

زبان برنامه نویسی C++ چیست؟

C++ زبانی پیچیده، کارآمد و جامع است که بر پایه ی C طراحی و در سال ۱۹۷۹ توسط بی یارنه استراس تروپ توسعه داده شده است.

بسیاری از سیستم عامل های فعلی، سیستم درایور ها، مرورگرها و بازی ها از C++ به عنوان زبان اصلی خود استفاده می کنند که باعث می شود C++ از محبوب ترین زبان های امروز باشد. از آن جایی که نسخه پیشرفته و تعمیم یافته زبان C است، C++ و C اغلب به عنوان C/C++ نام گذاری می شوند.

تاریخچه C++

زمانی که بی یارنه استراس تروپ در سال ۱۹۷۹ در AT&T Bell Labs کار می کرد با مشکلاتی در زمینه تحلیل کرنل UNIX برای سیستم های توزیع شده مواجه شد. زبان آن یا بسیار کند و یا بسیار سطح پایین بود. از این رو تصمیم گرفت زبان جدیدی ایجاد کند.

وی برای ساخت این زبان جدید C را انتخاب کرد. دلیل این مسئله آن بود که C یک زبان عمومی، کارآمد و سریع بود. او از دانش خود در زمینه مدل شی گرای زبان SIMULA استفاده کرد و کار روی بسط کلاس ها به زبان C را آغاز نمود. هدف او ایجاد زبانی بود که سطوح بالاتری از تجرید را داشته باشد و همزمان کارایی زبان C نیز حفظ شود.

زبان برنامه نویسی جدید به صورت C with Classe نام گذاری شد اما در ادامه به C++ تغییر کرد. (منظور از ++ اشاره به عملگر افزایشی در زبان C دارد).

ویژگی های C++

C++ به علت عمومی بودن دارای قابلیت های متنوعی است. پرداختن به همه این قابلیت ها کاری زمان بر است. در این جا برخی از قابلیت های مهم این زبان را که باید بدانید ارائه داده ایم.

۱ - C++-سریع است: از آنجا که C++ نسخه بسط یافته ای از C است، بخش زبان C آن کاملاً سطح پایین است. این امر موجب ارتقای زیادی در سرعت آن شده است که زبان های سطح بالا مانند پایتون یا جاوا نمی توانند ارائه کنند.

۲ - C++ دارای نوع بندی صریح است: C++: یک زبان با بیان نوع صریح است، یعنی این زبان به کامپایلر امکان حدس نوع داده ها را نمی دهد. برای نمونه در این زبان نمی توانید عدد ۱۰ را به صورت "۱۰" بنویسید و انتظار داشته باشید که C++ متوجه شود رشته است. بدین ترتیب کامپایلر می تواند خطاهای موجود در نوع داده را پیش از اجرای برنامه تشخیص دهد.

۳ - C++ زبان برنامه نویسی چند پارادایمی است: C++ از حداقل ۷ سبک مختلف برنامه نویسی پشتیبانی می کند و توسعه دهنده ها آزادی عمل انتخاب هر یک را به دلخواه دارند. برخلاف جاوا و پایتون در این زبان مجبور نیستید از اشیا برای حل همه وظایف استفاده کنید. شما می توانید سبک برنامه نویسی که با خواست شما سازگارتر است انتخاب کنید.

۴ - برنامه نویسی شی گرا با C++ امکان پذیر است: برنامه نویسی شی گرا به حل مسائل پیچیده کمک می کند. با بهره گیری از مفاهیم شی گرایی در C++ می توان مسائل پیچیده را با ساخت اشیا به مسائل کوچک تری تقسیم کرد.

۵ - کتابخانه استاندارد قدرتمند (Standard template library – STL): قدرت C++ با بهره گیری از کتابخانه های استاندارد که در آن وجود دارد افزایش می یابد. این کتابخانه ها شامل الگوریتم های مفیدی هستند که می توان در زمان کد نویسی از آن ها استفاده کرد.

۵ دلیل که چرا باید C++ یاد بگیرید

این یکی از سوالاتی است که قبل از شروع هر زبان برنامه نویسی باید پرسید و کمک می کند تا حوزه زبان را درک کنید، قابلیت استفاده در دنیای واقعی را بدانید و تا چه حد می توانید از آن در زمینه پشتیبانی استفاده کنید. در اینجا ۵ دلیل برای یادگیری C++ آورده شده است.

۱ - غیر قابل جایگزین است

با توجه به کاربرد زبان برنامه نویسی C++ در بازی های مدرن، سیستم های عامل، مرورگرها و غیره می توان به جرات گفت که غیر قابل جایگزینی است. بسیاری از برنامه های کاربردی با استفاده از C++ ایجاد شده اند:

- محصولات Adobe مانند فتوشاپ، ایلستریاتور، ایندیزاین
- آمازون - یکی از بزرگ ترین سایت های تجارت الکترونیک
- محصولات Autodesk برای طراحی کامپیوتری
- فیسبوک - یکی از بزرگ ترین شبکه های اجتماعی

به علاوه این واقعیت نیز وجود دارد که یک جامعه بزرگ در حال بهبود ++C هستند و انتظار می رود استفاده از این زبان در آینده بیشتر هم شود.

