

```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodeName()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '      %s [label="%s' % (nodename, label),
    if isinstance(ast[1], str):
        print '      %s' % ast[1]
    else:
        print '['
    else:
        print '];'
    children = []
    for n, childrenumerate(ast[1:]):
        children.append(dotwrite(child))
    print ', ' % nodename
    for in :namechildren
        print '%s' % name,
```

CSE131: Computer Programming

Chapter (6)



- * structure definitions do not reserve any space in memory

Syntax error
=
نقصان الـ semi بناء
تعريف الـ struct

طريقة
جودة ان
طريقة جودة
struct /

∴ 2nd way -

↓ in this way struct name can be omitted

Accessing members of the struct

member vars pointer

• (->)

```

struct student
{
    int id;
    char name[50];
    float gpa;
};

void main()
{
    struct student stu1;
    stu1.id = 19000000;
    stu1.gpa = 4.00;
    strcpy(stu1.name, "Juba");
}

```

```

2 void main()
{
    struct student stu1 = {1980000, "Juba", 4.00};
}

```

OR

```

struct date dateofBirth;
dateofBirth = (struct date){.month=11,.day=11,.year=2000};

```

```
printf("ID:%d \n", stu1.id);
printf("Name:%s \n", stu1.name);
printf("GPA:%f \n", stu1.GPA);
```

```
ID:1900000
Name:Juba
GPA:4.000000
```

in this way struct name can be omitted

```
typedef struct student
{
    int id;
    char name[50];
    float GPA;
} Student;

// تعریف
// ساختار
// Capital
```

```
void main()
{
    student stu1;
    stu1.id = 19000000;
    stu1.GPA = 4.00;
    strcpy(stu1.name, "Juba");

    printf("ID:\t%d\n", stu1.id);
    printf("Name:\t%s\n", stu1.name);
    printf("GPA:\t%f\n", stu1.GPA);
}
```

⇒

ID:	19000000
Name:	Juba
GPA:	4.000000

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

struct student
{
    int id;
    char name[50];
    float GPA;
    struct date dateofbirth;
};
```

```
void main()
{
    struct student stu1 = { 1908080, "Juba", 4, {11,11,2008} };

    printf("ID: %d\n", stu1.id);
    printf("Name: %s\n", stu1.name);
    printf("GPA: %f\n", stu1.GPA);
    printf("date of birth: %d/%d/%d", stu1.dateofbirth.day, stu1.dateofbirth.month, stu1.dateofbirth.year);
}
```

Stru عبارة عن
معلومات لشخص
الذي ليس له اسم
شخصي

```
#define MAX 3
struct person
{
    char name[5];
    int age;
};
```

```
void main()
{
    int i;
    struct person info[MAX];
    for(i=0; i<MAX; i++)
    {
        printf("please enter name(%d):", i+1);
        scanf("%s", info[i].name);
        printf("please enter age(%d):", i+1);
        scanf("%d", &info[i].age);
    }
}
```

→ Passing structs to Functions ⇒

```
#include <stdio.h>
struct student
{
    char name[50];
    int age;
};

void display(struct student s);

① struct نام دار
function
```

```
int main()
{
    struct student s1;
    printf("Enter name: ");
    scanf("%s", s1.name);

    printf("Enter age: ");
    scanf("%d", &s1.age);

    display(s1);
    return 0;
}
```

```
void display(struct student s)
{
    printf("\nDisplaying information\n");
    printf("Name: %s", s.name);
    printf("Age: %d", s.age);
}
```

Function ال بتقبل بيانات المسترمت
ال متخزنه في ال struct

⑤. بطلان من ار user
ی بخل اسم و سن و ی حفظ
فی struct اسم SI
ثم استدی ار function

Return Structs From a Functions \Rightarrow

```
#include <stdio.h>
struct student
{
    char name[50];
    int age;
};
struct student getInformation();
```

struct → function
function → struct student

```
int main()
{
    struct
    s = get
    printf
    printf
    printf
    return
}
```

عرفنا Variable اسمه `struct student`
بياضه البيانات من الـ `function`

```
void display(struct student s)
{
    printf("\nDisplaying information\n");
    printf("Name: %s", s.name);
    printf("\nAge: %d", s.age);
}
```

بجملہ list من Variables
تصنيف category
وارثہ کے

كل واحد يحدد قيمة بالترتيب

1 2 3

enum week{Mon, Tue, Wed};
enum day;

OR

enum week{Mon, Tue, Wed};
enum week day;

- جملة اسمها week
- بديين حرفت ال variables
- كل واحد يومه day

* enum week{Mon=2, Tue=3, Wed=7};
 ← لوانيز قيمه معينه اكل واحد
 بعد لها أنا

* **Unions** : it's a user define data type, as all members share the same memory location

Structure	Unions
-----------	--------

* defining Union \Rightarrow

```
union Data {
    int i;
    float f;
    char str[20];
} data;
```

} Union members

- * accessing members of Union is same as the struct

③ الـ function بتقبل بيانات المستخدم
التي متخزنة في الـ struct

```

struct student getInformation()
{
    struct student s1;

    printf("Enter name: ");
    scanf("%s", &s1.name);

    printf("Enter age: ");
    scanf("%d", &s1.age);

    return s1;
}

```

function بيانات من
المستخدم و return
البيانات دي