Object Oriented Programming

Shaobai Kan

chapter 22

Outline

Outline

- Structure

Outline

- Structure
- Array of struct data types

Structure

Structure

Definition. Structure are aggregate data type, i.e., they can be built using elements of several types including other *structs.*

Structure

Definition. Structure are aggregate data type, i.e., they can be built using elements of several types including other *structs.*

e.g.

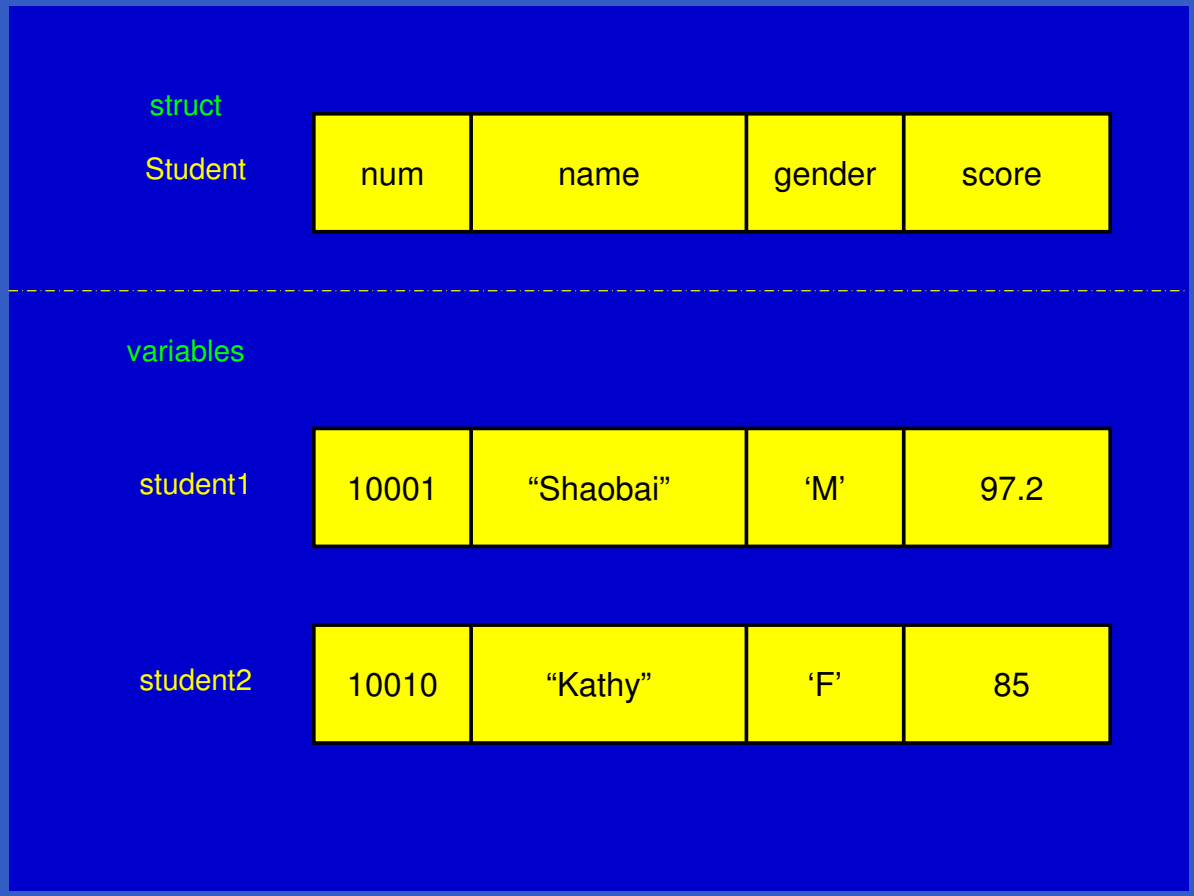
```
struct Student
{
    int number;
    char name[20];
    char gender;
    int age;
    float score;
};
```