۲- معماری داخلی کامپیوتر را یاد می گیرید

++C زبان سطح متوسط است، کدهایی که با آن نوشته می شوند، مستقیماً با سخت افزار داخلی کامپیوتر کار می کنند به همین دلیل با نحوه ی کار حافظه، نحوه ی ذخیره اطلاعات در آن، نوع بازیابی اطلاعات و... آشنا می شوید و دانش معماری کامپیوتر شما افزایش می یابد.

۳- بیش از ۶۰۰۰۰۰ مخزن ++C در گیت هاب وجود دارد

گیت هاب بزرگ ترین سایت همکاری منبع باز دنیاست. در آن بیش از ۶۰۰۰۰۰ مخزن فقط برای ++C وجود دارد. این معیار به تنهایی اهمیت ++C را ثابت می کند. شما در هر زمینه ای کار کنید، هزاران پروژه منبع باز مرتبط با موضوعتان وجود دارد.

۴- نرخ پاسخ دهی فعال ۶۰ درصد در StackOverflow

سایت StackOverflow بزرگ ترین منبع پرسش و پاسخ برنامه نویسی دنیاست. بیش از ۴۰۰۰۰۰ سوال در مورد ++C و بیش از ۶۰ درصد پاسخ برای آن ها وجود دارد. تعداد سوالات و درصد پاسخ آن ها نشان دهنده ی جامعه ی فعال توسعه دهندگان ++C است. پس می توان انتظار داشت توسعه دهندگان زیادی وجود دارد که در صورت بروز مشکل در زمینه ی ++C به شما کمک کنند.

۵- فرصت های شغلی و دستمزد در ++C

بر اساس آمار، توسعه دهندگان ++C میانگین حقوق ۱۰۰۰۰۰ دلار در سال دارند و بیش از ۷۷۰۰ موقعیت شغلی ماهانه برای آن ها وجود دارد. بیشتر موقعیت های شغلی در زمینه های بازی سازی، موتور های رندرینگ و برنامه های ویندوزی پیشنهاد می شوند.

۴ چیزی که قبل از کدنویسی ++C باید بدانید

۱- **یک روزه نمی توان آن را آموخت** یادگیری هر زبانی به زمان احتیاج دارد و در مورد ++C مشهود تر است. اگر فکر می کنید که می توانید در یک روز آن را یاد بگیرید، در انتهای روز نا امید خواهید شد. صادقانه می توان گفت هیچ زمانی را برای یادگیری کامل ++C نمی توان گفت. می بایست زمان زیادی را صرف یادگیری آن بکنید.

۲- **یادگیری آن می توان دشوار باشد** چون ++C زبان سطح بالایی نیست، امکان دارد در ابتدا یادگیری آن مشکل باشد و مجبور شوید ساعت ها وقت خود را صرف تمرکز برای یادگیری بکنید.

۳- **احتیاجی به یادگیری C قبل از C++ نیست** بعضی افراد فکر می کنند که قبل از C++ باید C را یاد بگیرند، اما چنین الزامی وجود ندارد. می توانید یادگیری را از C++ شروع کنید و هیچ مشکلی هم پیش نخواهد آمد. اگر از قبل با C آشنایی داشته باشید، یادگیری برایتان آسان تر خواهد بود زیرا خصوصیت های مشابهی از نظر ساختار و معنا با هم دارند.

۴- **منتظر انتشار نسخه ی بعدی C++ نمانید** بسیاری از افراد می پرسند آیا بهتر نیست صبر کنند تا نسخه ی جدید منتشر شود و سپس شروع به یادگیری کنند؟ پاسخ نه است. با اینکه به نسخه هایی که در آینده منتشر می شوند امکانات زیادی اضافه می شود اما مفاهیم اساسی یکی هستند. پس بهتر است زمان خود را هدر ندهید و از همین آآن یادگیری را شروع کنید.

اولین برنامه C++

اگر کامپایلر را بر مبنای سیستم عامل خود نصب کرده اید. زمان آن رسیده که نخستین برنامه خود را بنویسید.

“Hello World!”

نخستین برنامه C++ برنامه Hello World! خواهد بود.

این برنامه ای است که در همه زبان های برنامه نویسی به عنوان نخستین برنامه نوشته می شود؛ زیرا:

- این برنامه یک رویه استاندارد برای بررسی صحیح بودن همه چیز است.
- در آغاز به کد نویسی کمی نیاز هست.
- این کد برای افراد مبتدی فهم آسان تری دارد و بهتر می تواند با زبان آشنا شوند.
- این کد برای یادگیری ساختار ابتدایی و معنا شناختی زبان کافی است.

