

اشاره‌گرها در زبان C++

مقدمه اشاره‌گرها یکی از مفاهیم کلیدی در زبان C++ هستند که به برنامه‌نویسان این امکان را می‌دهند تا به صورت مؤثرتری با حافظه کار کنند. در این جزوه، به معرفی، کاربرد، و روش‌های استفاده از اشاره‌گرها خواهیم پرداخت.

۱. تعریف اشاره‌گر

اشاره‌گر (Pointer) یک متغیر است که آدرس یک متغیر دیگر را در حافظه نگه می‌دارد. به عبارت دیگر، اشاره‌گرها به شما اجازه می‌دهند تا به داده‌ها به صورت غیرمستقیم دسترسی پیدا کنید.

۲. نمونه تعریف اشاره‌گر

برای تعریف یک اشاره‌گر، ابتدا نوع داده‌ای که می‌خواهید به آن اشاره کنید را مشخص کرده و سپس با علامت * به تعریف اشاره‌گر می‌پردازید.

```
// اشاره‌گری به نوع int* ptr;
```

۳. مثال از ایجاد و استفاده از اشاره‌گر

```
1#include <iostream>
2using namespace std;
3
4int main() {
5    int var = 20; // یک متغیر معمولی
6    int* ptr = &var; // اشاره‌گر ptr به آدرس var اشاره می‌کند
7
8    cout << "آدرس متغیر var: " << &var << endl; //
9    cout << "آدرس نگهداری شده در ptr: " << ptr << endl; //
10   cout << "مقدار var از طریق اشاره‌گر *ptr: " << *ptr << endl; //
11
12   return 0;
13}
```

فرومی:

آدرس متغیر var: 0x7ffeefbfff588

آدرس نگهداری شده در ptr: 0x7ffeefbfff588

مقدار var از طریق اشاره‌گر: ۲۰

۴. عملگرها

- عملگر (& آدرس): برای به دست آوردن آدرس یک متغیر استفاده می شود.
- عملگر * (مقدار دهی): برای دسترسی به مقداری که اشاره گر به آن اشاره می کند، استفاده می شود.

۵. اشاره گرهای خالی (Null Pointers)

اشاره گرهای خالی اشاره گرهایی هستند که به هیچ آدرسی اشاره نمی کنند. بهتر است همیشه لزوم مقداردهی اولیه به اشاره گرها را رعایت کنید.

```
int* ptr = nullptr; // اشاره گر خالی
```

۶. اشاره گرها و آرایه ها

اشاره گرها می توانند به آرایه ها اشاره کنند. در ++C، نام آرایه خود به خود به اشاره گر به اولین عنصر آرایه تبدیل می شود.

```
int arr[] = {10, 20, 30};
int* ptr = arr; // ptr به اولین عنصر آرایه اشاره می کند
cout << *ptr << endl; // 10 خروجی:
cout << *(ptr + 1) << endl; // 20 خروجی:
```

۷. اشاره گرهای تو در تو

می توان اشاره گرهایی به اشاره گرهای دیگر نیز تعریف کرد:

```
int var = 30;
int* ptr = &var; // اشاره گر اول
int** ptr2 = &ptr; // اشاره گر دوم به اشاره گر اول
cout << **ptr2; // 30 خروجی:
```

۸. نکات مهم

- برای مدیریت مطلوب حافظه، در هنگام استفاده از اشاره گرها دقت کنید و از هرگونه دسترسی به آدرس های نامعتبر جلوگیری کنید.
- delete[] و delete برای آزادسازی حافظه ای که با new [] و new تخصیص داده شده است، استفاده کنید.

نتیجه‌گیری اشاره‌گرها ابزارهای قدرتمندی در ++C هستند که با استفاده صحیح از آنها می‌توانید مدیریت حافظه
بهتری داشته باشید و کارایی کد خود را افزایش دهید.