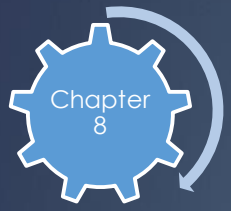


Structures

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:

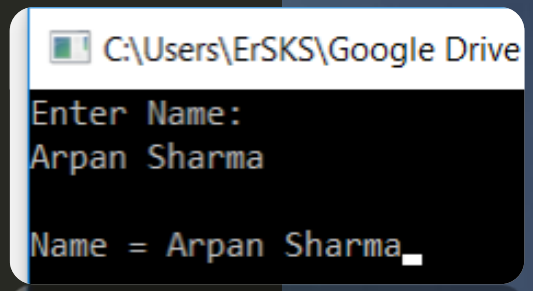
```
struct student{
        char name[50];
        int roll;
        float marks;
        char gender;
};
```

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

C8_1.C

```c
#include<stdio.h>
#include<conio.h>
struct account{
    int acc_no;
    char acc_type;
    char name[80];
    float balance;
};
struct account cust;
int main(){
    printf("Enter Name: \n");
    //scanf("%s", cust.name);
    gets(cust.name);
    printf("\nName = %s", cust.name);
    getch();
    return 0;
}
```



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

Task

- 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

Er. Shiva K. Shrestha (HoD, Computer Department)

10/3/2018

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

- structure_name.member_name

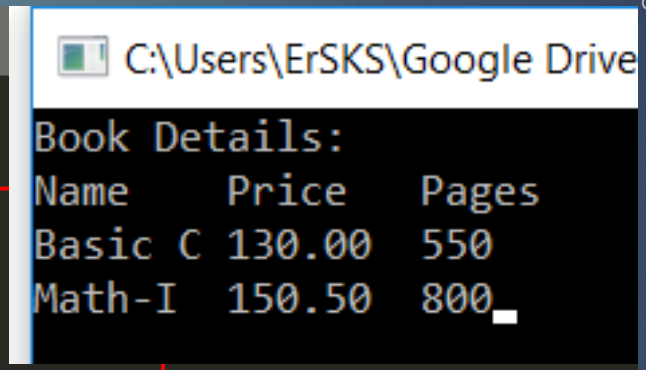
- printf("Book Name = %s", b1.name); // C Programming
- printf("Book Name = %f", b1.price); // 130.00
- printf("Book Name = %d", b1.pages); // 550
- printf("Book Name = %s", b2.name); // ?
- printf("Book Name = %f", b2.price); // ?
- printf("Book Name = %d", b2.pages); // ?

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.