Declare and initialize structure variables

Example 1

```
# include <iostream>
using namespace std;

struct  Student
{
        int  num;
        char  name[20];
        char  gender;
        double  score;
} ;

int  main ( )
{
        // Declare and initialize two student type variables
        Student student1 = { 10001, “Shaobai”, ‘M’, 97.2};
        Student student2 = { 10010, “Kathy”, ‘F’, 85};

        cout << student1.num << ‘\t’ << student1.name << ‘\t’
             << student1.gender << ‘\t’ << student1.score << endl;

        return 0 ;
}
```

Example 1: Visual representation

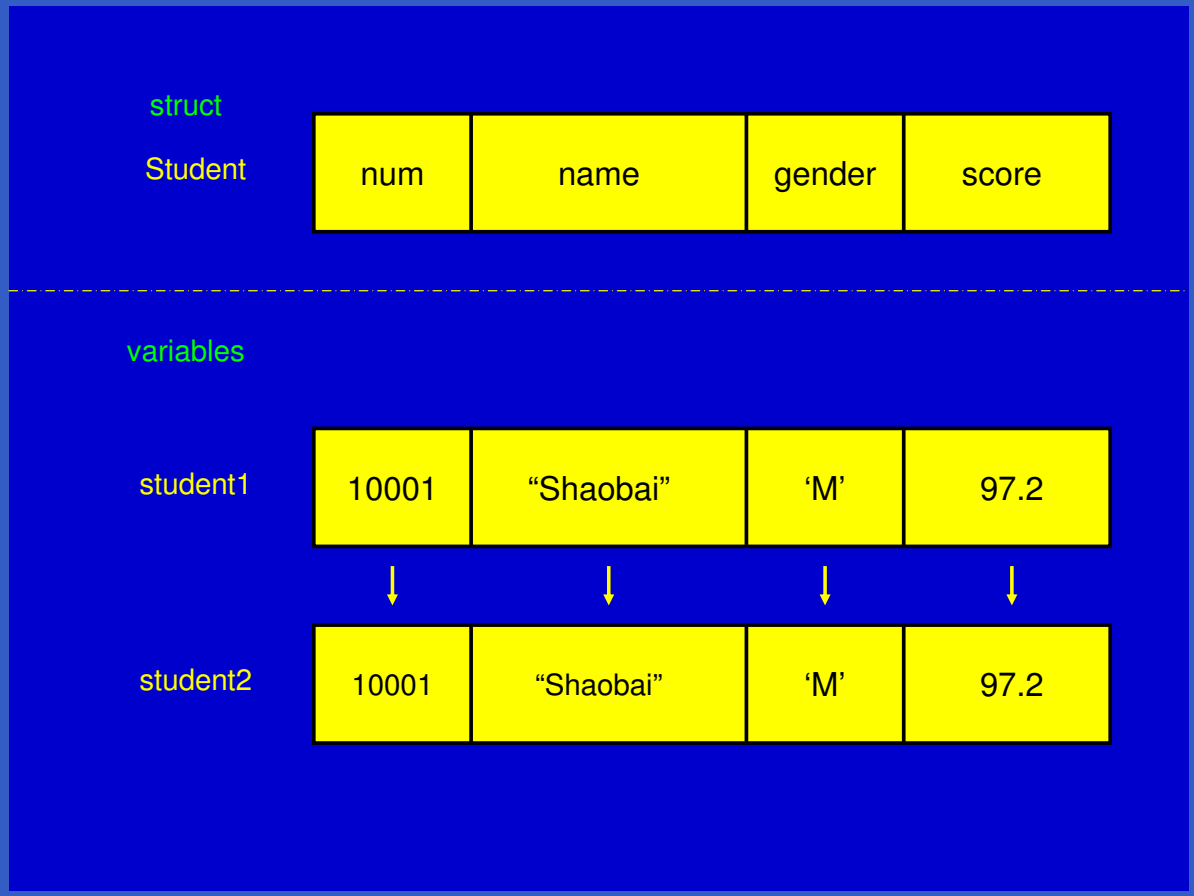


Declare and initialize structure variables

Example 2

```
 # include <iostream>
using namespace std;

struct  Student
{
        int  num;
       char  name[20];
        char  gender;
        double  score;
} student1= {10001, "Shaobai", 'M', 97.2}, student2 ;


 int  main ( )
{
       student2 = student1;

       cout << student2.num << '\t' << student2.name << '\t'
            << student2.gender << '\t' << student2.score << endl;

       return 0 ;
 }
```

