

# Structures

Er. Shiva K. Shrestha (HoD)

Department of Computer Engineering,  
Khwopa College of Engineering



# Overview

- ▶ Defining a Structure
- ▶ Structure Initialization
- ▶ Accessing Structure Member
- ▶ How Structure Elements are Stored?
- ▶ Comparing of Array & Structure
- ▶ Array of Structure
- ▶ Initializing Array of Structure
- ▶ Nested Structure
- ▶ Pointers to Structure
- ▶ Function & Structure
  - ▶ Passing structure members to a function
  - ▶ Passing an entire structure to a function
  - ▶ Passing structure pointer to a function

# Structure Definition

## (Structure Template Declaration)

```
struct tag_name{  
    data_type member_1;  
    data_type member_2;  
    ...           ...  
    data_type member_n;  
};
```

### **Example:**

```
struct student{  
    char name[50];  
    int roll;  
    float marks;  
    char gender;  
};
```

A structure is a collection of one or more than one variable, possibly of different data type, grouped together under a single name for convenient handling.

# Example# 1

4

```
C8_1.C
1  #include<stdio.h>
2  #include<conio.h>
3  struct account{
4      ....int acc_no;
5      ....char acc_type;
6      ....char name[80];
7      float balance;
8  };
9  struct account cust;
10 int main(){
11     printf("Enter Name: \n");
12     //scanf("%s", cust.name);
13     gets(cust.name);
14     printf("\nName = %s", cust.name);
15     getch();
16     return 0;
17 }
```

C:\Users\ErSKS\Google Drive

Enter Name:  
Arpan Sharma

Name = Arpan Sharma\_

Er. Shiva K. Shrestha (HoD, Computer Department)  
10/3/2018

# Task

5

- ▶ WAP to read roll number & name of a student from console & display it after reading both parameters.
- ▶ WAP to read roll number & name of a student from console & display it after reading both parameters.  
Hint:- Use structure

# Declaration of a Structure variable

- ▶ `[storage_class] struct tag variable_1, variable_2, ..., variable_n;`

```
C8_2.c
1  #include<stdio.h>
2  #include<conio.h>
3  struct{
4      int acc_no;
5      char acc_type;
6      char name[80];
7      float balance;
8  }cust;
9  int main(){
10     printf("Enter Name: ");
11     //scanf("%s", cust.name);
12     gets(cust.name);
13     printf("\nName = %s", cust.name);
14     getch();
15     return 0;
16 }
```

```
struct account{
    int acc_no;
    char acc_type;
    char name[80];
    float balance;
};
struct account old_customer,
new_customer;
```

or

```
struct account{
    int acc_no;
    char acc_type;
    char name[80];
    float balance;
}old_customer, new_customer;
```

# Accessing Structure Member

Structure use a dot(.) operator to access individual elements.

▶ `structure_name.member_name`

- ▶ `printf("Book Name = %s", b1.name); // C Programming`
- ▶ `printf("Book Name = %f", b1.price); // 130.00`
- ▶ `printf("Book Name = %d", b1.pages); // 550`
- ▶ `printf("Book Name = %s", b2.name); // ?`
- ▶ `printf("Book Name = %f", b2.price); // ?`
- ▶ `printf("Book Name = %d", b2.pages); // ?`



# Structure Initialization

- ▶ Like primary variables and arrays, structure variables can also be initialized where they are declared. The format used is quite similar to that used to initialize arrays.

```
C8_3.C
1  #include<stdio.h>
2  #include<conio.h>
3  struct book{
4      char name[20];
5      float price;
6      int pages;
7  };
8  struct book b1={"Basic C",130.00,550};
9  struct book b2={"Math-I",150.50,800};
10
11 int main(){
12     printf("Book Details:\nName\tPrice\tPages\n");
13     printf("%s\t%.2f\t%d", b1.name, b1.price, b1.pages);
14     printf("\n%s\t%.2f\t%d", b2.name, b2.price, b2.pages);
15     getch();
16     return 0;
17 }
```

C:\Users\ErSKS\Google Drive

Book Details:

Name	Price	Pages
Basic C	130.00	550
Math-I	150.50	800



s.roll	s.marks	s.gender
25	88.00	'M'
170	172	176

# How Structure Elements are Stored?

```

File Edit Search Run Compile Debug Project Options Window Help
C8_3_2.C 1=[↑]
#include<stdio.h>
#include<conio.h>
struct student{
    int roll;
    float marks;
    char gender;
};
struct student s={25, 88.00, 'M'};

int main(){
    clrscr();
    printf("Address of roll = %u, ", &s.roll);
    printf("sizeof(roll) = %d\n", sizeof(int));
    printf("Address of marks = %u, ", &s.marks);
    printf("sizeof(marks) = %d\n", sizeof(float));
    printf("Address of gender = %u, ", &s.gender);
    printf("sizeof(gender) = %d\n", sizeof(char));
    getch();
    return 0;
}
16:37
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```

DOSBox 0.74, Cpu speed: max 100% cycles, Frame  
Address of roll = 170, sizeof(roll) = 2  
Address of marks = 172, sizeof(marks) = 4  
Address of gender = 176, sizeof(gender) = 1

# WAP to enter name, price & pages of 3 books & display entered information.

10

