BASIC PROGRAMMING

STRUCTURED DATA TYPES

Le Thi Ngoc Tho, PhD
Faculty of Information Technology, HUTECH
ltn.tho@hutech.edu.vn

1

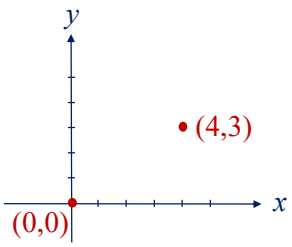
CONTENTS

- Basics of structure
- Structure definition and declaration
- Structure initialization
- The assignment of a structure to another
- Array of structures
- Some algorithms on array of structures

2

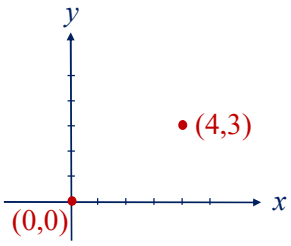
BASICS OF STRUCTURE

- Structure:
  - Collection of one or more variables,
  - Possibly different types,
  - Grouped together in a single name.
- E.g., point object consists of an x coordinate and y coordinate.



BASICS OF STRUCTURE

- Definition:



keywords | structure tag

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

members

BASICS OF STRUCTURE

- Declaration:

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

keywords — structure tag

```
struct POINT pt1;
```

variable name

- Usage: Access to members of struct by dot operator (.)

```
pt1.x = 4;
pt1.y = 3;
```


BASICS OF STRUCTURE

- Initialization:

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

```
struct POINT pt = {3, 4};
```

pt.x = 3 — pt.y = 4

- The values of initialization will be assigned to corresponding members of structure.

BASICS OF STRUCTURE

- Assignment of a structure to another

```
struct POINT A, B;
A.x = 3;
A.y = 4;
B = A;
```

```
B.x = 3
B.y = 4
```

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

- Values of members will be assigned correspondingly.

BASICS OF STRUCTURE

- Example

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

Assign values to members

Declare and initialize

Get values from keyboard

```
int main(){
    struct POINT pt1;
    pt1.x = 1;
    pt1.y = 2;
    printf("point 1: (%d, %d)\n", pt1.x, pt1.y);

    struct POINT pt2 = {2,3};
    printf("point 2: (%d, %d)\n", pt2.x, pt2.y);

    struct POINT pt3;
    printf("Input your prefer coordinate: ");
    scanf("%d%d", &pt3.x, &pt3.y);
    printf("point 3: (%d, %d)\n", pt3.x, pt3.y);

    return 0;
}
```

EXAMPLE

- Problem: Write a program to compute the distance between two points $A(x_A, yA)$ and $B(x_B, yB)$. Given the formula:

$$d(A,B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

```
int main() {
    struct POINT A, B;
    //Input coordinator values for A and B yourself
    long sum = (A.x-B.x)*(A.x-B.x) + (A.y-B.y)*(A.y-B.y);
    float dis = sqrt(sum);
    printf("Distance between two points is %f", dis);

    return 0;
}
```


EXAMPLE

- Problems: Using structure of POINT, write program as following requirements:
  - Input 2 points, check if they are the same
  - Input 3 points, tell us what kind of triangle they form
    - Right triangle
    - Equilateral triangle
    - Isosceles triangle

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

ARRAY OF STRUCTURES

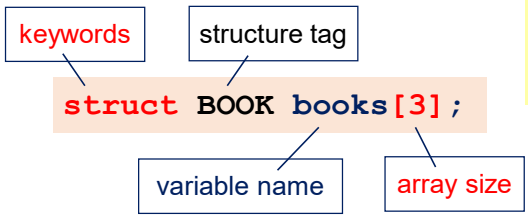
- Problems: Define structure of BOOK, which includes information about: title, price and page. Write functions:
  - Input a book
  - Print information of a book
  - Input an array of books
  - Print an array of books
  - Find a book whose pages is greater than 50
  - Find a book whose name contains “good”

```
struct BOOK{
    char title[10];
    float price;
    int pages;
};
```


ARRAY OF STRUCTURES

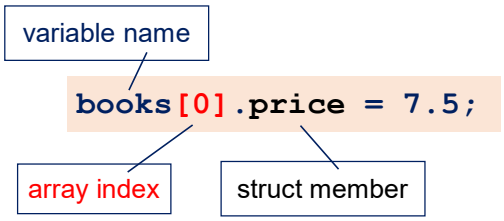
- We can declare and use an array of structure as the same fashion with other data type
- Declare array of structures

```
struct BOOK{
    char title[10];
    float price;
    int pages;
};
```



ARRAY OF STRUCTURES

- We can declare and use an array of structure as the same fashion with other data type
- Access to member of array element:




```
struct BOOK{
    char title[10];
    float price;
    int pages;
};
```


ARRAY OF STRUCTURES

- We can declare and use an array of structure as the same fashion with other data type

```
int main(){
  struct BOOK books[3];
  for(int i = 0; i < 2; i++){
    printf("Book %d: \n", i+1);
    fflush(stdin);
    gets(books[i].name);
    scanf("%f", &books[i].price);
    scanf("%d", &books[i].pages);
    printf("%s %f %d\n", books[i].name,
           books[i].price, books[i].pages);
  }
  return 0;
}
```

```
struct BOOK{
    char title[10];
    float price;
    int pages;
};
```

Any Questions?