

Fundamentals of Structured Programming

Lecture 4 Structures

Course Coordinator: Prof. Zaki Taha Fayed

Presented by: Dr. Sally Saad

SallySaad@gmail.com

DropBox folder link

<https://www.dropbox.com/sh/85vnrgkfqgrzhwn/AABdwKLJZqZs26a7u-y0AFwia?dl=0>

Credits to Dr. Salma Hamdy for Content Preparation

Quote of the Day!

If you are not willing to learn,
No one can help you!
If you are determined to learn,
No one can stop you!

Remarks on Arrays

1. Assigning a 2D array to another .

(Do I always nested loop for 2D Array?)

2. 2D can be viewed as an Array of Array.

3. Static Array Vs. Dynamic Array

(vstudio vs codeblocks)

Class Accumulative Project: **Employees Salary for Companies**



Class Accumulative Project:

Employees Salary for Companies

- Refer to Bonus Exercise in Dropbox folder under folder Lecture 3.
- Updated code with 2DArrays (Task 3) → (*difficulties? Disadvantage?*)
- Task 4 : at the end of the slides 😊

Structures

Contents

1. Structures

- Why?
- Definition
- Declaration
- Initialization
- Elements (fields/member variables) Referencing

2. Arrays of Structures

3. Examples

4. Bonus Task 4

What is a Structure? Why?

Structure: collection of data of *different types*.

- Called an “aggregate” data type.
- Enables you to manage several variables under one name.
- Adds organization and structure to your program.
- Examples

Information about a student

```
graph LR; A[Information about a student] --> B[Multidimensional arrays]; A --> C[Structure];
```

Multidimensional arrays

Structure

What is a Structure? Why? – (cont.)

Information about a student

Multidimensional arrays

```
int ID[4];
```

St1	2008170001
St2	2008170002
St3	2008170003
St4	2008170004

```
int grades[4][5];
```

	Subj1	Subj2	Subj3	Subj4	Subj5
St1	59	89	90	100	50
St2	58	56	89	43	89
St3	27	58	92	46	89
St4	76	30	25	90	94

```
char names[4][50];
```

St1	S	A	L	M	A		H	A	...
St2	S	A	R	A		A	H	M	...
St3	A	H	M	E	D		A	L	Y
St4	W	A	L	A	A	

What is a Structure? Why? – (cont.)

Information about a student

Structure

(WHAT TYPE) Stud;

Stu1

ID	2008170001									
Name	S	A	L	M	A		H	A	M	...
Grades	59	89	90	100	50					

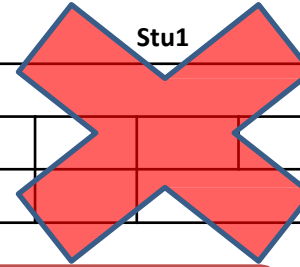
Stu2

ID	2008170002									
Name	S	A	R	A		A	H	M	E	D
Grades	58	56	89	43	89					

Structure Definition

```
1 // This program demos the use of structures
2 #include <iostream>
3 using namespace std;
4
5 struct Stud
6 {
7     int ID;
8     char name[50];
9     int grades[5];
10 }; // end of Stud
11
12
13 int main()
14 {
15     // variables
16
17
18     // processing
19
20     // output
21
22     return 0;
23 } // end main
```

ID										
Name										
Grades										



Name of new struct "type"

Name and types of members

- Does not allocate memory. It's just an indication of what our structure will look like.

Structure Variable Declaration (1)

```
1 // This program demos the use of structures
2 #include <iostream>
3 using namespace std;
4
5 struct Stud
6 {
7     int ID;
8     char name[50];
9     int grades[5];
10
11 }; // end of Stud
12
13 int main()
14 {
15     // variables
16     Stud stu1;
17
18     // processing
19
20     // output
21
22     return 0;
23 } // end main
```

Structure
variable

Stu1

Member
variables

ID
Name
Grades

ID									
Name									
Grades									

- Just like declaring simple variables: `type name;`
- Allocates enough memory for all members.
- Structure variable contains member variables for the defined “parts”.