```
C8_4.C
1  #include<stdio.h>
2  #include<conio.h>
3  struct book{
4      char name[20];
5      float price;
6      int pages;
7  };
8  struct book b1,b2,b3;
9  int main(){
10     printf("Enter Name, Price & No. of Pages of 3 Books:\n");
11     scanf("%s%f%d",b1.name,&b1.price,&b1.pages);
12     scanf("%s%f%d",b2.name,&b2.price,&b2.pages);
13     scanf("%s%f%d",b3.name,&b3.price,&b3.pages);
14
15     printf("\nThis is what you entered");
16     printf("\n%s\t%.2f\t%d",b1.name,b1.price,b1.pages);
17     printf("\n%s\t%.2f\t%d",b2.name,b2.price,b2.pages);
18     printf("\n%s\t%.2f\t%d",b3.name,b3.price,b3.pages);
19     getch();
20     return 0;
21 }
```

```
C:\Users\ErSKS\Google Drive (c.khwopa@gmail.com)\C_
Enter Name, Price & No. of Pages of 3 Books:
C-Programming    600    542
Learning-C       650    422
Let-Us-C         336    748

This is what you entered
C-Programming    600.00   542
Learning-C       650.00   422
Let-Us-C         336.00   748_
```

# Array vs. Structure

## Array

- ▶ Collection of similar data types
- ▶ Derived data type
- ▶ Subscript/index is used to access the member of an array.
- ▶ Array behaves like a built-in data types only we need to declare to it.
- ▶ Example: `int a[5];`

## Structure

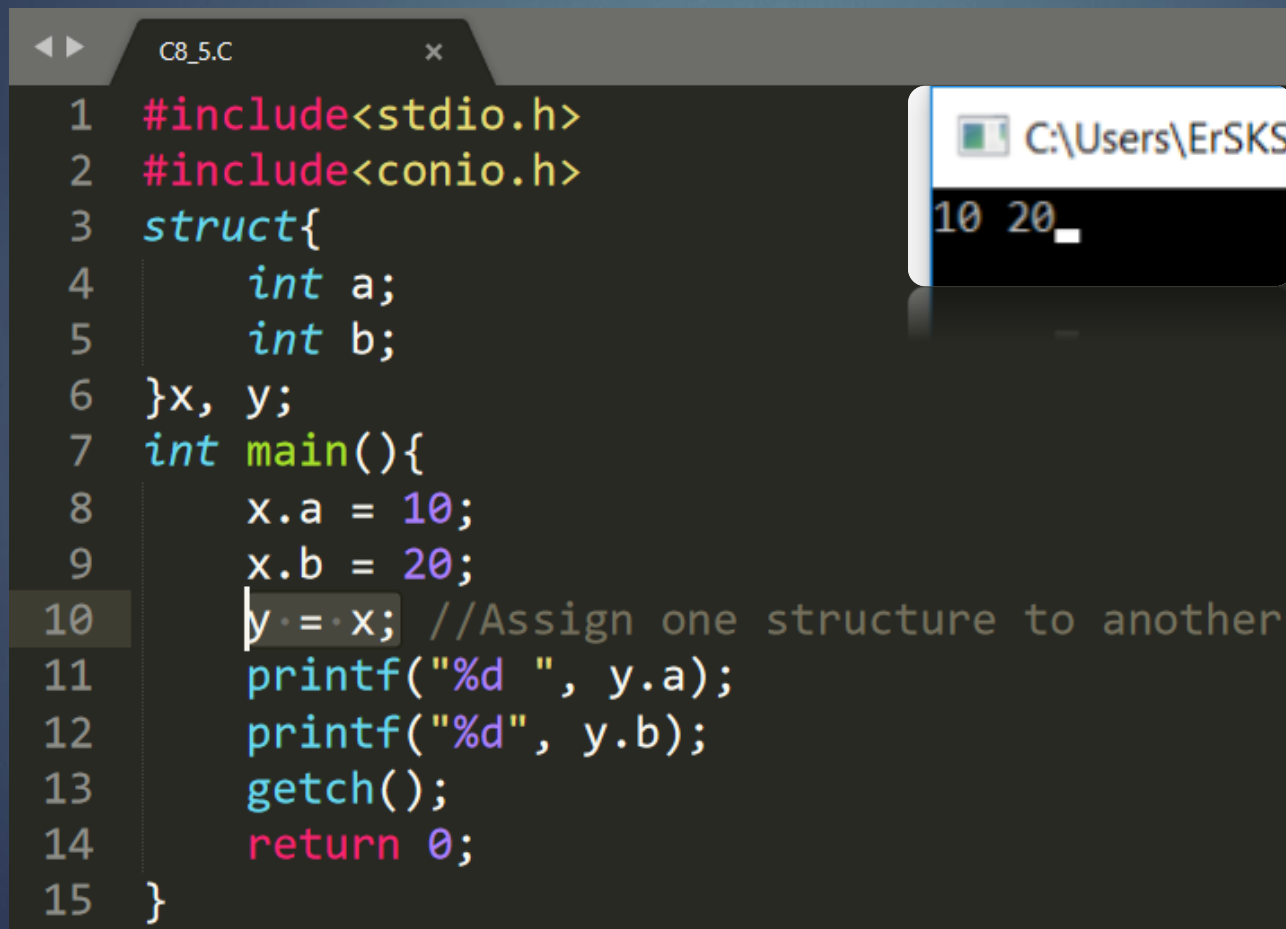
- ▶ Collection of dissimilar data types
- ▶ User-defined data type
- ▶ Dot operator is used to access the member of a structure.
- ▶ First need to design & declare structure before the variable of that type are declared & used
- ▶ Example:

```
struct num{  
    int a; float b; char c;  
};
```

# Structure Assignment

12

- ▶ The information contained in one structure may be assigned to another structure of the same type using a single assignment statement. That is, you do not need to assign the value of each member separately.



The screenshot shows a code editor window titled 'C8\_5.C' with a dark theme. The code defines a structure with two integer members, 'a' and 'b', and declares two variables 'x' and 'y' of that type. In the 'main' function, 'x.a' is assigned 10 and 'x.b' is assigned 20. Line 10, which is highlighted, shows 'y = x;' followed by a comment '//Assign one structure to another'. Lines 11 and 12 use 'printf' to output the values of 'y.a' and 'y.b'. Line 13 calls 'getch()' and line 14 returns 0. A small terminal window is overlaid on the right side of the code editor, showing the output '10 20'.

```
1  #include<stdio.h>
2  #include<conio.h>
3  struct{
4      int a;
5      int b;
6  }x, y;
7  int main(){
8      x.a = 10;
9      x.b = 20;
10     y = x; //Assign one structure to another
11     printf("%d ", y.a);
12     printf("%d", y.b);
13     getch();
14     return 0;
15 }
```

# Nested Structure

```
C8_6.C x
1 #include<stdio.h>
2 #include<conio.h>
3 struct employee{
4     char name[25];
5     long int emp_code;
6     char post[20];
7     float salary;
8     struct emp_address{
9         int no;
10        char street[25];
11        char area[20];
12    }adr1;
13 }e1 = {"Er. Shiva K. Shrestha",74801,"HoD",65000.00,8,"Garud Kundal Road","Libali"};
14
15 int main(){
16     printf("Employee Adress Area = %s", e1.adr1.area);
17     getch();
18     return 0;
19 }
```

C:\Users\ErSKS\Google Drive (c.khwopa@gmail.com)  
Employee Adress Area = Libali

**struct** employee{  
    char name[25];  
    long int emp\_code;  
    char post[20];  
    float salary;  
} emp1, emp2;

**struct** emp\_address{  
    int no;  
    char street[25];  
    char area[20];  
} address1;

- ▶ Two structures left figure can be consolidated as shown in program above

# Example#7: We can also tag names to define inner structures

```
struct date{
    int month;
    int day;
    int year;
};

struct account{
    int acc_no;
    char acc_type;
    char name[80];
    float balance;
    struct date last_payment;
}customers;
```

It is also permissible to nest more than one type of structures.

```
struct personal_record{
    struct name_part n;
    struct address_part a;
    struct date_part dob;
    .....
};

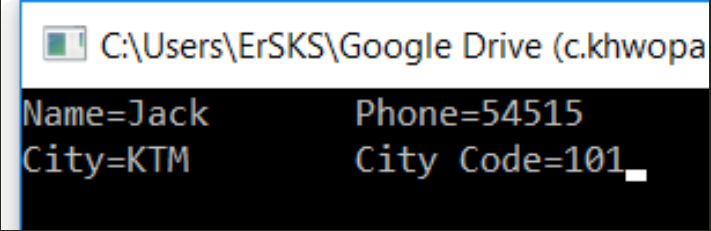
struct personal_record p;
```

If a structure member is itself a structure, then a member of the embedded structure can be accessed by writing variable.member.sub\_member

e.g. **customers**.last\_payment.month

# Example#8: A sample to demonstrate nesting of structure