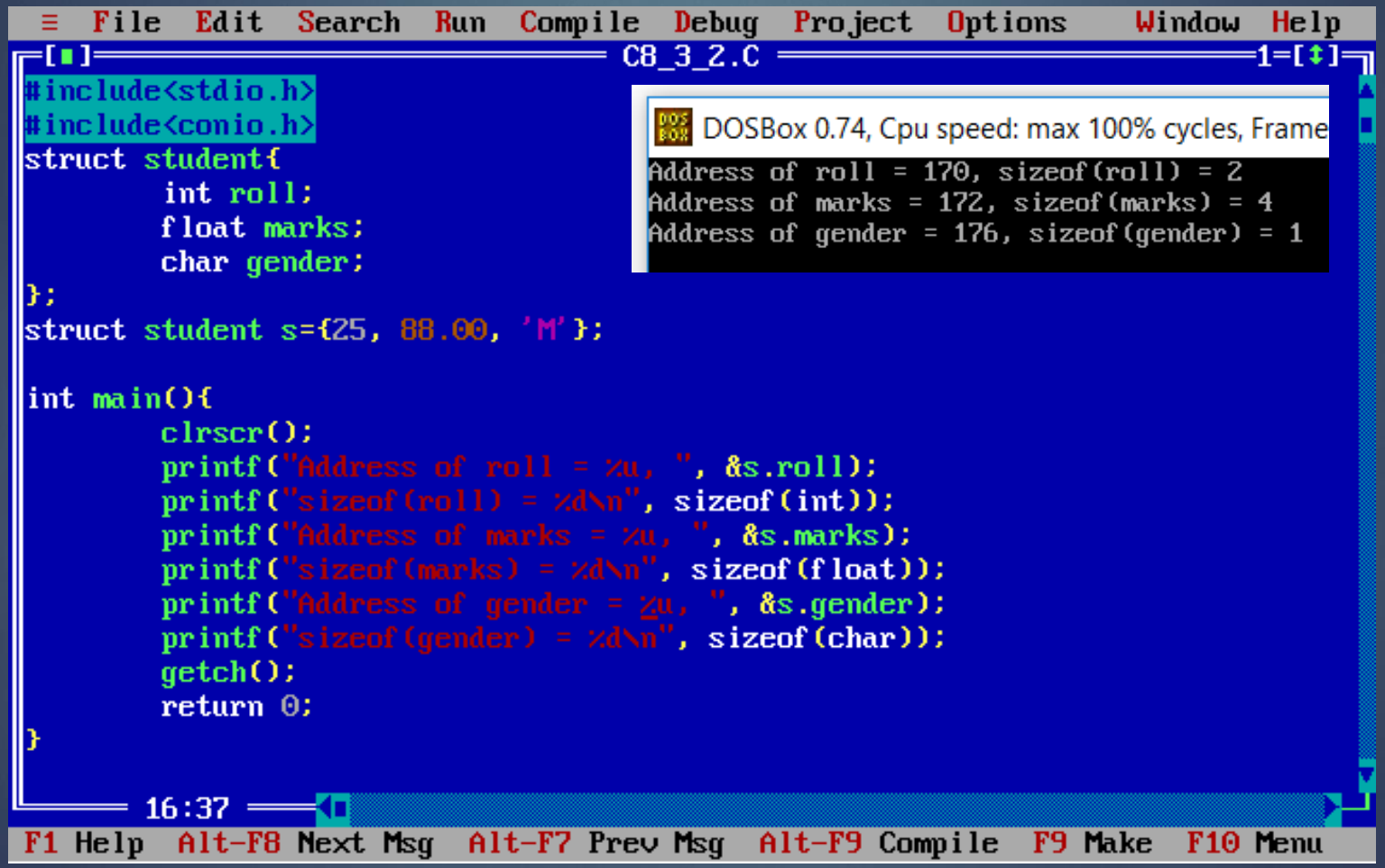
C8_3.C

```c
#include<stdio.h>
#include<conio.h>
struct book{
    char name[20];
    float price;
    int pages;
};
struct book b1={"Basic C",130.00,550};
struct book b2={"Math-I",150.50,800};

int main(){
    printf("Book Details:\nName\tPrice\tPages\n");
    printf("%s\t%.2f\t%d", b1.name, b1.price, b1.pages);
    printf("\n%s\t%.2f\t%d", b2.name, b2.price, b2.pages);
    getch();
    return 0;
}
```



How Structure Elements are Stored?

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



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

C8_4.C

```c
#include<stdio.h>
#include<conio.h>
struct book{
    char name[20];
    float price;
    int pages;
};
struct book b1,b2,b3;
int main(){
    printf("Enter Name, Price & No. of Pages of 3 Books:\n");
    scanf("%s%f%d",b1.name,&b1.price,&b1.pages);
    scanf("%s%f%d",b2.name,&b2.price,&b2.pages);
    scanf("%s%f%d",b3.name,&b3.price,&b3.pages);

    printf("\nThis is what you entered");
    printf("\n%s\t%.2f\t%d",b1.name,b1.price,b1.pages);
    printf("\n%s\t%.2f\t%d",b2.name,b2.price,b2.pages);
    printf("\n%s\t%.2f\t%d",b3.name,b3.price,b3.pages);
    getch();
    return 0;
}
```

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

- 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.

