

آموزش C++ - درس ۴- ورودی و خروجی

به سری آموزشهای C++ خوش آمدید. این درس به شما چیزهایی درباره اساس ورودی و خروجی در C++ یاد خواهد داد و کلاس string را معرفی می کند. معمولا I/O (ورودی و خروجی) یکی از مهمترین قسمت های هر برنامه ای است. برای انجام هر کار مفیدی برنامه شما باید بتواند داده های ورودی را دریافت کرده و نتیجه را گزارش دهد. در C++ ورودی و خروجی بوسیله کتابخانه iostream فراهم شده است. برای استفاده از این کتابخانه، به بالای برنامه خودتان عبارت `#include <iostream>` را اضافه کنید. یعنی پیش کمپایلر کد را از فایل `iostream.h` به فایل منبع شما اضافه می کند. ضمیمه کردن این فایل اشیایی را که مشاهده می کنید برای استفاده در برنامه تعریف می کند.

- **cin** : این شی برای ورودی از ترمینال (صفحه کلید) تدارک دیده شده است.
- **cout** : این شی برای خروجی به صفحه نمایش تدارک دیده شده است.
- **cerr** : این شی برای خروجی بافر نشده به ابزار خطای استاندارد، که بطور پیش فرض صفحه نمایش است، تدارک دیده شده است. بافر نشده به این معنی است که هر پیغام یا داده ای بطور مستقیم می تواند نوشته شود. با ورودی بافر شده، داده بوسیله سیستم عامل در یک بافر آشکار برای برنامه شما ذخیره می شود. وقتی بافر پر است هر چیزی در آن نوشته شده خارج می شود. این بسیار مفید است زیرا هر نوشتن یک سرپار برای سیستم عامل است. خارج کردن یک بافر بزرگ سرپار کمتری از خارج کردن چند پیغام کوچک دارد. اگر یک برنامه قبل از اینکه بافر نوشته شده باشد با شکست مواجه شود، در بافر خروجی وجود ندارد. خروجی از طریق **cerr** بافر نشده است تا مطمئن باشید که پیغام های خطا نوشته خواهند شد.
- **clog** : این شی برای خروجی بافر شده برای ابزار خطای استاندارد، که معمولا صفحه نمایش است، تدارک دیده شده است.

بهترین راه برای یاد گرفتن روش استفاده از هر شی، استفاده از مثال و تمرین است.

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

int main()
{
    string name;
    int ID;

    cout << "Enter your name ";
    cin >> name;

    cout << "Enter your ID number ";
    cin >> ID;

    cout << "Hello " << name << " or should I say " << ID << endl;
    return 0;
}
```

C++ یک کلاس string را به عنوان بخشی از کتابخانه استاندارد تعریف کرده است. برای استفاده از این کلاس، شما باید فایل کتابخانه ای "string" را به فایل منبع خودتان ضمیمه کنید. برای استفاده از کلاس string شما باید یک شی string تعریف کنید. کلاس تعیین می کند که چه اعضای در آن وجود دارد و چه متدهایی دارد. اعضا برای ذخیره داده های کلاس بکار می روند، متدها اعضای تابعی کلاس را تهیه می کنند، که نشان می دهد که چه کاری می تواند انجام دهد. این مطالب بطور کامل و واضح در درس های آینده بررسی خواهند شد. فقط همانطور که ID یک شی از نوع داده int است، name هم یک شی از نوع داده string است. برنامه فوق را برای تمرین کمپایل و اجرا کنید.

عملگر ورودی >> برای قرار دادن داده در متغیرها در برنامه شما بکار می رود. اغلب آن را عملگر الحاق می نامند. عملگر خروجی << برای خروجی مستقیم به خروجی استاندارد یا ابزار خطای استاندارد استفاده می شود. اغلب آن را عملگر استخراج می نامند. همچنین این عملگر ها با دیگر جریانه ها (استریمها) از جمله فایلها بکار می روند. ورودی و خروجی فایل در درس های آینده بررسی خواهد شد.

تمرین

همانطور که قبلا نیز تاکید کرده بودم، بهترین راه برای یادگیری نوشتن کدهای واقعی است. بنابراین در اینجا تعدادی تمرین برای شما داده ایم.

(۱) برنامه مثال را برای ورود و خروج نام و نام خانوادگی اصلاح کنید.

```
using namespace std;

int main()
```

```
{
    string firstName;
    string lastName;
    int ID;

    cout << "Enter your first name ";
    cin >> firstName;

    cout << "Enter your last name ";
    cin >> lastName;

    cout << "Enter your ID number ";
    cin >> ID;

    cout << "Hello " << firstName << " " << lastName;
    cout << " or should I say " << ID << endl;

    return 0;
}
```

۲) برنامه را برای ورود و خروج اطلاعات دیگر توسعه دهید. شامل سن، شغل، حقوق و آدرس. برای هر کدام از این متغیرهای جدید چه نوع داده ای مناسب است؟

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

int main()
{
    string firstName;
    string lastName;
    int ID;
    int age;
    string job;
    float salary;

    cout << "Enter your first name ";
    cin >> firstName;
    cout << "Enter your last name ";
    cin >> lastName;

    cout << "Enter your ID number ";
    cin >> ID;

    cout << "Enter your age ";
    cin >> age;

    cout << "Enter you occupation and salary ";
    cin >> job >> salary;

    cout << "Hello " << firstName << " " << lastName;
    cout << " or should I say " << ID << endl;
    cout << "Age: " << age << endl;
    cout << "Occupation: " << job << endl;
    cout << "Salary: " << salary << endl;

    return 0;
}
```