Example 2: Visual representation



Assign values to structure members

Syntax error!!!

```
# include <iostream>
using namespace std;

struct  Student
{
        int  num;
       char  name[20];
        char  gender;
       double  score;
} ;

int  main ( )
{
      Student student1;
       student1 = { 10001, “Shaobai”, ‘M’, 97.2};

       cout << student1.num << ‘\t’ << student1.name << ‘\t’
            << student1.gender << ‘\t’ << student1.score << endl;

       return 0 ;
}
```

Watch out!
Syntax error

Assign values to structure members

Correct code!!!

```
# include <iostream>
using namespace std;
struct  Student
{
        int  num;
       char  name[20];
        char  gender;
        double  score;
} ;

int  main ( )
{
       Student student1;
       student1.num = 10001;
       strcpy_s (student1.name, "Shaobai");
        student1.gender = 'M';
        student1.score = 97.2;
       cout << student1.num << '\t' << student1.name << '\t'
            << student1.gender << '\t' << student1.score << endl;
       return 0 ;
}
```

strcpy *v.s.* strcpy_s

Functions: strcpy (strcpy_s) *v.s.* strncpy

strcpy. Function strcpy copies its second argument—a string — into its first argument — a character array that must be large enough to store the string and its terminating null character.

Functions: strcpy (strcpy_s) *v.s.* strncpy

strcpy. Function strcpy copies its second argument—a string — into its first argument — a character array that must be large enough to store the string and its terminating null character.

strncpy. Function strncpy is much like strcpy, except that strncpy specifies the number of characters to be copied from the string into the array.

strcpy (strcpy_s) *v.s.* strncpy

Example 3

```
# include <iostream>
using namespace std;

int  main ( )
{
        char x[ ] = "Happy Birthday to You" ;
        char y[ 25 ] ;
        char z[ 15 ] ;

        strcpy ( y, x );

        cout << "The string in array x is: " << x
             << "\nThe string in array y is: " << y << '\n';

        //copy first 14 characters of x into z
        strncpy ( z, x, 14);
        z[ 14 ] = '\0';
        cout << "The string in array z is: " << z << endl;

        return 0 ;
}
```

Example3: Output

```
The string in array x is : Happy Birthday to You
The string in array y is : Happy Birthday to You
The string in array z is : Happy Birthday
```

Array of struct data types

Example 4: array of struct data types

Array of **struct** datatypes

```
# include <iostream>
using namespace std;

struct  Candidate
{
      char  name[20];
      int  count;
} ;

int  main ( )
{
      Candidate leader[3] = {{“John”, 0 }, {“Mike”, 0}, {“Thomas”, 0} };
      char candidate_name[20];
      for ( int i = 0; i < 10; i++ )
      {
            cin >> candidate_name
            for ( int j = 0; j < 3; j++ )
                  if ( strcmp (candidate_name, leader[ j ].name) == 0)
                        leader[ j ].count ++ ;
      }
      cout << endl ;

      for ( int i = 0 ; i < 3 ; i++)
          cout << leader[ i ].name << “:” << leade[ i ].count << endl;
}
```

Example 4: Application

```
John ↙
Mike ↙
John ↙
John ↙
Thomas ↙
Mike ↙
John ↙
Thomas ↙
Mike ↙
John ↙

John:5
Mike:3
Thomas:2
```

strcmp function

strcmp. Function strcmp compare its first string argument with its second string argument character by character.

- the 1st string = the 2nd string, return 0
- the 1st string < the 2nd string, return negative value
- the 1st string > the 2nd string, return positive value

strcmp function

strcmp. Function strcmp compare its first string argument with its second string argument character by character.

- the 1st string = the 2nd string, return 0
- the 1st string < the 2nd string, return negative value
- the 1st string > the 2nd string, return positive value

e.g.

- "Boy" > "Axle"
- "Happy Holiday" < "Happy New Year"

Homework:

- Read Sec. 22.1, 22.2, 22.3, 22.4
- Exercise 22.6, 22.7