JECRC Foundation

JAIPUR ENGINEERING COLLEGE AND RESEARCH CENTRE

JAIPUR ENGINEERING COLLEGE AND RESEARCH CENTRE

Year & Semester - B.Tech I year (I Semester)

Subject - Programming for Problem Solving

Presented by - Ms. Abhilasha /Ms. Yogita Punjabi/

Mr. Gajendra Sharma

Designation - Asst. Professor

Department - Computer Science (First Year)

VISION OF INSTITUTE

To become a renowned centre of outcome based learning, and work towards academic, professional, cultural and social enrichment of the lives of individuals and communities.

MISSION OF INSTITUTE

❖Focus on evaluation of learning outcomes and motivate students to inculcate research aptitude by project based learning.

❖Identify, based on informed perception of Indian, regional and global needs, the areas of focus and provide platform to gain knowledge and solutions.

❖Offer opportunities for interaction between academia and industry.

❖Develop human potential to its fullest extent so that intellectually capable and imaginatively gifted leaders can emerge in a range of professions .

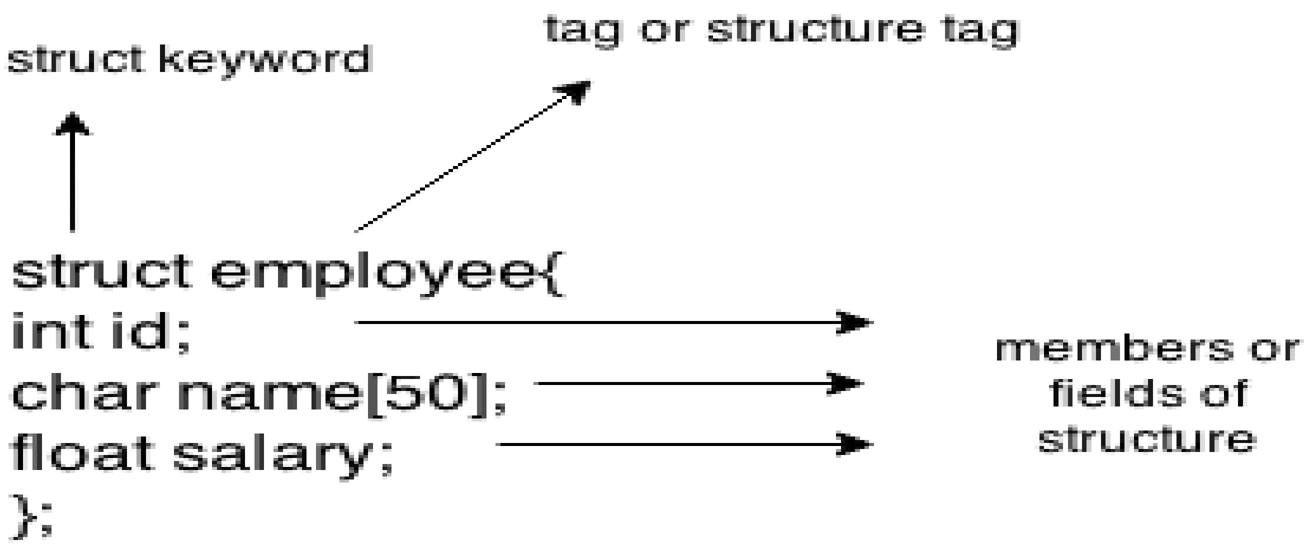
Programming for Problem Solving : Course Outcomes

Students will be able to:
CO1: Understand concept of low-level and high-level languages, primary and secondary memory. Represent algorithm through flowchart and pseudo code for problem solving.
CO2: Represent and convert numbers & alphabets in various notations.
CO3: Analyze and implement decision making statements and looping.
CO4: Apply pointers, memory allocation and data handling through files in 'C' Programming Language.

Structure AND Union

Structures in C

Structure is a group of variables of different data types represented by a single name.



```
struct employee{
int id;
char name[50];
float salary;
};
```

Create struct variables

How to declare variable of a structure?

```
struct  struct_name  var_name;
or
struct struct_name
 {
DataType member1_name;
DataType member2_name;
DataType member3_name;
 …
} var_name;
```

How to access data members of a structure using a struct variable?

```
var_name.member1_name;
 var_name.member2_name; …
```

How to assign values to structure members?

There are three ways to do this.

