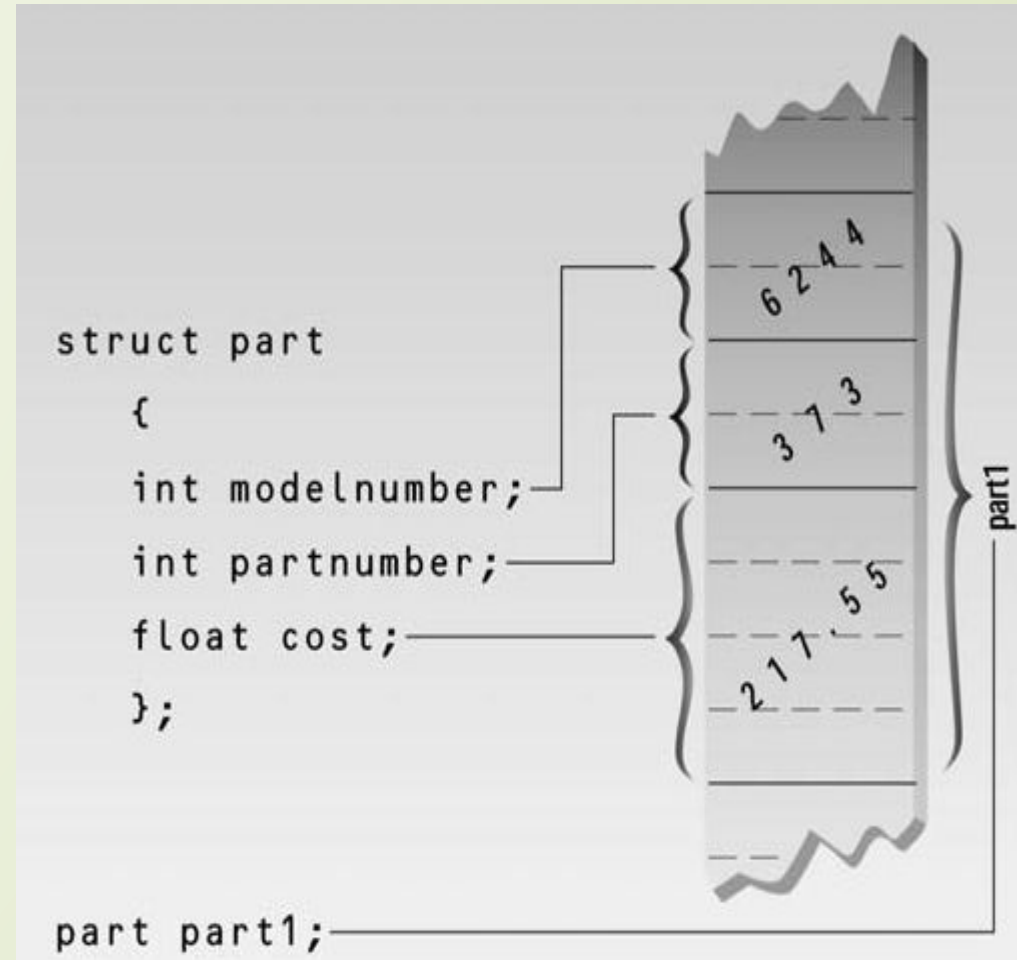
- ❑ When a structure variable is declared, space is reserved in the computer memory to hold all the member of the structure.
- ❑ The structure can be defined inside or outside the main() function
 - ❑ If the structure is defined outside main() it's variable can be declared inside or outside the main()
 - ❑ If the structure is defined inside main() it's variable can only be declared inside the main()

Defining a Structure Variable

- ❑ The structure variable can be defined after the declaration of a structure.
- ❑ The process of defining a structure variable is same as defining a variable of basic types such as int and char.
- ❑ The definition tells the compiler to allocate memory space for the variable.
 - ❑ The compiler automatically allocates sufficient memory according to the elements of the structure.
 - ❑ The memory occupied by the structure variable is equal to the sum of the memory occupied by each member of the structure.

