

# Introduction to Programming in C

---

(IS-FEE-10061S)

Białystok University of Technology  
Faculty of Electrical Engineering  
Academic year 2023/2024

**Workshop no. 10 (09.05.2024)**

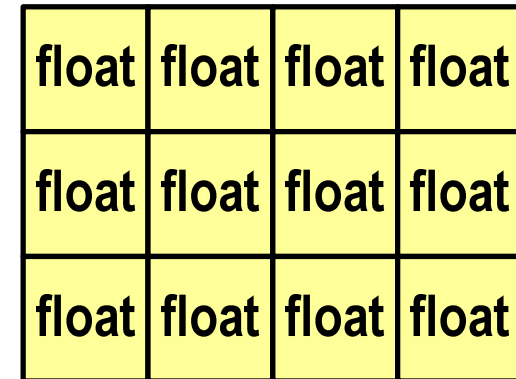
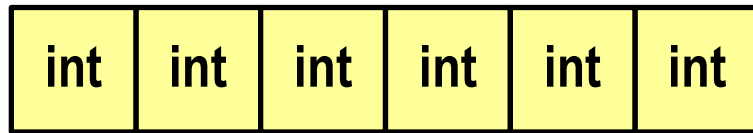
Jarosław Forenc, PhD

# Topics

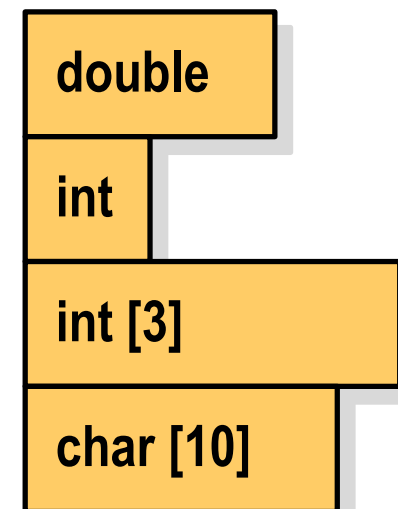
- Structures
  - ❑ a structure declaration, defining a structure variable
  - ❑ access to members of a structure
  - ❑ initialization of a structure variable
  - ❑ complex declarations of structures

# Structures

- **Array** - a series of values of the same type (elements), stored sequentially



- **Structure** - a set of elements of different types, grouped under one name



## Structures: declaration

```
struct name
{
    member_1;
    member_2;
    ...
    member_n;
};
```

```
struct point
{
    int x;
    int y;
};
```

- The elements of a structure are called **members** (or fields)
- Member declarations have the same form as variable declarations
- By declaring a structure, we create a new data type (such as **struct point**), which can be used in the same way as any other standard type

## Structures: declaration

```
struct person
{
    char fname[15];
    char surname[20];
    int age, weight;
};
```

```
struct MyComplex
{
    float Re, Im;
};
```

- Structure declaration does not create an object (does not allocate memory for structure members)
- Storing data in a structure requires defining a **structure variable**

# Structures: defining a structure variable

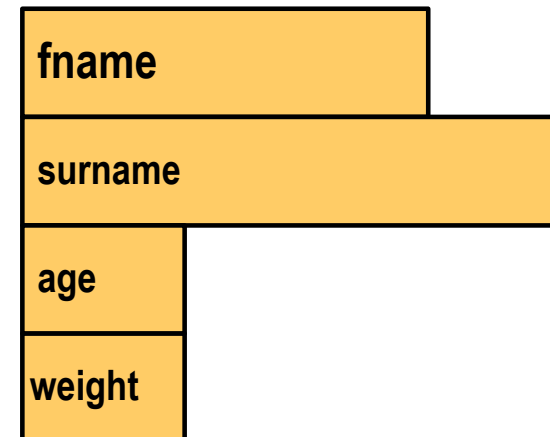
```
#include <stdio.h>

struct person
{
    char fname[15];
    char surname[20];
    int age, weight;
} Smith;

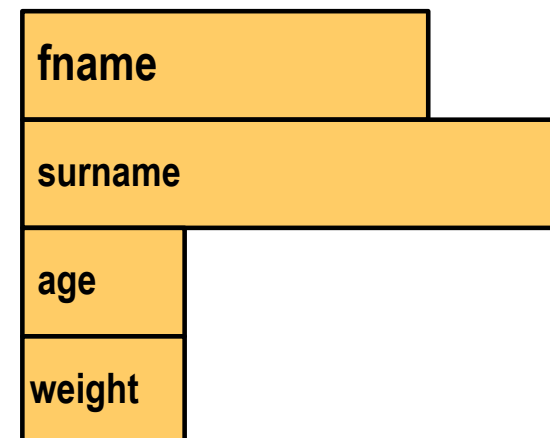
int main(void)
{
    struct person Jones;
    ...
}
```

