Department of
Computer Science and Engineering

Title: Structure in C

Structure Programming Lab

CSE 104

Green University of Bangladesh

1 Introduction

Structure is a user-defined datatype in C language which allows us to combine data of different types together. Structure helps to construct a complex data type which is more meaningful. It is somewhat similar to an Array, but an array holds data of similar type only. But structure on the other hand, can store data of any type, which is practical more useful.

1.1 How to define structures?

Before you can create structure variables, you need to define its data type. To define a struct, the struct keyword is used.

1.2 Syntax of struct:

```
struct structureName
{
    dataType member1;
    dataType member2;
    ...
};
```

Here is an example:

```
struct Person
{
    char name[50];
    int number;
    float salary;
};
```

1.3 Declaring Structure Variables

It is possible to declare variables of a structure, either along with structure definition or after the structure is defined. Structure variable declaration is similar to the declaration of any normal variable of any other datatype. Structure variables can be declared in following two ways:

```
struct Student
{
    char name[25];
    int age;
    char branch[10];
    char gender;
};

struct Student S1, S2; //declaring variables of struct Student
```

There are two types of variable:

- Local Variable
- Global Variable

1.4 Example of Local Variable:

```
struct person
{
    int age;
    float salary;
};
```

```c
int main()
{
    struct person variable1,variable2; // local struct variable
}
```

1.5 Example of Global Variable:

```c
struct person
{
    int age;
    float salary;
};
struct person variable1,variable2; // Global struct variable
int main()
{
}
```

In this lab we will discuss about the following types of Structure:

- Structure Type
- Structure Variable
- Nested Structure

1.6 Nested Structure in C: Struct inside another struct

You can use a structure inside another structure, which is fairly possible. As I explained above that once you declared a structure, the struct structname acts as a new data type so you can include it in another struct just like the data type of other data members. Sounds confusing? Don't worry. The following example will clear your doubt. Example of Nested Structure in C Programming Lets say we have two structure like this:
Structure 1: stuaddress

```c
struct stuaddress
{
     int street;
     char *state;
     char *city;
     char *country;
}
```

Structure 2: studata

```c
struct stu_data
{
    int stu_id;
    int stu_age;
    char *stu_name;
    struct stu_address stuAddress;
}
```

As you can see here that I have nested a structure inside another structure. Assignment for struct inside struct (Nested struct) Lets take the example of the two structure that we seen above to understand the logic

```c
struct  studata  mydata;
mydata.stuid = 1001;
mydata.stuage = 30;
mydata.stuAddress.state = "UP"; //Nested struct assignment
..
```

How to access nested structure members? Using chain of "." operator. Suppose you want to display the city alone from nested struct –

```
printf("%s",  mydata.stuAddress.city);
```

2 Structure

A struct in the C programming language is a composite data type declaration that defines a physically grouped list of variables under one name in a block of memory, allowing the different variables to be accessed via a single pointer or by the struct declared name which returns the same address. **Example 1: Create a structure, pass 2 variable and print them using c program.**

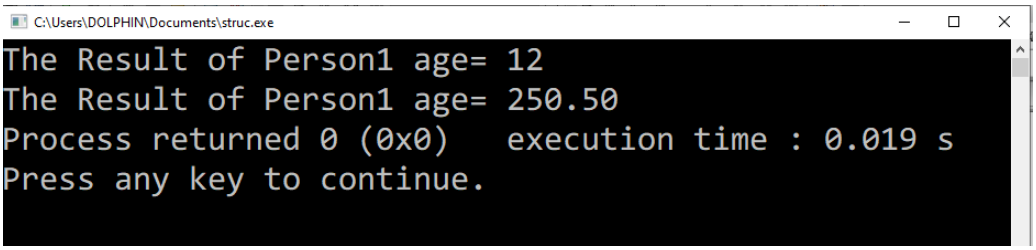
Algorithm 1 Steps in pseudo code:

1: Step1: Start
2: step2: Create structure
3: step3: Take variable by the structure
4: step4: Print them
5: step5: end

Code:

```c
#include<stdio.h>
//global structure
struct person
{
    int age;
    float sallary;
};
int main()
{
    struct person variable1,variable2; // local struct variable
    variable1.age=12;
    variable1.sallary=250.50;
    printf("The Result of Person1 age= %d\n",variable1.age);
    printf("The Result of Person1 age= %.2f",variable1.sallary);
    return 0;
}
```

Output:



Example 2: Take 2 input rectangle length and breadth from user.Find the area of rectangle using structure by C.

