Walkthrough (TA said after topic 8)

Bookmark this page

In C++, struct and class are exactly the same things, except for the fact that struct defaults to public visibility and class defaults to private visibility.

Some important differences between the C and C++ structures:

1. Member functions inside the structure:

Structures in C cannot have member functions inside structure but structures in C++ can have member functions along with data members.

2. Direct Initialization:

We cannot directly initialize structure data members in C but we can do it in C++.

```
// C program to demonstrate that direct
// member initialization is not possible in C
#include<stdio.h>

struct Record
{
   int x = 7;
};

// Driver Program
int main()
{
    struct Record s;
    printf("%d", s.x);
    return 0;
}
/* Output :  Compiler Error
   6:8: error: expected ':', ',', ';', '}' or
  '__attribute__' before '=' token
  int x = 7;
        ^
  In function 'main': */

// CPP program to initialize data member in c++
#include<iostream>
usingnamespacestd;

structRecord
{
    intx = 7;
};

//  Driver Program
intmain()
{
    Record s;
    cout << s.x << endl;
    return0;
}
```

```
// Output
// 7
```

3. Using struct keyword:

In C, we need to use struct to declare a struct variable. In C++, struct is not necessary. For example, let there be a structure for Record. In C, we must use “struct Record” for Record variables. In C++, we need not use struct and using ‘Record‘ only would work.

```
// C program
#include<stdio.h>

struct Record
{
   int x;
};

// Driver Program
int main()
{
    struct Record s;
    return 0;
}

// CPP program
#include<iostream>
usingnamespacestd;

structRecord
{
    intx;
};

//  Driver Program
intmain()
{
    Record s;
    return0;
}
```

4. Static Members:

C structures cannot have static members but it is allowed in C++.

```
// program with structure static member
structRecord
{
    staticintx;
};

// Driver program
intmain()
{
    return0;
}
/* This will generate an error in C but not in C++*/
5. sizeof operator:
```

This operator will generate 0 for an empty structure in C whereas 1 for an empty structure in C++.

```
// program to illustrate empty structure
#include<stdio.h>
//empty structure
struct Record
{
};

//Driver program
int main()
{
    struct Record s;
    printf("%d\n",sizeof(s));
    return 0;
}
```

Output in C: 0

Output in C++: 1

6. Data Hiding:

C structures do not allow the concept of Data hiding but it is permitted in C++ as C++ is an object oriented language whereas C is not.

7. Access Modifiers:

C structures does not have access modifiers as these modifiers are not suppoted by the language. C++ structures can have this concept as it is inbuilt in the language.