# **فصل 3**

## **محدودیت نام‌ها**

در زبان برنامه‌نویسی C++, متغیرها باید با رعایت یک سری قوانین نام‌گذاری شوند. برخی از محدودیت‌های نام‌گذاری متغیرها در C++ عبارتند از:  
  
1. تشکیل نام: متغیرها باید با یک حرف Aتا Z یا a تا z یا \_ شروع شوند.  
  
2. استفاده از حروف و ارقام: می‌توانید از حروف انگلیسی بزرگ و کوچک، اعداد (0 تا 9) و زیرخط در نام متغیرها استفاده کنید.  
  
3. حساس به حروف بزرگ و کوچک: C++ حساس به حروف بزرگ و کوچک است، بنابراین "myVariable" و "MyVariable" به عنوان دو متغیر مجزا در نظر گرفته می‌شوند.  
  
4. استفاده از کلمات کلیدی نه: نمی‌توانید از کلمات کلیدی مانند if، else، while و... به عنوان نام متغیرها استفاده کنید.

5. حداکثر تعداد کاراکتر مجاز نام متغیرها: معمولاً این حد در محدوده ۳۱ تا ۶۳ کاراکتر قرار دارد، اما برخی کامپایلرها ممکن است حداکثر تعداد کاراکترها را محدودتر کنند.

6. معنا داشتن: نام متغییرها باید توصیفی و مفهومی باشند تا برنامه‌نویسان بتوانند به راحتی معنی متغییر ها را درک کنند.

## **فضای نام‌ها**

در زبان برنامه‌نویسی C++، namespace یک مفهوم است که برای گروه‌بندی عناصری مانند کلاس‌ها، توابع، و متغیرها به یک دامنه (دسته) خاص استفاده می‌شود. این امکان را فراهم می‌کند تا نام‌های مختلف در دامنه‌های مختلف قرار گیرند و از تداخل نام‌ها جلوگیری شود. از namespace برای سازماندهی و مدیریت کد در برنامه‌های بزرگ استفاده می‌شود.  
مثال:

// namespace تعریف یک   
namespace example {  
    // تعریف یک کلاس در دامنه

    class MyClass {  
    public:  
        void myFunction();  
    };  
  
    // تعریف یک تابع در دامنه   
    void myFunctionOutsideClass();  
}  
  
// تعریف یک تابع خارج از دامنه

void someFunction() {  
    // ...  
}  
int main() {  
    // namespace example ایجاد یک شی از کلاس در   
    example::MyClass obj;  
  
    // example فراخوانی تابع در دامنه   
    obj.myFunction();  
  
    // namespace فراخوانی تابع خارج از   
    someFunction();  
  
    // example فراخوانی تابع در دامنه   
    example::myFunctionOutsideClass();  
  
    return 0;  
}

استفاده از namespace به تقسیم و ارتقاء قابلیت‌ها و خوانایی کد کمک می‌کند، به ویژه زمانی که کدها به حجم بزرگ می‌رسند و احتمال تداخل نام‌ها افزایش می‌یابد.

## **انقیاد**

در سی پلاس پلاس هر دو نوع انقیاد(انقیاد پویا، انقیاد ایستا) وجود دارد. مفهوم انقیاد پویا به وسیله مفهوم پلی مورفیسم و "virtual functions" پیاده‌سازی می‌شود. وقتی یک تابع را به عنوان virtual تعریف کنید و در کلاس پایه آن را پیاده‌سازی کنید، می‌توانید توابع مشابه در کلاس‌های مشتق شده را با استفاده از کلمه کلیدی

Override بازنویسی کنید. در زمان اجرا، با استفاده از آدرس‌های تابع مجازی (virtual function table)، تصمیم گرفته می‌شود که کدام تابع باید فراخوانی شود. این عملیات به عنوان انقیاد پویا شناخته می‌شود.  
  
مثال:

#include <iostream>  
  
class Base {  
public:  
    virtual void show() {  
        std::cout << "Base class\n";  
    }  
};  
  
class Derived : public Base {  
public:  
    void show() override {  
        std::cout << "Derived class\n";  
    }  
};  
  
int main() {  
    Base\* obj = new Derived();  
    obj->show();  // Dynamic binding: calls the derived class function  
    delete obj;  
    return 0;  
}

در انقیاد ایستا، از constبرای تعریف یک متغیر یا شیء که مقدار آن پس از اولین تخصیص نمی‌تواند تغییر کند، استفاده می‌شود.