Algorithm 2 Steps in pseudo code:

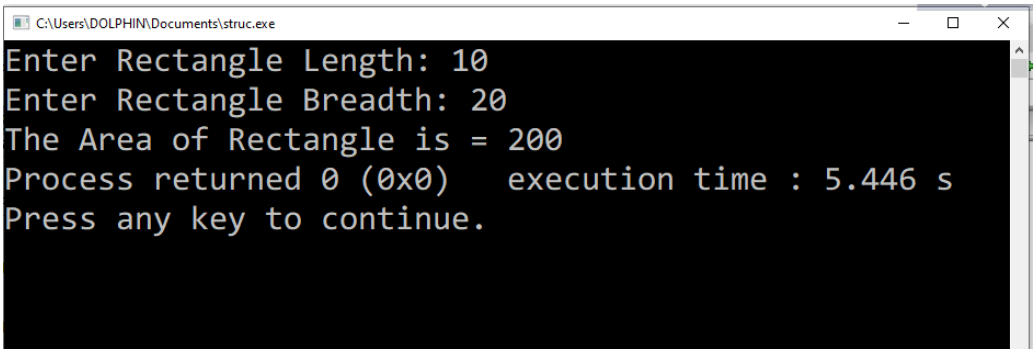
Step1: Start
step2: Input length and breadth
step3: area = length * breadth
step4: print area
step5: end

Code:

```c
#include<stdio.h>
//global structure
struct person
{
    int length;
    int breadth;
};
int main()
{
    struct person variable1,variable2; // local struct variable
    printf("Enter Rectangle Length: ");
    scanf("%d",&variable1.length);
    printf("Enter Rectangle Breadth: ");
    scanf("%d",&variable1.breadth);
    printf("The Area of Rectangle is = %d",variable1.length*variable1.breadth);

    return 0;
}
```

Output:



3 Structure Variable

It is possible to declare variables of a structure,either along with structure definition or after the structure is defined. Structure variable declaration is similar to the declaration of any normal variable of any other datatype.

Example 1: Create a structure and declare 2 local variable and check it summation even or odd. Code:

Algorithm 3 Steps in pseudo code:

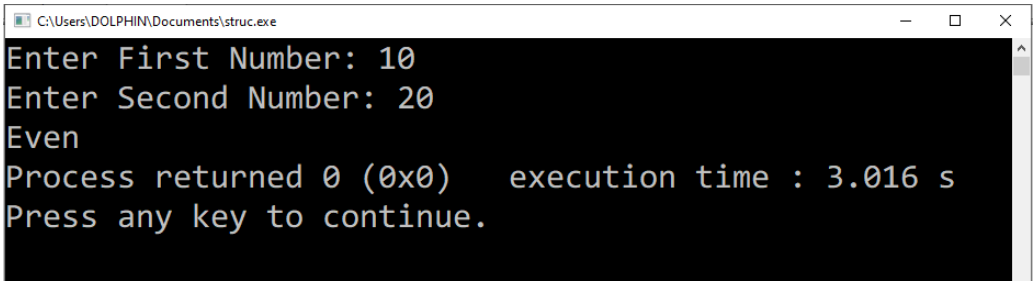
Step1: Start
Step2:Input 2 number
Step3:Add them
Step4:if summation
Step5:Else print Odd
Step6: End

```c
#include<stdio.h>
//global structure
struct person
{
    int a;
    int b;
    int result;
};
struct person variable1,variable2; // Global struct variable
int main()
{

    printf("Enter First Number: ");
    scanf("%d",&variable1.a);
    printf("Enter Second Number: ");
    scanf("%d",&variable1.b);
    variable2.result=variable1.a+variable1.b;
    if(variable2.result%2==0)
        printf("Even");
    else
        printf("Odd");

    return 0;
}
```

Output:



Example 2: Input an array size and take input from user and print them using structure C programming.

