# struct and class

## struct in c and c++

1. `In c, functions cannot be declared (and defined) directly in the body of struct`
2. `In c, the data member cannot be initialized in the struct`

```
typedef struct A{
    int foo();  // error
    int (*pf)(void); // ok ,pf is a pointer to function that take no parameter returning int type
    int a;      // ok
    int b=0;    // error
    char *s;    // ok
    char * string ="string";// error
};
```

3. `In c++, the only difference between struct and class is the default access level, therefore there is no restrictions like that in c.`
4. `We can implement the method that is one of concepts in class by using struct in c.`

**demo**

```
#include <iostream>
#include <string>
struct HELLO {
    public:
    void sayHello(const std::string name) {
        std::cout << "Hello " << name << std::endl;
    }
};
int main() {
    //HELLO hello = HELLO();
    //hello.sayHello("World!");
    //equivalent statement to the following
    HELLO *hello = new HELLO();
    hello->sayHello("World!");
    delete(hello);
}
```

**In c, we can do that as the following form.**

```
#include <stdio.h>
#include <stdlib.h>// malloc
struct HELLO{
    // sayHello is a pointer to function
    void (*sayHello)(const char *name);
    //void *sayHello(const char *name); //error: sayHello has function type.
};
void sayHello(const char *name){
        printf("Hello ,%s\n",name);
}
int main(){
    //struct HELLO hello;
    //hello.sayHello=sayHello;
    //hello.sayHello("World!");
    struct HELLO *hello=(struct HELLO *)malloc(sizeof(*hello));
    hello->sayHello=sayHello;
    hello->sayHello("World!");
    free(hello);
}
```