C8_5.C

```
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 }
```

```
C:\Users\ErSKS
10 20
```

Nested Structure

C8_6.C

```c
#include<stdio.h>
#include<conio.h>
struct employee{
    char name[25];
    long int emp_code;
    char post[20];
    float salary;
    struct emp_address{
        int no;
        char street[25];
        char area[20];
    }adr1;
}e1 = {"Er. Shiva K. Shrestha",74801,"HoD",65000.00,8,"Garud Kundal Road","Libali"};

int main(){
    printf("Employee Adress Area = %s", e1.adr1.area);
    getch();
    return 0;
}
```

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

```c
struct employee{
    char name[25];
    long int emp_code;
    char post[20];
    float salary;
} emp1, emp2;
```

```c
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

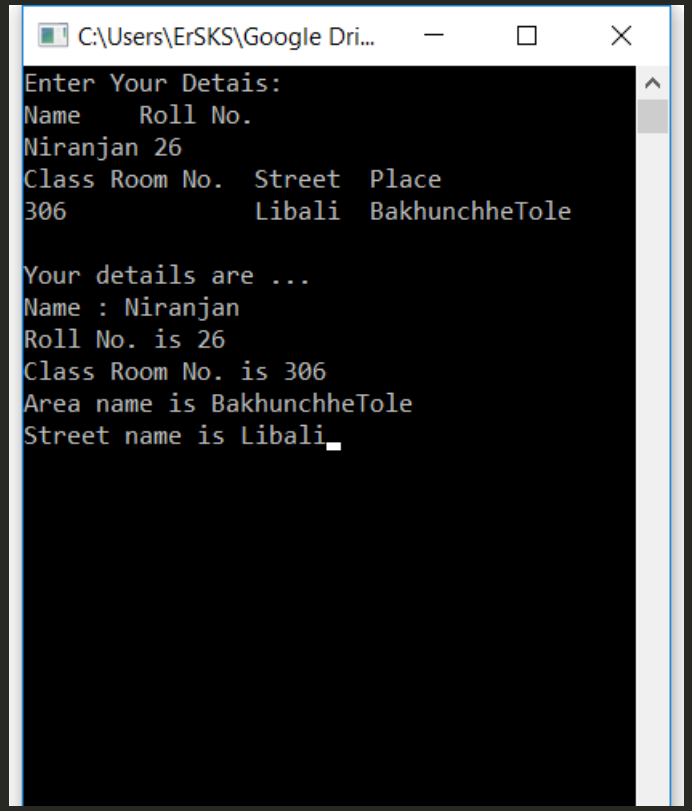
C8_8.C

```c
#include<stdio.h>
#include<conio.h>
struct address{
    char phone[15];
    char city[25];
    int citycode;
};
struct emp{
    char name[25];
    struct address a;
};
int main(){
    int i;
    struct emp e={"Jack","54515","KTM",101};
    printf("Name=%s\tPhone=%s",e.name,e.a.phone);
    printf("\nCity=%s\tCity Code=%d",e.a.city,e.a.citycode);
    getch();
    return 0;
}
```

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

C8_9.C

```c
#include <stdio.h>
#include<conio.h>
struct add{
    int door_no;
    char street[20];
    char place[30];
};
struct student{
    char name[30];
    int roll_no;
    struct add address;
};
int main(){
    struct student std;
    printf("Enter Your Detais:\n");
    printf("Name\tRoll No.\n");
    scanf("%s%d",std.name,&std.roll_no);
    printf("Class Room No.\tStreet\tPlace\n");
    scanf("%d%s%s",&std.address.door_no,std.address.street,std.address.place);
    printf("\nYour details are ...\n");
    printf("Name : %s",std.name);
    printf("\nRoll No. is %d",std.roll_no);
    printf("\nClass Room No. is %d",std.address.door_no);
    printf("\nArea name is %s",std.address.place);
    printf("\nStreet name is %s",std.address.street);
    getch();
    return 0;
}
```



Array of Structures

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

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

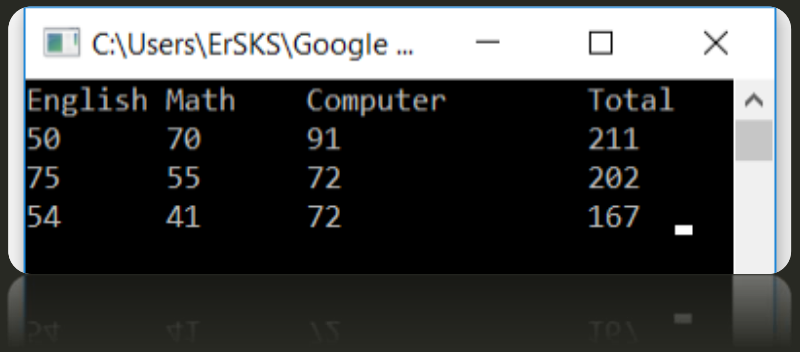
To print the price of structure 3, write

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

Er. Shiva K.
10/3/2018

C8_10.C