Structure Variable Declaration (2)

```
1 // This program demos the use of structures
2 #include <iostream>
3 using namespace std;
4
5 struct Stud
6 {
7     int ID;
8     char name[50];
9     int grades[5];
10
11 } stu1, stu2; // end of Stud
12
13 int main()
14 {
15     // variables
16     Stud stu3, stu4;
17
18     // processing
19
20     // output
21
22     return 0;
23 } // end main
```

ID	0		
Name			
Grades	0	0	0

ID	0		
Name			
Grades	0	0	0

ID			
Name			
Grades			

ID			
Name			
Grades			

Stu1									
ID	0								
Name									
Grades	0	0	0	0	0				

Stu2									
ID	0								
Name									
Grades	0	0	0	0	0				

Stu3									
ID									
Name									
Grades									

Stu4									
ID									
Name									
Grades									

Structure Variable Initialization

```
1 // This program demos the use of structures
2 #include <iostream>
3 using namespace std;
4
5 struct Stud
6 {
7     int ID;
8     char name[50];
9     int grades[5];
10
11 } stu1, stu2; // end of Stud
12
13 int main()
14 {
15     // variables
16     Stud stu3 = {2008170003, "SALMA HAMDY", {59, 89, 90, 100, 50}},
17     stu4 = {2008170004};
18
19     // processing
20
21     // output
22
23     return 0;
24 } // end main
```

Stu1										
ID	0									
Name										
Grades	0	0	0	0	0					

Stu2										
ID	0									
Name										
Grades	0	0	0	0	0					

Auto initialized to zeros

Stu3										
ID	2008170003									
Name	S	A	L	M	A		H	A	M	...
Grades	59	89	90	100	50					

Stu4										
ID	2008170004									
Name										
Grades	0	0	0	0	0					

Structure Variable Members

```
5 struct Stud
6 {
7     int ID;
8     char name[50];
9     int grades[5];
10
11 } stu1, stu2; // end of Stud
12
13 int main()
14 {
15     // variables
16     Stud stu3 = {2008170003, "SALMA HAMDY", {59, 89, 90, 100, 50}},
17     stu4 = {2008170004};
18
19     // processing
20
21     // output
22     cout<<stu3;
23
24     return 0;
25 } // end main
```

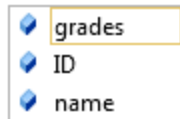
Stu3										
ID	2008170003									
Name	S	A	L	M	A		H	A	M	...
Grades	59	89	90	100	50					

Structure Variable Members

```
5 struct Stud
6 {
7     int ID;
8     char name[50];
9     int grades[5];
10
11 } stu1, stu2; // end of Stud
12
13 int main()
14 {
15     // variables
16     Stud stu3 = {2008170003, "SALMA HAMDY", {59, 89, 90, 100, 50}},
17     stu4 = {2008170004};
18
19     // processing
20
21     // output
22     cout<<stu1.
23
24     return 0;
25 } // end main
```

Stu3									
ID	2008170003								
Name	S	A	L	M	A		H	A	M
Grades	59	89	90	100	50				

- Your IDE will list all the member variables of this structure type.



Structure Variable Members

```
5 struct Stud
6 {
7     int ID;
8     char name[50];
9     int grades[5];
10
11 } stu1, stu2; // end of Stud
12
13 int main()
14 {
15     // variables
16     Stud stu3 = {2008170003, "SALMA HAMDY", {59, 89, 90, 100, 50}},
17     stu4 = {2008170004};
18
19     // processing
20
21     // output
22     cout<<stu3.ID<<endl;
23
24     return 0;
25 } // end main
```

Stu3										
ID	2008170003									
Name	S	A	L	M	A		H	A	M	...
Grades	59	89	90	100	50					

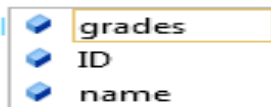
Structure variable name
Dot operator
Member variable name

Structure Variable Members

```
5 struct Stud
6 {
7     int ID;
8     char name[50];
9     int grades[5];
10
11 } stu1, stu2; // end of Stud
```

```
13 int main()
14 {
15     // variables
16     Stud stu3 = {2008170003, "SALMA HAMDY", {59, 89, 90, 100, 50}},
17     stu4 = {2008170004};
```

```
19 Stud.
```



```
23 cout<<Stud.ID;
```

```
25
```