Defining a Structure Variable

- ❑ Structure variable can be declared by giving the structure name and any variable name like so
- ❑ `Structure_name structure variable_name;`



Defining a Structure Variable

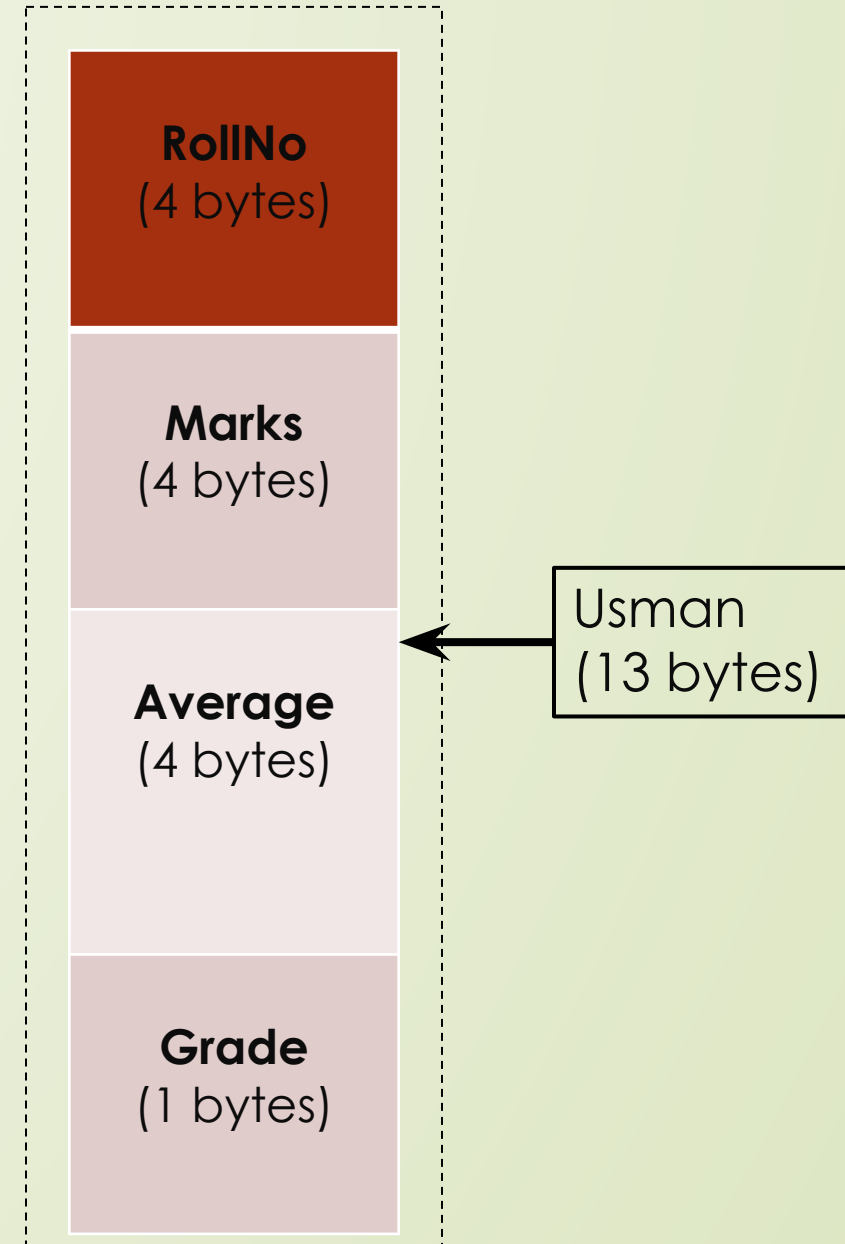
Syntax

Struct_Name Identifier;

Example

```
struct Student
{
    int RollNo, Marks;
    float Average;
    char Grade;
};
```

Student Usman;



Accessing Structure Members

- Once a structure variable has been defined, its members can be accessed using the dot operator
- The structure member is written in three parts:
 - the name of the structure variable
 - The dot operator(.)
 - and the member name
- The dot operator is also called member access operator
- **Syntax:**
struct_variable . struct_memberVariable;



Accessing Structure Members

□ **Example : Accessing members using Assignment Statement**

- Student Usman;
- Usman.RollNo = 10;
- Usman.Marks = 864;
- Usman.Average = 89.52;
- Usman.Grade = 'A';