```
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 }
```



Task

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

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

C8_11.C

```c
#include<stdio.h>
#include<conio.h>
struct book{
  char name[20];
  int price;
  int pages;
}b[5];

int main(){
  int i;
  for(i=0;i<5;i++){
    printf("%d. Enter Name, Price, and Pages: ",(i+1));
    scanf("%s%d%d", &b[i].name, &b[i].price, &b[i].pages);
  }
  printf("\n%-15s\tPrice\tPages","Book Name");
  for(i=0;i<5;i++){
    printf("\n%-15s\t%d\t%d", b[i].name, b[i].price, b[i].pages);
  }
  getch();
  return 0;
}
```

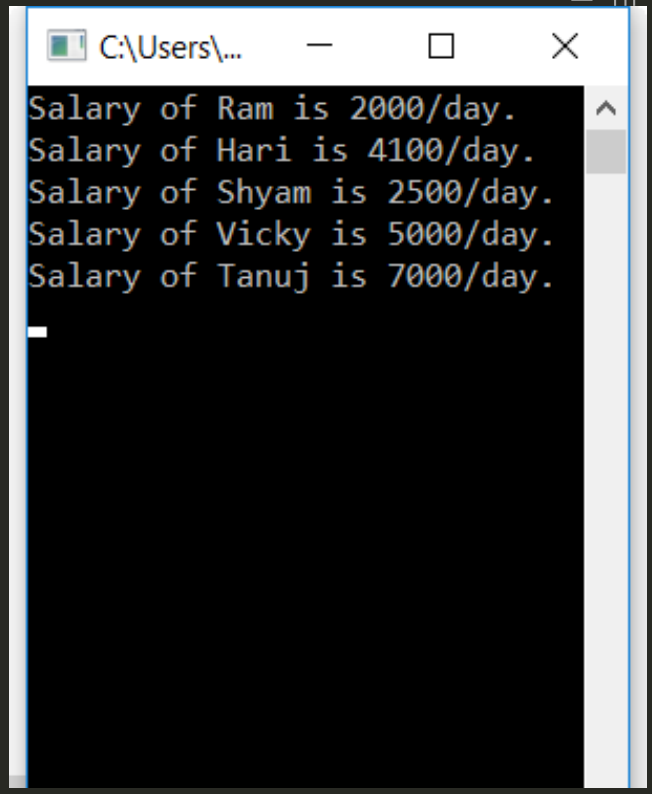
C:\Users\ErSKS\Google Drive (c.khwopa@gmail.com)\C_Codes\Ch8\C8_11.exe

```
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

C8_12.C

```c
#include<stdio.h>
#include<conio.h>
struct employee{
  char name[20];
  char code[5];
  int salary;
} emp [5] = {
  {"Ram","E01", 2000},
  {"Hari","E02", 4100},
  {"Shyam","E03", 2500},
  {"Vicky","E04", 5000},
  {"Tanuj","E05", 7000},
};

int main(){
  int i;
  for(i=0;i<5;i++){
    printf("Salary of %s is %d/day.\n",emp[i].name,emp[i].salary);
  }
  getch();
  return 0;
}
```



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*.

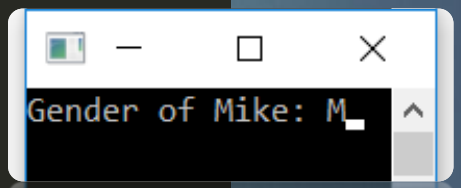
Function & Structure

- 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

C8_13.C

```c
#include<stdio.h>
#include<conio.h>
struct{
    char g;
    int y;
    float z;
    char s[10];
}mike;
void func(char);

int main(){
  mike.g='M';
  func(mike.g);
  getch();
  return 0;
}

void func(char a){
  printf("Gender of Mike: %c", a);
}
```

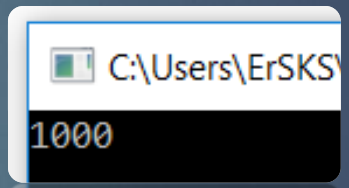


Er. Shiva K. Shrestha (HoD, Computer Department)