Stu3									
ID	2008170003								
Name	S	A	L	M	A		H	A	M
Grades	59	89	90	100	50				

NOT AN ACTUAL PLACE
IN MEMORY

int Stud::ID

Error: a nonstatic member reference must be relative to a specific object

Structure Variable Members

```
9      int grades[5];
10
11 } stu1, stu2; // end of Stud
12
13 int main()
14 {
15     // variables
16     Stud stu3 = {2008170003, "SALMA HAMDY", {59, 89, 90, 100, 50}},
17     stu4 = {2008170004};
18
19     // processing
20
21     // output
22     cout<<stu3.ID<<endl;
23     cout<<stu3.name<<endl;
24     cout<<stu3.grades<<endl;
25
26
27     cout<<endl;
28
29     return 0;
30 } // end main
```

Stu3										
ID	2008170003									
Name	S	A	L	M	A		H	A	M	...
Grades	59	89	90	100	50					

Structure Variable Members

```
9      int grades[5];
10
11 } stu1, stu2; // end of Stud
12
13 int main()
14 {
15     // variables
16     Stud stu3 = {2008170003, "SALMA HAMDY", {59, 89, 90, 100, 50}},
17     stu4 = {2008170004};
18
19     // processing
20
21     // output
22     cout<<stu3.ID<<endl;
23     cout<<stu3.name<<endl;
24     cout<<stu3.grades<<endl;
25     for(int i=0; i<5; i++)
26         cout<<stu2.grades[i]<<"\t";
27     cout<<endl;
28
29     return 0;
30 } // end main
```

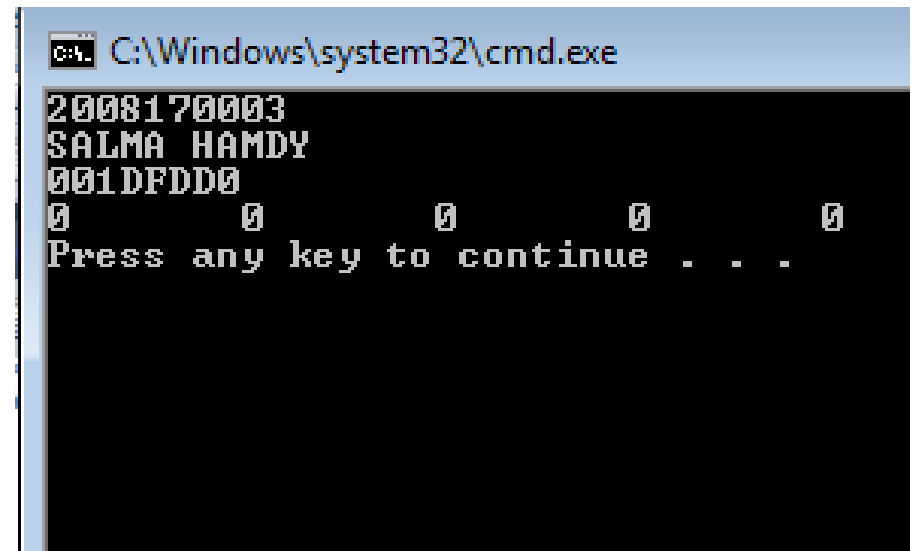
Stu3									
ID	2008170003								
Name	S	A	L	M	A		H	A	M
Grades	59	89	90	100	50				

What's the difference between line 24 and lines 25-26 ?