Algorithm 4 Steps in pseudo code:

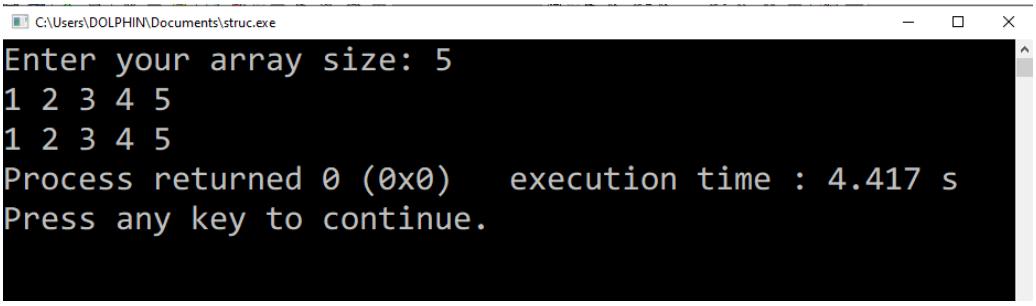
Step1: Start
step2: Input n
step3: for 1 to n
step4: Input element and store in array step5: print array
step6: end

Code:

```c
#include<stdio.h>
//global structure
struct person
{
    int arr[200];
    int n;
};
struct person variable1,variable2; // Global struct variable
int main()
{
    printf("Enter your array size: ");
    scanf("%d",&variable1.n);
    for(int i=0;i<variable1.n;i++)
    {
        scanf("%d",&variable1.arr[i]);
    }
    for(int i=0;i<variable1.n;i++)
    {
        printf("%d ",variable1.arr[i]);
    }


    return 0;
}
```

Output:



4 Nested Structure

Nested structure in C is nothing but structure within structure. One structure can be declared inside other structure as we declare structure members inside a structure. The structure variables can be a normal structure variable or a pointer variable to access the data.

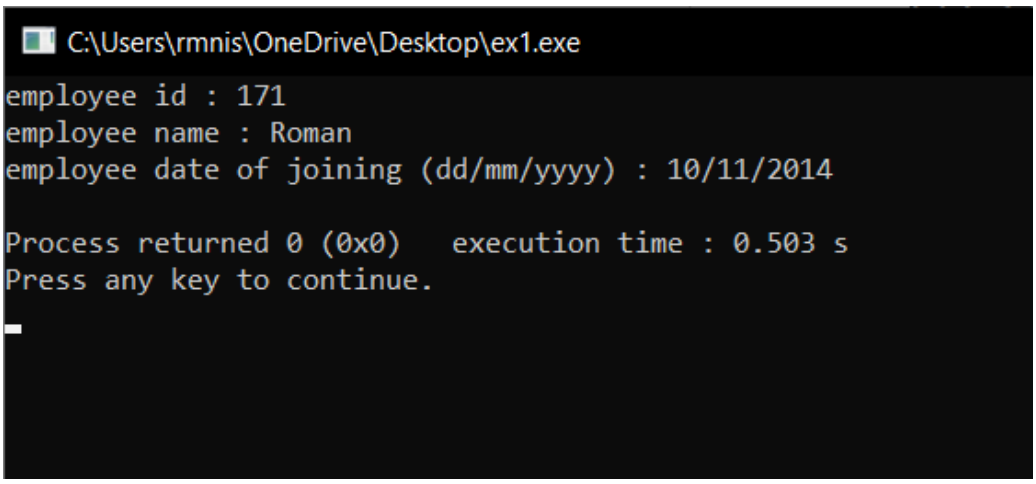
Example 1:Print your name and joining date using nested structure using C

Code:

```c
#include <stdio.h>
#include <string.h>
struct Employee
{
    int id;
    char name[20];
    struct Date
     {
       int dd;
       int mm;
       int yyyy;
     }doj;
}e1;
int main( )
{
    //storing employee information
    e1.id=171;
    strcpy(e1.name, "Roman");//copying string into char array
    e1.doj.dd=10;
    e1.doj.mm=11;
    e1.doj.yyyy=2014;

    //printing first employee information
    printf( "employee id : %d\n", e1.id);
    printf( "employee name : %s\n", e1.name);
    printf( "employee date of joining (dd/mm/yyyy) : %d/%d/%d\n", e1.doj.dd,e1.doj.mm,e1.doj.
     yyyy);
    return 0;
}
```