- Smith, Jones - struct person type variables

Smith



Jones



## Structures: access to members of a structure

- Access to members of a structure is possible by using the notation:

```
structure_variable_name.member_name
```

- The dot operator (.) is called the **direct member selection operator**
- Writing the value **25** to the **age** member of **Jones** structure variable:

```
Jones.age = 25;
```

- The expression **Jones.age** is treated as an **int** variable

```
printf("Age: %d\n", Jones.age);  
scanf("%d", &Jones.age);
```

## Structures: access to members of a structure

- Access to members of a structure is possible by using the notation:

```
structure_variable_name.member_name
```

- The dot operator (.) is called the **direct member selection operator**
- Writing the value **Paul** to the **fname** member of **Jones** variable:

```
strcpy(Jones.fname, "Paul");
```

- The expression **Jones.fname** is treated as a character string

```
printf("First name: %s\n", Jones.fname);  
gets(Jones.fname);
```



## Structures: example

```
#include <stdio.h>

struct person
{
    char fname[15];
    char surname[20];
    int age;
};

int main(void)
{
    struct person Jones;
```

## Structures: example

```
printf("First name:  ");  
gets(Jones.fname);  
  
printf("Surname:    ");  
gets(Jones.surname);  
  
printf("Age:        ");  
scanf("%d", &Jones.age);  
  
printf("%s %s, age: %d\n", Jones.fname,  
      Jones.surname, Jones.age);  
  
return 0;  
}
```

```
First name:  Paul  
Surname:     Jones  
Age:        21  
Paul Jones, age: 21
```

## Structures: initialization

- **Initialization** can only apply to structure variables, initialization of members in a struct declaration is not possible

```
struct person
{
    char fname[15], surname[20];
    int age, weight;
};
```

```
struct person Jones = {"Paul", "Jones", 21, 74};
```

- We can use the assignment operator (=) with structure variables

```
struct person Smith = {"Kate", "Smith", 20, 54};
struct person Smith1;
Smith1 = Smith;
```

# Structures: example

```
#include <stdio.h>

struct date
{
    int day;
    int month;
    int year;
} day1;

int main(void)
{
    struct date day2 = {19, 11, 2018};
}
```

day1

day	?
month	?
year	?

day2

day	19
month	11
year	2018

## Structures: example

```
day1.day = 1;
day1.month = 9;
day1.year = 2018;

printf("Date1: %02d-%02d-%4d\n",
       day1.day, day1.month, day1.year);
printf("Date2: %02d-%02d-%4d\n",
       day2.day, day2.month, day2.year);

return 0;
}
```

day1

day	1
month	9
year	2018

day2

day	19
month	11
year	2018

```
Date1: 01-09-2018
Date2: 19-11-2018
```

# Structures: complex declarations of structures

```
struct point  
{  
    int x;  
    int y;  
} tab[3];
```

tab

0	x	y
1	x	y
2	x	y

```
tab[0].x = 10;  
tab[0].y = 20;  
tab[1].x = 15;  
...
```

```
struct triangle  
{  
    int no;  
    struct point A, B, C;  
} Tr1;
```

Tr1

nr		
A	x	y
B	x	y
C	x	y

```
Tr1.no = 1;  
Tr1.A.x = 10;  
Tr1.A.y = 20;  
Tr1.B.x = 15;  
...
```

End of workshop no. 10

Thank you for your attention!