Passing an entire structure to a function

C8_14.C

```c
#include<stdio.h>
#include<conio.h>
struct test{
    int a, b;
    char ch;
};
void function(struct test);

int main(){
    struct test arg;
    arg.a=1000;
    function(arg);
    getch();
    return 0;
}

void function(struct test param) {
    printf("%d",param.a);
}
```



Er. Shiva K. Shrestha (HoD, Computer Department)

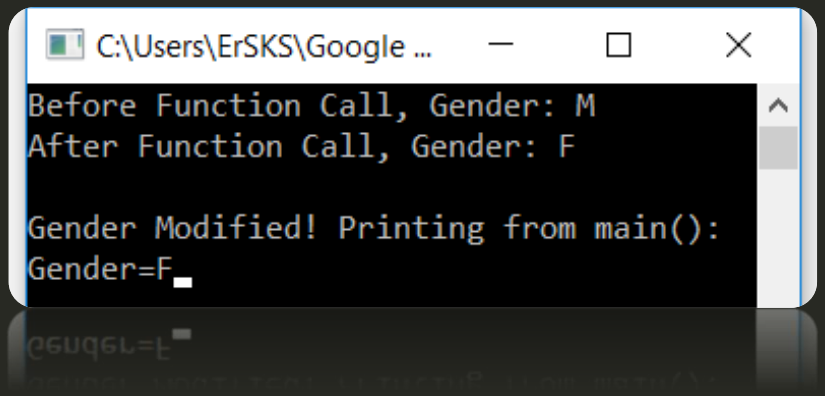
Task

- 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

C8_15.C

```c
#include<stdio.h>
#include<conio.h>
struct{
    char g;
    int y;
}bishnu;
void func(char *gen);

int main(){
    bishnu.g='M';
    printf("Before Function Call, Gender: %c\n",bishnu.g);
    func(&bishnu.g);
    printf("\nGender Modified! Printing from main():");
    printf("\nGender=%c",bishnu.g);
    getch();
    return 0;
}

void func(char *gen){
    *gen='F';
    printf("After Function Call, Gender: %c\n", *gen);
}
```



Er. Shiva K. Shrestha (HoD, Computer Department)

C8_15_2.C

```c
#include<stdio.h>
#include<conio.h>
typedef struct{
    char name[20];
    int price, quantity;
}stores;

stores update(stores product, int p, int q){
    product.price+=p;
    product.quantity+=q;
    return(product);
}
float mul(stores stock){
    return(stock.price*stock.quantity);
}
int main( ){
    float value; int p_inc,q_inc;
    static stores item={"XYZ",25,12};
    printf("Input increment values ...\n");
    printf("Price increment and quantity increment: ");
    scanf("%d%d",&p_inc,&q_inc);
    item=update(item,p_inc,q_inc);
    printf("\nUpdated values of item:\n");
    printf("Name:     %s\n",item.name);
    printf("Price:    %d\n",item.price);
    printf("Quantity: %d\n",item.quantity);
    value=mul(item);
    printf("\nTotal Amount = %.2f\n",value);
    getch(); return 0;
}
```

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
```

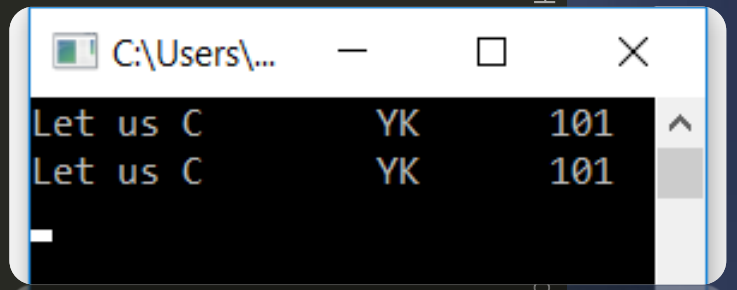
Structure & Pointer

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;
```

C8_16.C

```
#include<stdio.h>
#include<conio.h>
int main(){
    struct book{
        char name[25];
        char author[25];
        int call_no;
    };
    struct book b1={"Let us C","YK",101};
    struct book *ptr;
    ptr=&b1;
    printf("%s\t%s\t%d\n",b1.name,b1.author,b1.call_no);
    printf("%s\t%s\t%d\n",ptr->name, ptr->author, ptr->call_no);
    getch();
    return 0;
}
```