Accessing Structure Members

□ Example : Accessing members by input value

```
Student Usman;
```

```
cin>>Usman.RollNo;
```

```
cin>>Usman.Marks ;
```

```
cin>>Usman.Average ;
```

```
cin>>Usman.Grade;
```

```
cout<< Usman.RollNo;
```

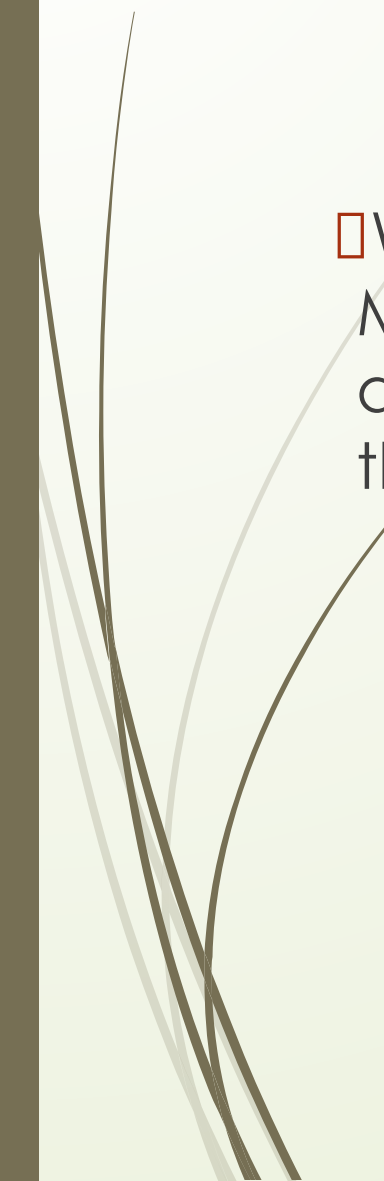
```
cout<< Usman.Marks ;
```

```
cout<<Usman.Average ;
```

```
cout<< Usman.Grade;
```




Example

- Write a program that declares a structure to store Roll No, Marks, Average and Grade of a student. The program should define a structure variable, inputs the values and then displays these values.
- 



Example

```
#include<iostream>
using namespace std;
struct Student //structure declaration
{
    int rno, marks; //structure members
    float avg;
    char grade;
};
int main()
{
    Student s; //defining structure variable
    s.rno=10; //assigning values to structure member
    s.marks=864;
    s.avg=72.5;
    s.grade='A';
    cout<<"RollNo = "<<s.rno<<endl; //display structure members
    cout<<"Marks = "<<s.marks<<endl;
    cout<<"Average = "<<s.avg<<endl;
    cout<<"Grade = "<<s.grade<<endl;
}
```

Initializing Structure Members

- ❑ Structure members can be initialized when the structure variable is defined.
- ❑ The values are written in the same sequence in which they are specified in structure declaration.
- ❑ The values written in braces { } .
- ❑ Each value is separated by comma.

❑ Syntax

- ❑ Struct_Name Identifier = { Value1, Value2,...};

❑ Example

- ❑ Student Usman = { 1, 864, 72.5, 'A' };



Initializing Structure Members

```
#include<iostream>
using namespace std;
struct Student //structure declaration
{
    int rno, marks;//structure members
    float avg;
    char grade;
};
int main()
{
    //initializing values structure variable
    Student s = {10, 864, 72.5, 'A'};
    //display structure members
    cout<<"RollNo = "<<s.rno<<endl;
    cout<<"Marks = "<<s.marks<<endl;
    cout<<"Average = "<<s.avg<<endl;
    cout<<"Grade = "<<s.grade<<endl;
}
```



Example



```
#include <iostream>

Using namespace std;

struct part
{
    int modelnumber;
    int partnumber;
    float cost;
};

void main() {
    part p1;                                //define a structure variable
    p1.modelnumber = 6244;                   //give values to structure members
    p1.partnumber = 373;
    p1.cost = 217.55;
    //display structure members
    cout << "Model " << p1.modelnumber;
    cout << ", part " << p1.partnumber;
    cout << ", costs" << p1.cost << endl;
}
```