Structure Variable Members

```
9      int grades[5];
10
11 } stu1, stu2; // end of Stud
12
13 int main()
14 {
15     // variables
16     Stud stu3 = {2008170003, "SALMA HAMDY", {59, 89, 90, 100, 50}},
17     stu4 = {2008170004};
18
19     // processing
20
21     // output
22     cout<<stu3.ID<<endl;
23     cout<<stu3.name<<endl;
24     cout<<stu3.grades<<endl;
25     for(int i=0; i<5; i++)
26         cout<<stu2.grades[i]<<"\t";
27     cout<<endl;
28
29     return 0;
30 } // end main
```

Stu3									
ID	2008170003								
Name	S	A	L	M	A		H	A	M
Grades	59	89	90	100	50				



```
C:\Windows\system32\cmd.exe
2008170003
SALMA HAMDY
001DFDD0
0      0      0      0      0
Press any key to continue . . .
```

More than one struct Type

```
1  |
2  |
3  |
4  |
5  | struct Stud
6  | {
7  |     int ID;
8  |     char name[50];
9  |     int grades[5];
10 |
11 | } stu1, stu2; // end of Stud
12 |
13 | struct Teacher
14 | {
15 |     int ID;
16 |     char name[50];
17 |     double salary;
18 | } t1;
19 |
20 | int main()
21 | {
22 |     // variables
23 |     Stud stu3 = {2008170003, "SALMA HAMDY", {59, 89, 90, 100, 50}};
24 |     Teacher t2 = {0};
25 |
26 |     // processing
27 |     cin>>t2.ID; cin>>t2.name;
28 |     cin>>stu2.ID;
29 |
30 |     // output
31 |     cout<<stu3.ID<<endl;
```

Assignment of Structure Variables

```
1  |
2  |
3  |
4  |
5  | struct Stud
6  | {
7  |     int ID;
8  |     char name[50];
9  |     int grades[5];
10 |
11 | } stu1, stu2; // end of Stud
12 |
13 | struct Teacher
14 | {
15 |     int ID;
16 |     char name[50];
17 |     double salary;
18 | } t1;
19 |
20 | int main()
21 | {
22 |     // variables
23 |     Stud stu3 = {2008170003, "SALMA HAMDY", {59, 89, 90, 100, 50}};
24 |     Teacher t2 = {0};
25 |
26 |     // processing_
27 |     stu2 = stu3;
28 |     t1    = stu2;
29 |
30 |     // output
31 |     cout<<stu3.ID<<endl;
```

Of course for
variable of the same
type only. Even if
they had the same
member variables!

Arrays of Structures

Example 1

- Write a program to accept the grades of three students in five subjects, and displays the average score of each student along with a pass/fail status.

stud[0]	id[6]			...
	name[50]			...
	gender			
	age			

stud[1]	id[6]			...
	name[50]			...
	gender			
	age			

stud[2]	id[6]			...
	name[50]			...
	gender			
	age			

...

	...			
	...			

...

A close-up photograph of a white computer keyboard. The central focus is a single, large white key with rounded corners. On this key, there is a dark blue icon of a coffee cup with three wavy lines above it representing steam. Below the icon, the word "Break" is printed in a dark blue, serif typeface. The key is set against the background of other white keys, including one with a double quote symbol to the left and one with a dash/slash symbol to the right. The lighting is soft, creating subtle shadows and highlights on the keys' surfaces.

Break

Arrays of Structures

Example 2 ****Class Bonus****

- Update the design of Example 1 to describe Year 1 class having 1 teacher and 3 students.
- Read in five subjects, and display the average score of each student along with a pass/fail status.

- Now can you think of a description for our poor Employee, in the bonus Package?



Class Accumulative Project: **Employees Salary for Companies**



Class Accumulative Project: Employees Salary for Companies

Tasks 1, 2, 3 (DONE😊)

TASK 4 (NEW* BONUS):



- **Re-design** your program to create your own data type representing necessary information of employees (**using 2 struct: Emp, Company**).
- **Re-implement Task 3** with your new created struct by declaring an Array of 10 employees- make the suitable changes.
- Submit your code as text in this form, **due Date Monday 5/3/2018 at 11:59 pm (Extended)**

<https://goo.gl/forms/Kq6g1sYp0FZbwTz93>



Roll of Honour

TASK 2

- Best Scorers😊
- General (G1):

1. Andrew Awny
2. Abanoub Lotfy
3. Habiba Khaled

GOOD JOB!