C8_17.C

```c
#include<stdio.h>
#include<conio.h>
int main(){
  struct s_record{
     char name[25], crn[12], grade;
     float average;
  }s[50], *ptr; /* ptr is a pointer of type structure s_record */

  int i, n;
  printf("Number of ss grades to be computed?");
  scanf("%d", &n);
  for(i=0; i<n; i++){
   printf("\nStudent[%d]'s information:\n", i+1);
   printf("Name: "); scanf("%s", s[i].name);
   printf("CRN: "); scanf("%s", s[i].crn);
   printf("Average Score: "); scanf("%f", &s[i].average);
   printf("%s\t%s%8.2f\n", s[i].name, s[i].crn, s[i].average);
  }

  ptr = s; /* pointer 'ptr' points to s[0] */
  /* Assigning grades to 'n' ss*/
  for(ptr = s; ptr<s+n; ptr++){
    if(ptr->average<30.0){
     ptr->grade = 'D';
   }else if(ptr->average<50.0){
     ptr->grade = 'C';
   }else if(ptr->average<70.0){
     ptr->grade = 'B';
   }else{
     ptr->grade = 'A';
   }
  }
  /*Displaying Student Records*/
  printf("\n%-20sREG_NUMBER  \tAVERAGE\tGRADE\n","NAME");
  for(ptr=s;ptr<s+n;ptr++){
    printf("%-20s%-12s\t", ptr->name, ptr->crn);
    printf("%-7.2f\t%c\n", ptr->average, ptr->grade);
  }
  getch(); return 0;
}
```