Initializing Structure Members

- structure members can be initialized when the structure variable is defined.

```
struct part
{
    int modelnumber;
    int partnumber;
    float cost;
};

void main() {
    part p1 = { 6244, 373, 217.55 };    //initialize the structure variable
    //display structure members
    cout << "Model " << p1.modelnumber;
    cout << ", part " << p1.partnumber;
    cout << ", costs" << p1.cost << endl;
}
```


Reading values from keyboard

```
struct part
```

```
{
```

```
int modelnumber;
```

```
int partnumber;
```

```
float cost;
```

```
};
```

```
void main()
```

```
{
```

```
part p1;
```

```
cout<<"Enter model number";
```

```
cin>>p1.modelnumber;
```

```
cout<<"Enter part number";
```

```
cin>>p1.partnumber;
```

```
cout<<"Enter cost";
```

```
cin>>p1.cost;
```

```
//display structure members
```

```
cout << "Model " << p1.modelnumber;
```

```
cout << ", part " << p1.partnumber;
```

```
cout << ", costs" << p1.cost << endl;
```

```
}
```

Array type member of structure

- Members of the structure can be of different types
 - Can be simple variable, or array variables

- **Example**

```
struct record
{
    char name[15];
    int sub[4];
};
record r ={"Ali" , {10,12,13,14}};
```

* complete the program by taking values from user

Array Type Member of Structure

□ Example

```
struct Student
{
    int RollNo;
    char Name[16];
    int Marks[4];
};
```

```
Student s = { 10 , "Usman" , {35, 45, 25, 22}};
```

Structure Variable as Array

- A structure variable may be declared as array
- This is done to manage large number of records.
- Each variable of the array represents a complete record
- Example

```
struct result
{
    char name[15];
    int id;
    int total;
};
result r[5];
```

* complete the program by taking values from user

□ Example

```
struct Book // structure declaration
```

```
{
```

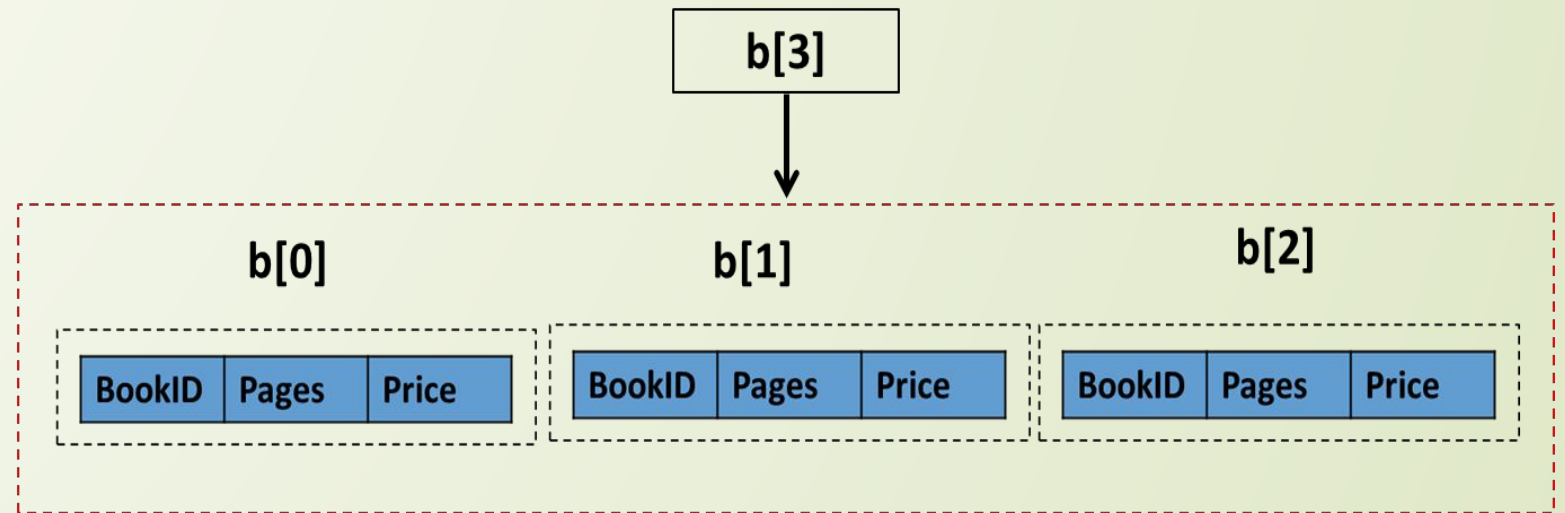
```
    int BookID;
```

```
    int Pages;
```

```
    float Price;
```

```
};
```


```
Book b[3]; // array of structures b[3]
```



The array can store records of three books.



