
دوره برنامه نویسی C++ پیشرفته

دانشگاه ولی عصر رفسنجان

مهندس حسین بازماندگان

جلسه چهارم

کلاس ها و اشیاء بخش دوم

سرفصل مطالب

- قرار دادن کلاس در فایل جداگانه
- اعضای کلاس با ویژگی static
- ارسال اشیاء به توابع
- انتساب اشیاء به یکدیگر
- آرایه ای از اشیاء

قرار دادن کلاس در فایل جداگانه

- می توانیم هر کلاس را به صورت جداگانه در یک فایل با پسوند `<.h>` قرار دهیم این کار باعث سازماندهی بهتر کدها می شود.

قرار دادن کلاس در فایل جداگانه

```
#ifndef PERSON_H
#define PERSON_H
#include <string>
using namespace std;
class Person {
public:
    string name;
    int id;
};
#endif //PERSON_H
```

 CMakeLists.txt

 main.cpp

 Person.h

قرار دادن کلاس در فایل جداگانه

- برای استفاده از کلاس باید آن را با دستور `#include` به برنامه اضافه کنیم.

```
#include "Person.h"
```

```
int main() {  
    Person john;  
    return 0;  
}
```

قرار دادن کلاس در فایل جداگانه

- برای استفاده از کلاس باید آن را با دستور `#include` به برنامه اضافه کنیم.

```
#include "Person.h"
```

```
int main() {  
    Person john;  
    return 0;  
}
```

قرار دادن کلاس در فایل جداگانه

- برای استفاده از کلاس باید آن را با دستور `#include` به برنامه اضافه کنیم.

```
#include "Person.h"
```

```
int main() {  
    Person john;  
    return 0;  
}
```


اعضای استاتیک چیست؟

- اعضای استاتیک (Static Members) متعلق به خود کلاس هستند، نه به یک نمونه خاص از آن. این بدان معناست که:
 - فقط یک نسخه از یک عضو استاتیک در حافظه وجود دارد، که توسط همه نمونه‌های کلاس به اشتراک گذاشته می‌شود.
 - می‌توانید به اعضای استاتیک بدون ایجاد شیء (object) دسترسی پیدا کنید، به شرطی که دسترسی عمومی (public) باشد.
 - برای تعریف یک عضو استاتیک، از کلمه کلیدی **static** استفاده می‌شود.

انواع اعضای استاتیک

- متغیرهای استاتیک (Static Variables):
 - یک متغیر استاتیک در سطح کلاس تعریف می‌شود و مقدار آن بین همه نمونه‌ها مشترک است.
 - باید خارج از بدنه کلاس مقداردهی اولیه شود (به جز در صورت استفاده از سازنده درون خطی).

متغیر استاتیک برای شمارش دانشجویان // `static int studentCount;`

انواع اعضای استاتیک

- توابع استاتیک (Static Functions):

- تابع استاتیک می‌تواند بدون نیاز به یک نمونه از کلاس فراخوانی شود.
- این توابع فقط می‌توانند به اعضای استاتیک دیگر (متغیرها یا توابع) دسترسی داشته باشند، نه به اعضای غیراستاتیک.

```
تابع استاتیک برای نمایش تعداد و اطلاعات //  
static void displayCount() {  
    cout << "Total students: " << studentCount << endl;  
    cout << "Last added student: " << lastAddedName << endl;  
    cout << "Maximum possible grade: " << MAX_GRADE << endl;  
}
```

اعضای استاتیک

- نکات مهم
- مقداردهی اولیه:
- متغیرهای استاتیک باید خارج از کلاس مقداردهی اولیه شوند، زیرا حافظه آنها در زمان کامپایل تخصیص می‌یابد.
- دسترسی:
- اعضای استاتیک با `ClassName::memberName` قابل دسترسی هستند.
- اگر عمومی باشند، می‌توانید از طریق شیء هم به آنها دسترسی پیدا کنید، اما بهتر است از نام کلاس استفاده کنید.

اعضای استاتیک

- نکات مهم
- محدودیت توابع استاتیک:
- توابع استاتیک نمی‌توانند به اعضای غیراستاتیک (مثل `name` در مثال بالا) دسترسی داشته باشند، چون به یک نمونه خاص وابسته نیستند.
- کاربردها:
- شمارش تعداد اشیاء
- ذخیره مقادیر ثابت یا مشترک بین همه نمونه‌ها (مثل حداکثر نمره).



ارسال اشياء به توابع

ارسال اشیاء به توابع

```
void changeName(Student stu, string name) {  
    stu.name = name;  
}
```

ارسال اشیاء به توابع

```
void changeName(Student&stu, string name) {  
    stu.name = name;  
}
```


انتساب اشیاء به یکدیگر

```
Student s1("Ali",23,19.5);
```

```
Student s2;
```

```
s2 = s1;
```

آرایه ای از اشیاء

```
Student stus[10];
```

```
for (int i = 0; i < 10; i++) {
```

```
    cin >> stus[i].name;
```

```
    cin >> stus[i].age;
```

```
    cin >> stus[i].gpa;
```

```
}
```

برگرداندن اشیاء توسط تابع

```
Student readStudent() {  
    Student stu;  
    cin >> stu.name;  
    cin >> stu.age;  
    cin >> stu.gpa;  
    return stu;  
}
```