1) Using Dot(.) operator

var_name.memeber_name = value;

2) All members assigned in one statement

struct struct_name var_name =

{value for memeber1, value for memeber2 …so on for all the members}

3) **Designated initializers** – We will discuss this later at the end of this post.

Program to add two distances (feet-inch)

```
 #include <stdio.h>
struct Distance
 {
 int feet; float inch;
} dist1, dist2, sum;
void main()
{
printf("1st distance\n");
printf("Enter feet: "); scanf("%d", &dist1.feet);
printf("Enter inch: "); scanf("%f", &dist1.inch);
printf("2nd distance\n");
printf("Enter feet: "); scanf("%d", &dist2.feet);
printf("Enter inch: "); scanf("%f", &dist2.inch);
sum.feet = dist1.feet + dist2.feet
sum.inch = dist1.inch + dist2.inch;
while (sum.inch >= 12)
 {
++sum.feet; sum.inch = sum.inch - 12; }
 printf("Sum of distances = %d\'-%.1f\"", sum.feet, sum.inch);
```

Output

```
1st distance Enter feet: 12
Enter inch: 7.9
2nd distance Enter feet: 2
Enter inch: 9.8
Sum of distances = 15'-5.7"
```

C Array of Structures

```
#include<stdio.h>
struct student
{
   char name[20];
   int id;
   float marks;
};
void main()
{
   struct student s1,s2,s3;
   int dummy;
   printf("Enter the name, id, and marks of student 1 ");
   scanf("%s %d %f",s1.name,&s1.id,&s1.marks);
   scanf("%c",&dummy);
   printf("Enter the name, id, and marks of student 2 ");
   scanf("%s %d %f",s2.name,&s2.id,&s2.marks);
   scanf("%c",&dummy);
   printf("Enter the name, id, and marks of student 3 ");
   scanf("%s %d %f",s3.name,&s3.id,&s3.marks);
   scanf("%c",&dummy);
   printf("Printing the details....\n");
   printf("%s %d %f\n",s1.name,s1.id,s1.marks);
   printf("%s %d %f\n",s2.name,s2.id,s2.marks);
   printf("%s %d %f\n",s3.name,s3.id,s3.marks);
}
```

Output

```
Enter the name, id, and marks of
student 1
 James 90 90
Enter the name, id, and marks of
student 2
Adoms 90 90
Enter the name, id, and marks of
student 3
Nick 90 90
Printing the details....
James 90 90.000000
Adoms 90 90.000000
Nick 90 90.000000
```

Keyword typedef

typedef makes the code short and improves readability. In the above discussion we have seen that while using structs every time we have to use the lengthy syntax, which makes the code confusing, lengthy, complex and less readable. The simple solution to this issue is use of typedef. It is like an alias of struct.

Code without typedef

```
struct home_address
{
 int local_street;
char *town;
 char *my_city;
char *my_country;
};
struct home_address var; var.town =
"Agra";
```

Code using tyepdef

```
typedef struct home_address
{
 int local_street;
char *town;
char *my_city;
char *my_country;
 }addr;
..
..
 addr var1;
var.town = "Agra";
```

Nested Structure in C: Struct inside another struct

You can use a structure inside another structure, which is fairly possible.

For example

```
struct complex
 {
 int imag;
 float real;
};
 struct number
 {
struct complex comp;
 int integers;
} num1, num2;
```

Why structs in C?

1. Suppose, you want to store information about a person: his/her name, citizenship number, and salary. You can create different variables name, citNo and salary to store this information.
2. What if you need to store information of more than one person? Now, you need to create different variables for each information per person: name1, citNo1, salary1, name2, citNo2, salary2,etc.
3. A better approach would be to have a collection of all related information **under a single name Person structure and use it for every person.**