```
C8_8.C
1  #include<stdio.h>
2  #include<conio.h>
3  struct address{
4      char phone[15];
5      char city[25];
6      int citycode;
7  };
8  struct emp{
9      char name[25];
10     struct address a;
11 };
12 int main(){
13     int i;
14     struct emp e={"Jack","54515","KTM",101};
15     printf("Name=%s\tPhone=%s",e.name,e.a.phone);
16     printf("\nCity=%s\tCity Code=%d",e.a.city,e.a.citycode);
17     getch();
18     return 0;
19 }
```



```
C:\Users\ErSKS\Google Drive (c.khwopa)
Name=Jack      Phone=54515
City=KTM       City Code=101
```





```
1  #include <stdio.h>
2  #include <conio.h>
3  struct add{
4      int door_no;
5      char street[20];
6      char place[30];
7  };
8  struct student{
9      char name[30];
10     int roll_no;
11     struct add address;
12 };
13 int main(){
14     struct student std;
15     printf("Enter Your Details:\n");
16     printf("Name\tRoll No.\n");
17     scanf("%s%d",std.name,&std.roll_no);
18     printf("Class Room No.\tStreet\tPlace\n");
19     scanf("%d%s%s",&std.address.door_no,std.address.street,std.address.place);
20     printf("\nYour details are ...\n");
21     printf("Name : %s",std.name);
22     printf("\nRoll No. is %d",std.roll_no);
23     printf("\nClass Room No. is %d",std.address.door_no);
24     printf("\nArea name is %s",std.address.place);
25     printf("\nStreet name is %s",std.address.street);
26     getch();
27     return 0;
28 }
```

Enter Your Details:

Name      Roll No.

Niranjan 26

Class Room No.    Street    Place

306                      Libali    BakhunchheTole

Your details are ...

Name : Niranjan

Roll No. is 26

Class Room No. is 306

Area name is BakhunchheTole

Street name is Libali\_

# Array of Structures

17

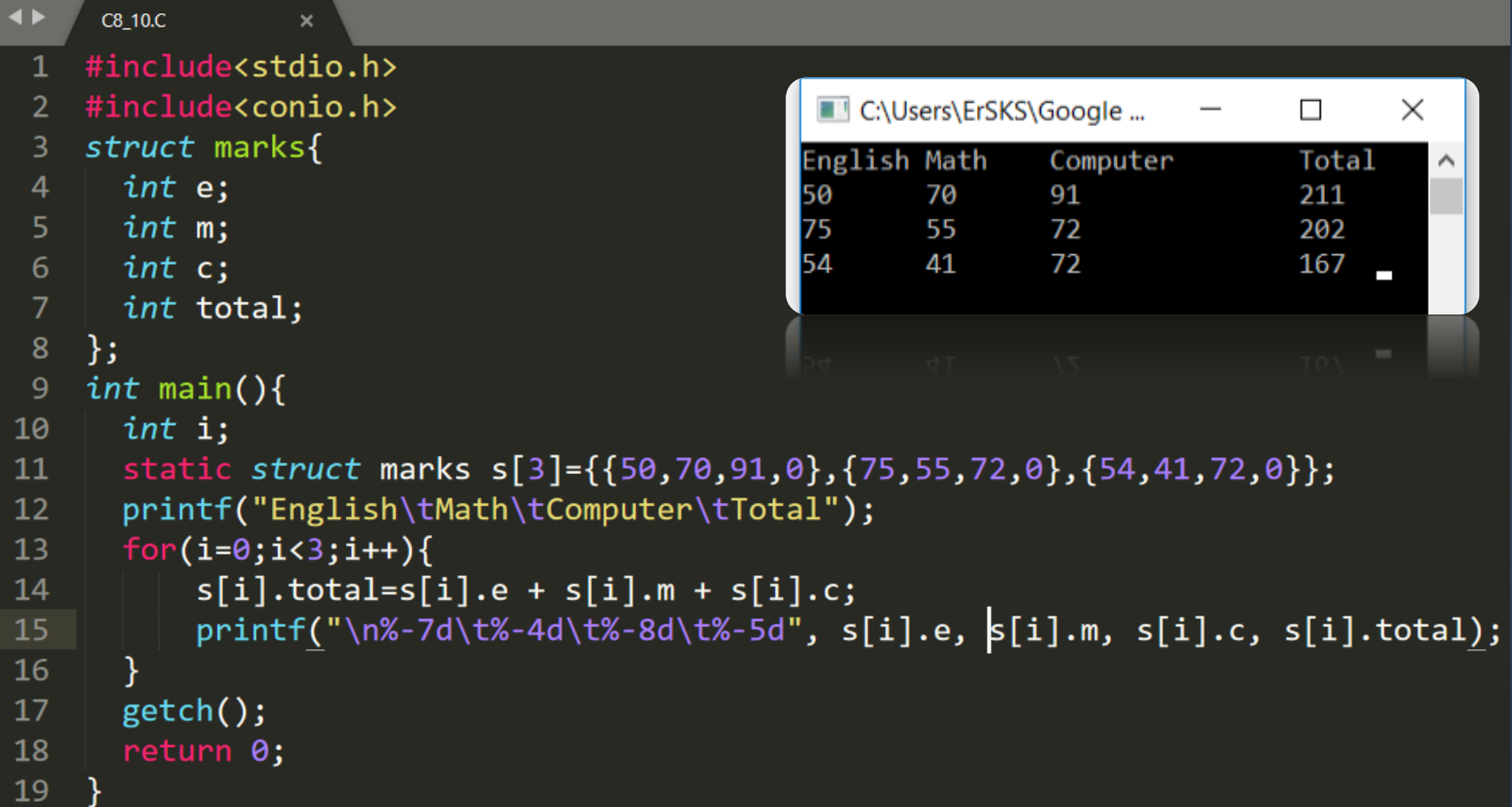
To declare a 100-element array of structures of type books:

```
struct book b[100];
```

To print the price of structure 3, write

```
printf("%f",b[2].price);
```

Er. Shiva K.  
10/3/2018



The screenshot shows a C program in a code editor and its output in a separate window. The code defines a structure 'marks' with four integer fields: 'e', 'm', 'c', and 'total'. It then declares an array 's' of three 'marks' structures and initializes them with specific values. The program calculates the total for each structure and prints the results in a table format.

```
1 #include<stdio.h>
2 #include<conio.h>
3 struct marks{
4     int e;
5     int m;
6     int c;
7     int total;
8 };
9 int main(){
10     int i;
11     static struct marks s[3]={{50,70,91,0},{75,55,72,0},{54,41,72,0}};
12     printf("English\tMath\tComputer\tTotal");
13     for(i=0;i<3;i++){
14         s[i].total=s[i].e + s[i].m + s[i].c;
15         printf("\n%-7d\t%-4d\t%-8d\t%-5d", s[i].e, s[i].m, s[i].c, s[i].total);
16     }
17     getch();
18     return 0;
19 }
```

The output window displays the following table:

English	Math	Computer	Total
50	70	91	211
75	55	72	202
54	41	72	167

1. WAP to use nested structure to hold information of student name, address, roll & date of birth within another structure. Ask user to enter information & display respective information.
2. WAP to read name, address, roll, year, month & day of 3 students using structure. And display that information using user-defined function.
3. WAP using nested structure to read info. Of 3 students. Use concept of array of structure to hold their data. Display their info from main function.

# Example# 11: Storing book information using structure

19

Er. Shiva K. Shrestha (HoD, Computer Department)  
10/3/2018

```
C8_11.C x
1 #include<stdio.h>
2 #include<conio.h>
3 struct book{
4     char name[20];
5     int price;
6     int pages;
7 }b[5];
8
9 int main(){
10     int i;
11     for(i=0;i<5;i++){
12         printf("%d. Enter Name, Price, and Pages: ",(i+1));
13         scanf("%s%d%d", &b[i].name, &b[i].price, &b[i].pages);
14     }
15     printf("\n%-15s\tPrice\tPages","Book Name");
16     for(i=0;i<5;i++){
17         printf("\n%-15s\t%d\t%d", b[i].name, b[i].price, b[i].pages);
18     }
19     getch();
20     return 0;
21 }
```

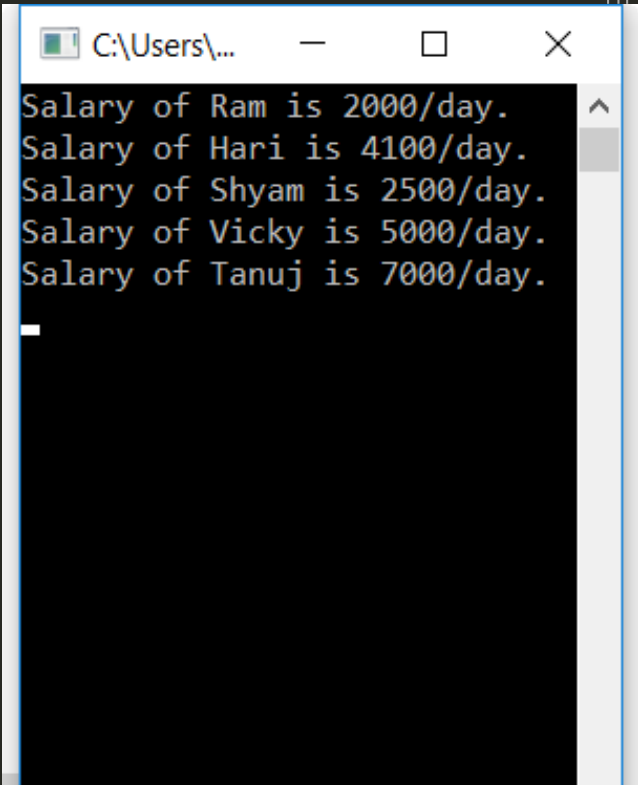
```
C:\Users\ErSKS\Google Drive (c.khwopa@gmail.com)\C_Codes\Ch8\C8_11.exe
1. Enter Name, Price, and Pages: C 650 650
2. Enter Name, Price, and Pages: Math-I 452 350
3. Enter Name, Price, and Pages: Drawing-I 263 300
4. Enter Name, Price, and Pages: Applied 485 600
5. Enter Name, Price, and Pages: BasicElectrical 378 450