مثال:

#include <iostream>  
  
int main() {  
    const int myConstantNumber = 42;

    // myConstantNumber = 10; error

    std::cout << "مقدار متغیر ایستا: " << myConstantNumber << std::endl;  
  
    return 0;  
}

همچنین می‌توانید از کلمه کلیدی constexpr نیز استفاده کنید. این کلمه کلیدی مخصوصاً برای تعریف ایستاها در زمان کامپایل (compile-time) استفاده می‌شود.  
مثال:

constexpr int myConstValue = 42;

مهم‌ترین تفاوت کلمه کلیدی const و constexpr به شرح زیر است:

1. زمان تصمیم‌گیری:

Const : انقیاد ایستا با const بر روی متغیرها و اشیاء در زمان اجرا اعمال می‌شود.

Constexpr: این کلمه کلیدی برای ایجاد انقیاد ایستا در زمان کامپایل استفاده می‌شود و بیشتر برای متغیرها و توابع constexpr مناسب است.  
2. استفاده در توابع:

Const: می‌تواند بر روی متغیرها و توابع اعمال شود.

Constexpr: اغلب برای تعریف متغیرها و توابع در زمان کامپایل مورد استفاده قرار می‌گیرد.  
3.محدودیت‌ها:

Const: مقدار یک const در زمان اجرا تعیین می‌شود و معمولاً برخی محدودیت‌ها در تعیین مقدار دارد.  
Constexpr: برای تعیین مقدار در زمان کامپایل، مقدارها و عبارات باید در زمان کامپایل قابل ارزیابی باشند.

همچنین کلمات کلیدی دیگری برای این انقیاد استفاده می‌شوند. البته هر کدام از این کلمات کلیدی دارای مفهوم و استفاده‌های خود هستند و به تنهایی به ایجاد انقیاد ایستا اشاره ندارند.

1.volatile:

برای اطلاع از اینکه مقدار یک متغیر ممکن است به صورت غیرمنتظره تغییر کند (مثل توسط سخت‌افزار یا نخ‌های موازی)، استفاده می‌شود. این کلمه کلیدی نشان‌دهنده این است که کامپایلر نباید بهینه‌سازی‌هایی انجام دهد که ممکن است با خصوصیات غیرمنتظره تغییرات در مقدار متغیر تداخل داشته باشد.

   volatile int sensorValue = 0;

۲ :Restrict.

در اصطلاح C99 برای بهینه‌سازی کارهای که از اشاره‌گرها استفاده می‌کنند، مورد استفاده قرار می‌گیرد. این کلمه کلیدی به کامپایلر اعلام می‌کند که اشاره‌گرها به یک متغیر restrict تنها توسط آن اشاره دارند و سایر اشاره‌گرها نمی‌توانند تداخل ایجاد کنند.

   void exampleFunction(int\* restrict ptrA, int\* restrict ptrB);

## صریح یا ضمنی

متغییر ها در سی پلاس پلاس صریح و ضمنی میتواند باشد.   
مثلا auto  بصورت ضمنی ست همچنین نوع داده مشخص نیست از template استفاده میکنیم.  
۴ نوع  را Static\_stack-dynamic\_ explicit heap-dynamin\_ implicit heap-dynamic داریم. که در سی پلاس پلاس به شرح زیر است تمام مقادیر سی پلاس از نوع دوم است همچنین  در فراخوانی بازگشتی متغییر ها داریم که با هر بار فراخوانی به پشته اضافه میشه. در صورت اضافه کردن کلمه static متغییر اینستا میشود.

Static data\_type var\_name = var\_value;

سومی وقتی اندازه ارایه را نمیدانیم از حافظه هیپ استفاده میکنیم. یا ساخت لینک لیست چون حداکثر آن مشخص نیست. لذا نیاز است فضای زیادی را به اختصاص دهیم که بدلیل زیاد بودن آن نیاز داریم نامنظم اختصاص دهیم. با استفاده از new حافظه تخصیص میدهیم و با استفاده از delete ان حافظه آزاد میشود.

pointer-variable = new data-type;

آخری رو cpp نداره.