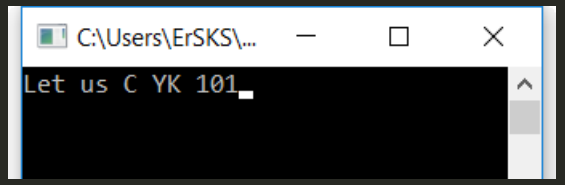
C:\Users\ErSKS\Google Drive (c.khwopa@gm...

```
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

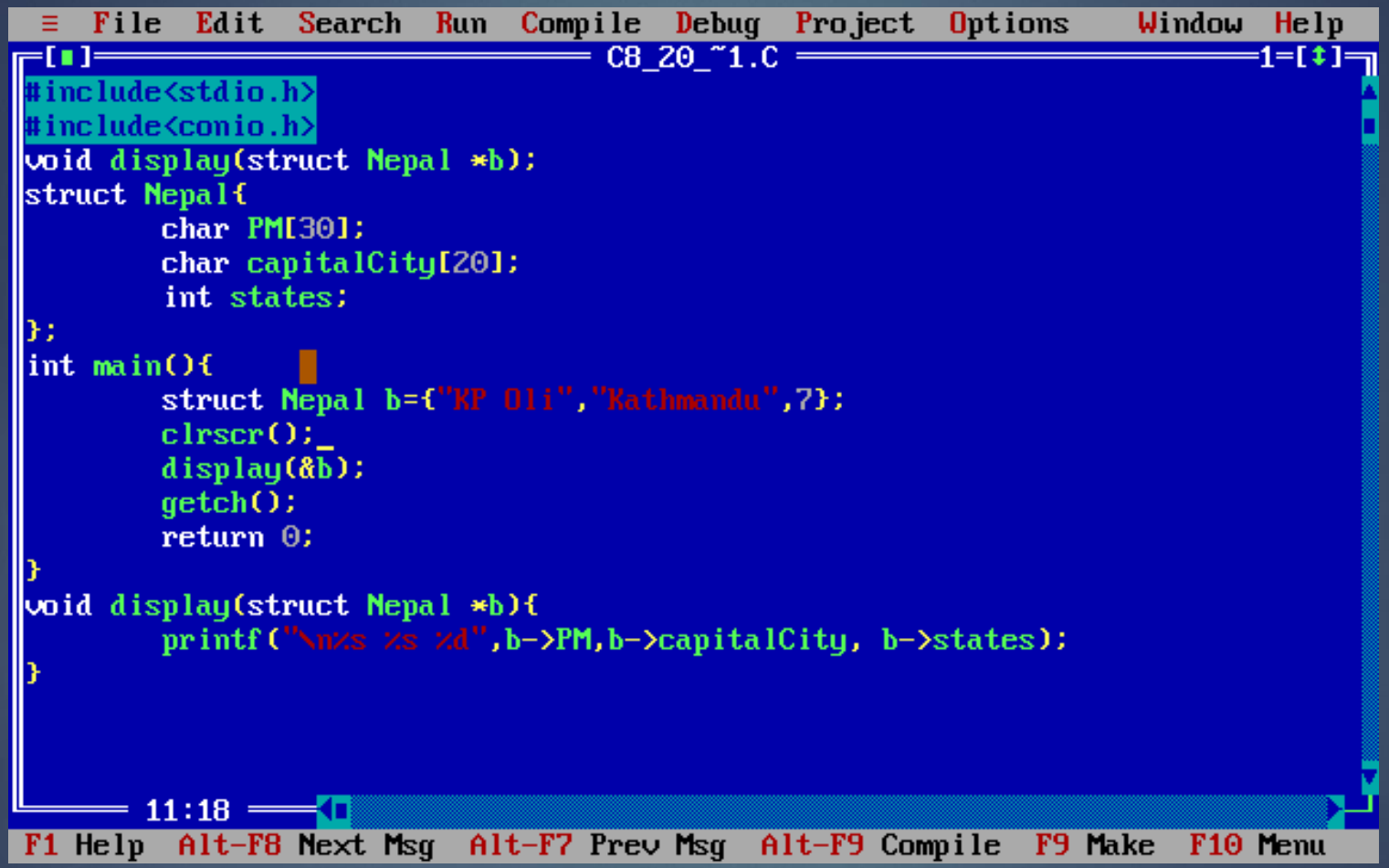
C8_18.C

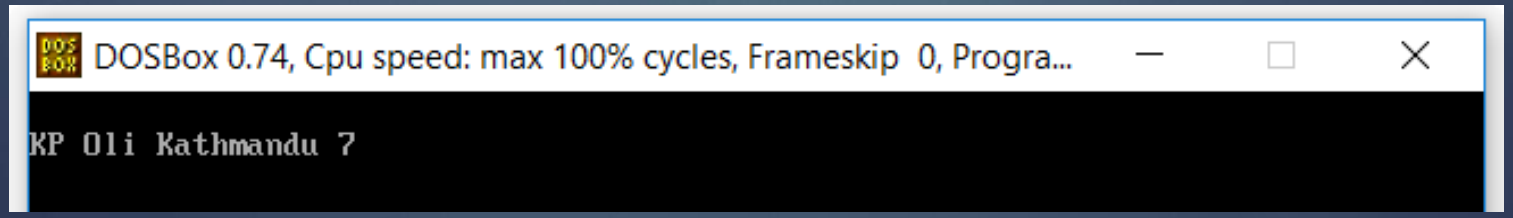
```c
#include<stdio.h>
#include<conio.h>
struct book{
    char name[25];
    char author[25];
    int call_no;
};
void display(struct book *);

int main(){
    struct book b1={"Let us C", "YK", 101};
    display(&b1);
    getch();
    return 0;
}

void display(struct book *p){
    printf("%s %s %d", p->name, p->author, p->call_no);
}
```



Er. Shiva K. Shrestha (HoD, Computer Department)

Function, Structure & Pointer





C8_19.c

```c
#include<stdio.h>
#include<conio.h>
struct employee{
    char name[20];
    float salary, allowance;
}e;
/*ep is a pointer of type struct employee*/
float grossCalc(struct employee *ep){
    /*adds up the employee's salary and allowance,
    and returns the result as gross salary*/
    return (ep->salary + ep->allowance);
}
int main(){
    float gross;
    printf("Enter Employee's Name, Salary & Allowance: ");
    scanf("%s%f%f", e.name, &e.salary, &e.allowance);
    fflush(stdin);
    gross = grossCalc(&e);
    printf("%s's gross salary is Rs. %.2f", e.name, gross);
    getch(); return 0;
}
```

```
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
```

Er. Shiva K. Shrestha (HoD, Computer Department)

Q/A?

Thank You!

Er. Shiva K. Shrestha
computer.khwopa@gmail.com