Output:



Example 2:
Code:

```c
#include<stdio.h>

/* Declaration of structure */
struct address
{
 int houseno;
 char street[20];
 int stateno;
};

/* Declaration of structure */
```

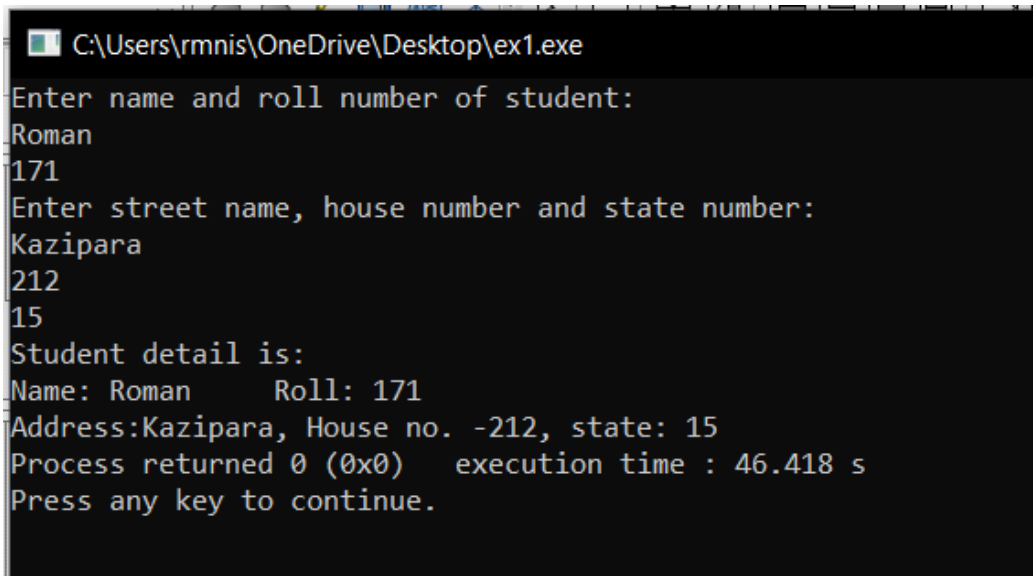
```c
struct student
{
 char name[30];
 int roll;
 struct address adrs;   /* Nested structure */
};

int main()
{
 struct student stud;

 printf("Enter name and roll number of student:\n");
 scanf("%s%d",stud.name, &stud.roll);
 printf("Enter street name, house number and state number:\n");
 scanf("%s%d%d",stud.adrs.street, &stud.adrs.houseno, &stud.adrs.stateno);
 printf("Student detail is:\n");
 printf("Name: %s\tRoll: %d\n", stud.name, stud.roll);
 printf("Address:%s, House no. -%d, state: %d",stud.adrs.street, stud.adrs.houseno, stud.adrs.
    stateno);

 return 0;
}
```

Output:



5 Discussion & Conclusion

Based on the focused objective(s) to understand about the stack operations, the additional lab exercise made me more confident towards the fulfilment of the objectives(s).

6 Lab Task (Please implement yourself and show the output to the instructor)

1. Write a C program to print your name, date of birth and mobile number.

2. Write a C program to calculate area of circle

3. Write a C program to calculate sum of 1+2+3+....+N.

4. Write a C program to check number is prime or not

5. Write a C program for input 3 numbers and print greatest number

6. Write a C program to print 1 to n numbers

7 Lab Exercise (Submit as a report)

- Write a C program to check a number is positive or negative.
- Write a C program for input N numbers in array and print the summation of all value in array.
- Write a C program to calculate Celsius to Fahrenheit
- Write a C Program to take input N size array and print reverse array.
- Write a C program to check a character is vowel or not.
- Write a C program using nested structure to print your name and address with location.

8 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.