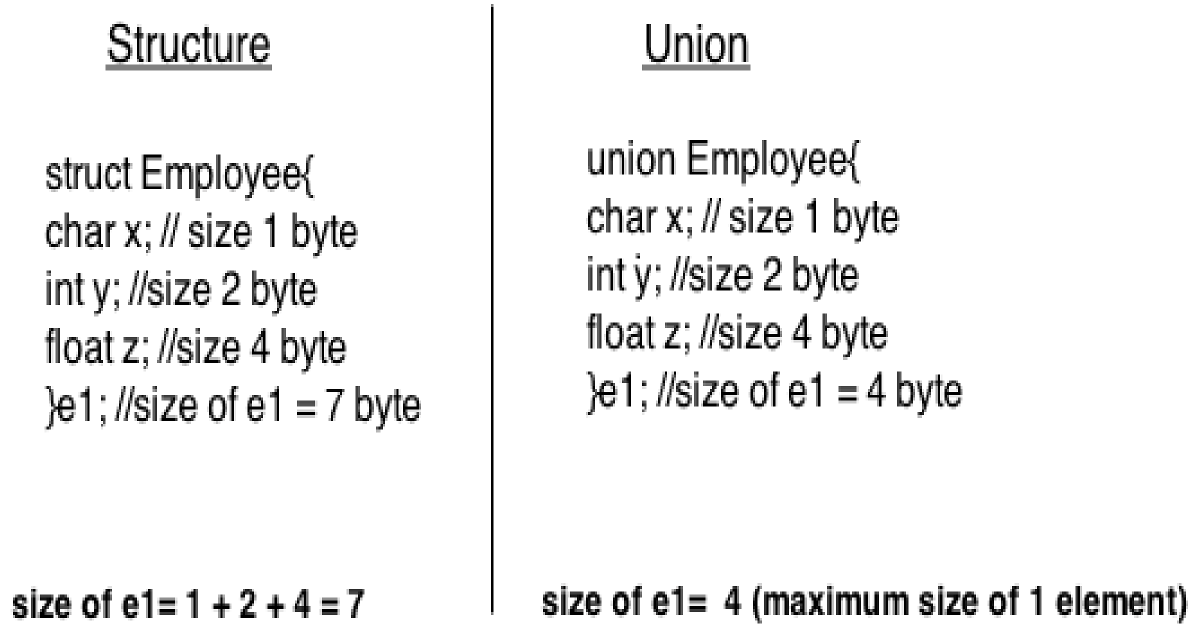
Limitations of C Structures

1. The C structure does not allow the struct data type to be treated like built-in data types:
2. We cannot use operators like +,- etc. on Structure variables.
3. **No Data Hiding:** C Structures do not permit data hiding. Structure members can be accessed by any function, anywhere in the scope of the Structure.
4. **Functions inside Structure:** C structures do not permit functions inside Structure
5. **Static Members:** C Structures cannot have static members inside their body
6. **Access Modifiers:** C Programming language do not support access modifiers. So they cannot be used in C Structures
7. .**Construction creation in Structure:** Structures in C cannot have constructor inside Structures.

Union in C

Like Structure, union is a user defined data type. In union, all members share the same memory location.

Union in c language is used to store the different type of elements.



Structure

```
struct Employee{
char x; // size 1 byte
int y; //size 2 byte
float z; //size 4 byte
}e1; //size of e1 = 7 byte
```

size of e1= 1 + 2 + 4 = 7

Union

```
union Employee{
char x; // size 1 byte
int y; //size 2 byte
float z; //size 4 byte
}e1; //size of e1 = 4 byte
```

size of e1= 4 (maximum size of 1 element)

Advantage & Disadvantage of union over structure

Advantage of union over structure

It **occupies less memory** because it occupies the size of the largest member only.

Disadvantage of union over structure

Only the last entered data can be stored in the union. It overwrites the data previously stored in the union.

Defining union

The **union** keyword is used to define the union. Let's see the syntax to define union in c.

```
union union_name
{
    data_type member1;
    data_type member2;
    .
    .
    data_type memeberN;
};
```

C Union example

```
#include <stdio.h>
#include <string.h>
union employee
{   int id;
    char name[50];
}e1;
void main( )
{
   e1.id=101;
   strcpy(e1.name, “Abhilasha");
  printf( "employee 1 id : %d\n", e1.id);
   printf( "employee 1 name : %s\n", e1.name);
}
```

Output:

```
employee 1 id : 1869508435 employee 1 name
: Abhilasha
```

As you can see, id gets garbage value because name has large memory size. So only name will have actual value.

Bibliography

Programming in ANSI C by E.Balagurusamy, McGrawHill

Let us C Yashavant Kanetkar, BPB Publication

https://www.geeksforgeeks.org/structures-c/

https://www.programiz.com/c-programming/c-structures

https://www.programiz.com/c-programming/c-unions

JECRC Foundation

JAIPUR ENGINEERING COLLEGE AND RESEARCH CENTRE

Thank you!