Book Name      Price    Pages
C               650      650
Math-I          452      350
Drawing-I       263      300
Applied         485      600
BasicElectrical 378      450_
```

# Initializing Array of Structure

20

```
C8_12.C
1  #include<stdio.h>
2  #include<conio.h>
3  struct employee{
4      char name[20];
5      char code[5];
6      int salary;
7  } emp [5] = {
8      {"Ram", "E01", 2000},
9      {"Hari", "E02", 4100},
10     {"Shyam", "E03", 2500},
11     {"Vicky", "E04", 5000},
12     {"Tanuj", "E05", 7000},
13 };
14
15 int main(){
16     int i;
17     for(i=0;i<5;i++){
18         printf("Salary of %s is %d/day.\n", emp[i].name, emp[i].salary);
19     }
20     getch();
21     return 0;
22 }
```



Salary of Ram is 2000/day.  
Salary of Hari is 4100/day.  
Salary of Shyam is 2500/day.  
Salary of Vicky is 5000/day.  
Salary of Tanuj is 7000/day.

# Typedef Structure

- ▶ C language provides the opportunity to define new data\_type equivalent to the existing system using the typedef statement. The declaration would be

```
typedef struct date{  
    int dd;  
    int mm;  
    int yyyy;  
}dob;  
dob emp_date_of_birth;
```

- ▶ The above declaration variable `emp_date_of_birth` has now become the type of *structure date*.

# Function & Structure

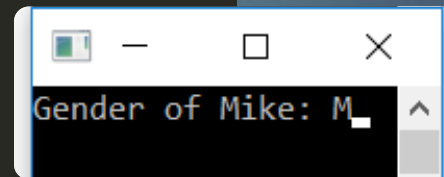
- ▶ Passing structure members to a function
- ▶ Passing an entire structure to a function
- ▶ Passing structure pointer to a function



# Passing structure members to a function

23

```
C8_13.C
1  #include<stdio.h>
2  #include<conio.h>
3  struct{
4      char g;
5      int y;
6      float z;
7      char s[10];
8  }mike;
9  void func(char);
10
11 int main(){
12     mike.g='M';
13     func(mike.g);
14     getch();
15     return 0;
16 }
17
18 void func(char a){
19     printf("Gender of Mike: %c", a);
20 }
```

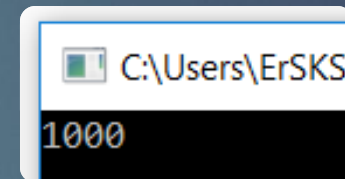


Gender of Mike: M

# Passing an entire structure to a function

24

```
C8_14.C x
1  #include<stdio.h>
2  #include<conio.h>
3  struct test{
4      int a, b;
5      char ch;
6  };
7  void function(struct test);
8
9  int main(){
10     struct test arg;
11     arg.a=1000;
12     function(arg);
13     getch();
14     return 0;
15 }
16
17 void function(struct test param) {
18     printf("%d",param.a);
19 }
```



# Task

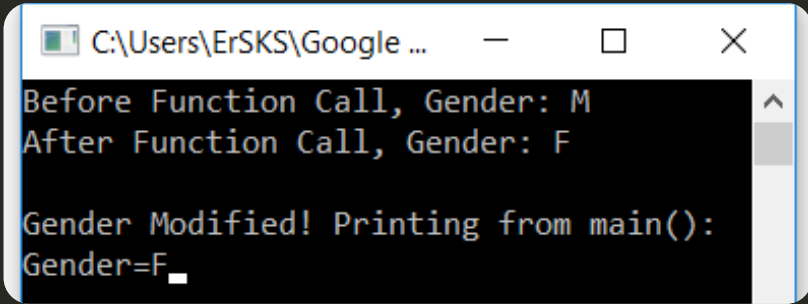
25

- ▶ WAP to set elements of “test” structure to 100, 1000, ‘A’ and display all members by passing structure to a function.

# Passing structure pointer to a function

26