Roll of Honour

TASK 2

- Best Scorers😊
- Software Engineering Department :

1. Hanin Tamer
2. Ahmed Safwat
3. Mohamed Amr



Roll of Honour

TASK 2

- Best Scorers 😊
- Bio-Informatics Department:

1. A'laa ElSayed Elfayoumi

2. Aya Hassan

3. Reem Osama



Roll of Honour

TASK 2

- Best Scorers 😊
- General (G2):

1. Monica Adel

2. Mohamed Adel Abo Elfetouh

3. Youssef halim Youssef

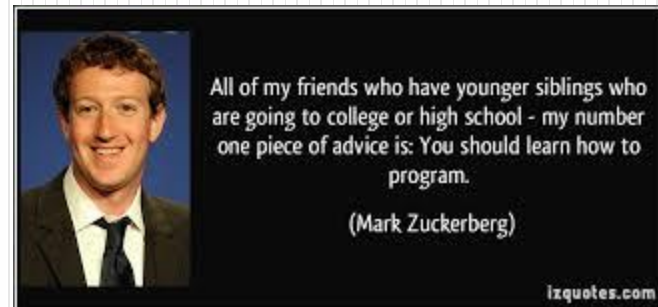
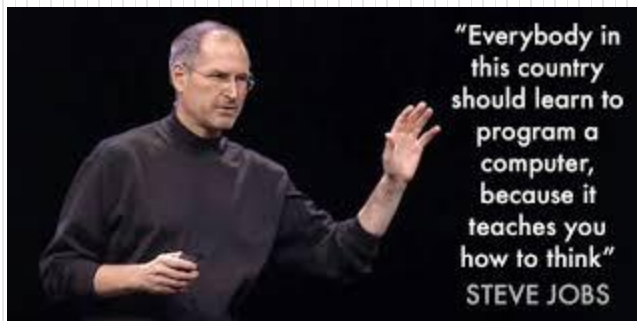
GOOD JOB!



Wanted List !

- The following students are kindly asked to pass by my office after the lecture:
- Mahmoud Sayed Afifi
- Moustafa Ismail Bakr
- Amr Khaled Hosny
- Amr Sayed Ahmed

Structured Programming Projects



**WORK
HARD
DREAM
BIG.**

Projects Regulations

1. Teams:

1) **General Group** : 4-6 members/team (a mix of G1 and G2 is allowed)

Credit Hours Program Group: 3-4 members/team (allowed to be of the same department but of different sections)

2) Each member **must** participate in the Project.

3) Assign **Team Leader** (responsible for assigning tasks among the group, communicate with the mentor TA, ...etc)

2. Registration Process:

(The form will be announced next lecture and will be open for registration from Wednesday 7/3 till Sunday 11/3 11:59 pm)

1) Choose 3 ideas from the listed Projects for your group([General Projects.pdf](#), [BIOSW Projects.pdf](#)).

2) **REGISTER ONLY ONCE!**

3) Your name should be a member **in ONLY ONE TEAM**..or else you will be removed from all teams.

3) Write the **ARABIC** name and **SEAT Number** for each member.

4) Provide a **valid email** for the team leader (check it regularly for any updates)

5) After registration ends, you will be assigned 1 of the 3 chosen ideas and will be informed a week later(maybe via email).

Projects Regulations

3. External Ideas:

Having your own idea is **MOST WELCOMED**, follow those steps:

1. **Write** a proposal for your idea(using the **template** in the dropbox).
2. **Discuss it** with me (maybe approved and maybe refused).
3. **Sign** your proposal by me for approval (or else it will not be considered) before Wednesday 7/3/2018, contact me to set an appointment)
4. **Register** also in the same form but by choosing the option of “**Other Project**”, you will need to give it a name.



Projects Regulations

4. Mentoring:

A schedule will be announced after assigning the projects having the mentor TA of each project and his/her available support timing.

5. Projects Delivery:

1. All team members must attend project delivery discussion.
2. The week before practical exam (most probably).
3. A schedule will be announced.
4. In the discussion get all the code (make a backup in different resources like flash memories or cds and get it with you).
5. Submitting Project design and documentation (short one, 3-5 pages) is an asset.
6. BEST Projects will be AWARDED 😊
7. LAZY MEMBERS WILL BE PUNISHED! ☹
8. COPIES WILL BE PENALIZED! ☹

Thank
You