پس شروع به کد نویسی می کنیم:

```

۱. #include <iostream>
۲. using namespace std;
۳. int main()
۴. {
۵. cout<<"Hello World!";
۶. return 0;
۷. }
```

این برنامه عبارت Hello World! را در خروجی چاپ می کند.

برنامه چطور کار می کند؟

اکنون نوبت به بررسی کد بالا رسیده است. این کد را به شش قسمت می توان تقسیم کرد:

- `#include <iostream>`
- `using namespace std`
- `;`
- `int main(){ }`
- `cout << "Hello World!";`
- `return 0;`

۱ `#include <iostream>` - به چه معناست؟

اگر قبلاً با زبان C کار کرده باشید احتمالاً این خط کد را دیده اید. اگر چنین نیست به توضیحاتی که در ادامه آمده توجه کنید.

این گزاره فایل هدر را در اپلیکیشن می گنجاند به طوری که می توانیم از عملیات موجود در آن استفاده کنیم. ضمناً می توانید فایل های هدر خاص خود را ایجاد کرده و آن ها را با استفاده از `#include` در برنامه بگنجانید.

`iostream` به چه معناست؟

`iostream` نام فایل هدر است. فایل کتابخانه استاندارد ورودی/خروجی C++ است. این کتابخانه به همراه کامپایلر IDE می آید و شامل مکانیزم هایی است که اطلاعات را از کاربر می گیرد و خروجی را در یک فایل، صفحه نمایش یا هر رسانه دیگری نمایش می دهد.

`include` چیست؟

برای گنجاندن فایل `iostream` در برنامه از دستور `#include` استفاده می کنیم. این دستور تضمین می کند که می توان از عملیات های فایل `iostream` مانند گرفتن ورودی از کاربر، نمایش خروجی روی صفحه در برنامه استفاده کرد.

۲- عبارت `using namespace std;` به چه معناست؟

به کمک آن از فضای نامی به نام `std` در برنامه استفاده می کنیم. ما از `namespace std` برای تسهیل ارجاع به عملیات موجود در آن فضای نام بهره می گیریم. اگر از این فضای نام استفاده نکنیم، باید به جای `cout` از `std::cout` استفاده کنیم. این دستور به کامپایلر اعلام می کند که هر `cout` در واقع `std::cout` است.

namespace یعنی چه؟

کد برنامه اینجا قرار می‌گیرد. این گزینه دامنه کد را به یک یا چند فایل محدود کرده یا گسترش می‌دهد.

چرا باید از namespace استفاده کرد؟

همان طور که دو نفر ممکن است نام مشابه داشته باشند، متغیرها و توابع هم در C++ ممکن است نام‌های مشابه داشته باشند. استفاده از namespace برای جلوگیری از تداخل متغیرها و توابع و ارجاع صحیح به هر کدام از آنها است.

std چیست؟

std فضای نام استاندارد مورد استفاده در C++ است.

۳- سمی کالن یا نقطه ویرگول؛

اگر از هر برنامه نویسی پیرسید، دست کم یک داستان ترسناک در مورد نقطه ویرگول برایتان تعریف می‌کند. نقطه ویرگول کاراکتر پایانی محسوب می‌شود و گزاره را می‌بندد. زمانی که آن را فراموش کنید یا به صورت نادرستی از آن استفاده کنید، مشکلات زیادی به وجود می‌آورد.

۴ int main(){} - به چه معناست؟

تابع اصلی برنامه است. کد درون {} است که body نامیده می‌شود و زمانی که برنامه را اجرا کنید، قبل از همه اجرا می‌شود. تنها کدی است که در برنامه‌های C++ وجود آن اجباری است. اگر تنها این کد را در برنامه داشته باشید، از نظر کامپایلر برنامه معتبر است.

"Hello World!" << cout -

این خط عبارت Hello World! را در صفحه خروجی نمایش می‌دهد cout. شیء استاندارد stream خروجی است؛ یعنی وظیفه آن چاپ داده‌های موجود (به عنوان مثال Hello world!) پس از << در یک stream است.

stream چیست؟

stream به یک توالی از اشیا و یا معمولاً بایت‌ها گفته می‌شود. این توالی می‌تواند فایل، ترمینال ورودی/خروجی، سوکت و... را توصیف کند.

>> به چه معنا است؟

>> عملگر درج است که برای نوشتن داده های قالب بندی شده در stream استفاده می شود.

۶- معنی ۰؛ return چیست؟

این گزاره مقدار ۰ را بازگشت می دهد. گزاره های return اجباری نیستند و الزامی برای بازگشت دادن یک مقدار از تابع main() وجود ندارد، اما نوشتن آن یک قرارداد است. اگر این گزاره ذکر نشود، کامپایلر به طور خودکار یک وضعیت را بازگشت می دهد.

چرا در گزاره بازگشت از ۰ استفاده می شود؟

مقدار ۰ به معنی وضعیت Exit از اپلیکیشن است که به سیستم اعلام می کند برنامه کار خود را به پایان رسانده است.