Example

- Write a program that declares a structure to store id, pages and price of a book. It defines an array of structures to store the record of five books. It inputs the records of five books and displays the record of most costly book.
- 

Example

```
#include<iostream>
using namespace std;
struct BookRecord // structure declaration
{
    int id; // structure members
    int pages;
    int price;
};
int main()
{
    BookRecord book[5]; // declaring structure variable as array
    int i, max, m;
    for(i=0; i<=4; i++)
    {
        cout<<"Enter Book ID: ";
        cin>>book[i].id;
        cout<<"Enter number of pages: ";
        cin>>book[i].pages;
        cout<<"Enter price of book: ";
        cin>>book[i].price;
    }
}
```

Example

```
max = book[0].price;
m = 0;
for(i=0;i<=4;i++)
    if(book[i].price>max)
    {
        max=book[i].price;
        m = i;
    }
cout<<"\nThe record of most costly book."<<endl;
cout<<"Book ID: "<<book[m].id<<endl;
cout<<"Pages: "<<book[m].pages<<endl;
cout<<"Price: "<<book[m].price<<endl;
}
```




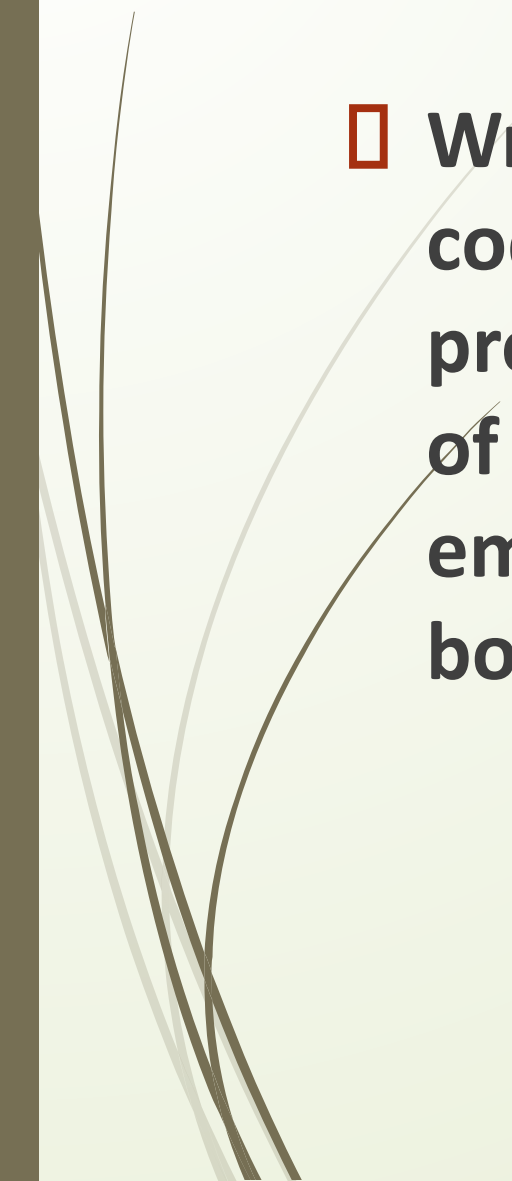
Output

```
Enter Book ID: 15
Enter number of pages: 500
Enter price of book: 350
Enter Book ID: 19
Enter number of pages: 425
Enter price of book: 250
Enter Book ID: 20
Enter number of pages: 600
Enter price of book: 400
Enter Book ID: 21
Enter number of pages: 65
Enter price of book: 25
Enter Book ID: 22
Enter number of pages: 100
Enter price of book: 50

The record of most costly book.
Book ID: 20
Pages: 600
Price: 400
```




Practice Task

-  Write a program that declares structure to store the code number, salary and grade of an employee. The program declares two structure variables, inputs record of two employee and then display the record of the employee with more salary or display the message that both employees have same salary?
- 



Questions...