```
C8_15.C
1  #include<stdio.h>
2  #include<conio.h>
3  struct{
4      char g;
5      int y;
6  }bishnu;
7  void func(char *gen);
8
9  int main(){
10     bishnu.g='M';
11     printf("Before Function Call, Gender: %c\n",bishnu.g);
12     func(&bishnu.g);
13     printf("\nGender Modified! Printing from main():");
14     printf("\nGender=%c",bishnu.g);
15     getch();
16     return 0;
17 }
18
19 void func(char *gen){
20     *gen='F';
21     printf("After Function Call, Gender: %c\n", *gen);
22 }
```



```
1 #include<stdio.h>
2 #include<conio.h>
3 typedef struct{
4     char name[20];
5     int price, quantity;
6 }stores;
7
8 stores update(stores product, int p, int q){
9     product.price+=p;
10    product.quantity+=q;
11    return(product);
12 }
13 float mul(stores stock){
14     return(stock.price*stock.quantity);
15 }
16 int main( ){
17     float value; int p_inc,q_inc;
18     static stores item={"XYZ",25,12};
19     printf("Input increment values ...\n");
20     printf("Price increment and quantity increment: ");
21     scanf("%d%d",&p_inc,&q_inc);
22     item=update(item,p_inc,q_inc);
23     printf("\nUpdated values of item:\n");
24     printf("Name:      %s\n",item.name);
25     printf("Price:      %d\n",item.price);
26     printf("Quantity: %d\n",item.quantity);
27     value=mul(item);
28     printf("\nTotal Amount = %.2f\n",value);
29     getch(); return 0;
30 }
```

```
C:\Users\ErSKS\Google Drive ...
Input increment values ...
Price increment and quantity increment: 5 3

Updated values of item:
Name:      XYZ
Price:      30
Quantity: 15

Total Amount = 450.00
```

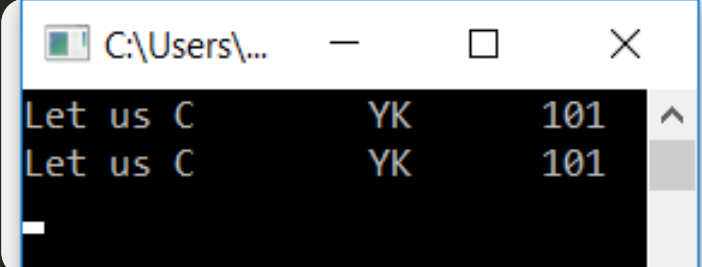
(rment)

# Structure & Pointer

C language allows declaring pointer to structure just like pointer to other ordinary variables. The declaration can be done in the following manner:

```
struct employee{  
    char *name;  
    char *roll_no;  
    int salary;  
}*emp1;
```

```
1  #include<stdio.h>  
2  #include<conio.h>  
3  int main(){  
4      struct book{  
5          char name[25];  
6          char author[25];  
7          int call_no;  
8      };  
9      struct book b1={"Let us C","YK",101};  
10     struct book *ptr;  
11     ptr=&b1;  
12     printf("%s\t%s\t%d\n",b1.name,b1.author,b1.call_no);  
13     printf("%s\t%s\t%d\n",ptr->name, ptr->author, ptr->call_no);  
14     getch();  
15     return 0;  
16 }
```



```
C:\Users\...  
Let us C      YK      101  
Let us C      YK      101
```

```

1  #include<stdio.h>
2  #include<conio.h>
3  int main(){
4      struct s_record{
5          char name[25], crn[12], grade;
6          float average;
7      }s[50], *ptr; /* ptr is a pointer of type structure s_record */
8
9      int i, n;
10     printf("Number of ss grades to be computed?");
11     scanf("%d", &n);
12     for(i=0; i<n; i++){
13         printf("\nStudent[%d]'s information:\n", i+1);
14         printf("Name: "); scanf("%s", s[i].name);
15         printf("CRN: "); scanf("%s", s[i].crn);
16         printf("Average Score: "); scanf("%f", &s[i].average);
17         printf("%s\t%s\t%.2f\n", s[i].name, s[i].crn, s[i].average);
18     }
19
20     ptr = s; /* pointer 'ptr' points to s[0] */
21     /* Assigning grades to 'n' ss*/
22     for(ptr = s; ptr<s+n; ptr++){
23         if(ptr->average<30.0){
24             ptr->grade = 'D';
25         }else if(ptr->average<50.0){
26             ptr->grade = 'C';
27         }else if(ptr->average<70.0){
28             ptr->grade = 'B';
29         }else{
30             ptr->grade = 'A';
31         }
32     }
33     /*Displaying Student Records*/
34     printf("\n%-20sREG_NUMBER \tAVERAGE\tGRADE\n", "NAME");
35     for(ptr=s; ptr<s+n; ptr++){
36         printf("%-20s%-12s\t", ptr->name, ptr->crn);
37         printf("%-7.2f\t%c\n", ptr->average, ptr->grade);
38     }
39     getch(); return 0;
40 }

```

```

C:\Users\ErSKS\Google Drive (c.khwopa@gm...
Number of students grades to be computed?3

Student[1]'s information:
Name: Niranjan
CRN: KCE074BCT025
Average Score: 88
Niranjan      KCE074BCT025    88.00

Student[2]'s information:
Name: Sushil
CRN: KCE074BCT048
Average Score: 86.92
Sushil KCE074BCT048    86.92

Student[3]'s information:
Name: Nabin
CRN: KCE074BCT023
Average Score: 80.62
Nabin KCE074BCT023    80.62

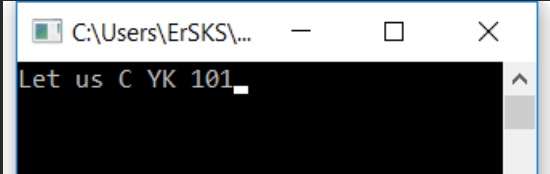
NAME                REG_NUMBER          AVERAGE GRADE
Niranjan            KCE074BCT025        88.00    A
Sushil              KCE074BCT048        86.92    A
Nabin               KCE074BCT023        80.62    A

```



# Passing Structure to Function using Reference

```
C8_18.C
1  #include<stdio.h>
2  #include<conio.h>
3  struct book{
4      char name[25];
5      char author[25];
6      int call_no;
7  };
8  void display(struct book *);
9
10 int main(){
11     struct book b1={"Let us C", "YK", 101};
12     display(&b1);
13     getch();
14     return 0;
15 }
16
17 void display(struct book *p){
18     printf("%s %s %d", p->name, p->author, p->call_no);
19 }
```



# Function, Structure & Pointer

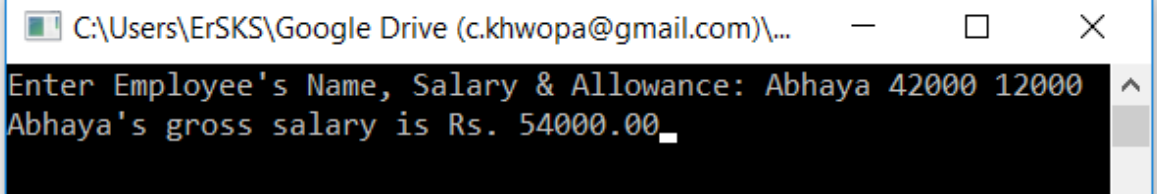
31

Er. Shiva K. Shrestha (HoD, Computer Department)  
10/3/2018

```
File Edit Search Run Compile Debug Project Options Window Help
C8_20_~1.C 1=[↑]
#include<stdio.h>
#include<conio.h>
void display(struct Nepal *b);
struct Nepal{
    char PM[30];
    char capitalCity[20];
    int states;
};
int main(){
    struct Nepal b={"KP Oli","Kathmandu",7};
    clrscr();
    display(&b);
    getch();
    return 0;
}
void display(struct Nepal *b){
    printf("\n%s %s %d",b->PM,b->capitalCity, b->states);
}
11:18
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
```

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
KP Oli Kathmandu 7
```

```
C8_19.c
1  #include<stdio.h>
2  #include<conio.h>
3  struct employee{
4      char name[20];
5      float salary, allowance;
6  }e;
7  /*ep is a pointer of type struct employee*/
8  float grossCalc(struct employee *ep){
9      /*adds up the employee's salary and allowance,
10     and returns the result as gross salary*/
11     return (ep->salary + ep->allowance);
12 }
13 int main(){
14     float gross;
15     printf("Enter Employee's Name, Salary & Allowance: ");
16     scanf("%s%f%f", e.name, &e.salary, &e.allowance);
17     fflush(stdin);
18     gross = grossCalc(&e);
19     printf("%s's gross salary is Rs. %.2f", e.name, gross);
20     getch(); return 0;
21 }
```



```
C:\Users\ErSKS\Google Drive (c.khwopa@gmail.com)\...
Enter Employee's Name, Salary & Allowance: Abhaya 42000 12000
Abhaya's gross salary is Rs. 54000.00
```

# Q/A?

33

## Thank You!

Er. Shiva K. Shrestha

[computer.khwopa@gmail.com](mailto:computer.khwopa@gmail.com)

Er. Shiva K. Shrestha (HoD, Computer Department)  
10/3/2018