

C++

C++ تاریخچه زبان برنامه نویسی

در سال 1967 شخصی بنام آقای "مارتین ریچاردز" برای نوشتن سیستم عامل و طراحی کامپایلر زبان جدیدی بنام BCPL طراحی نمود.

سه سال بعد، یعنی در سال 1970، "کن تامپسون" با افزودن ویژگی های جدیدی به زبان BCPL زبان جدیدی طراحی نمود و نسخه های اولیه سیستم عامل unix را با آن طراحی نمود. او این زبان را B نامید.

BCPL و B هر دو از زبانهای بدون نوع بودند. به همین جهت بررسی یک عنصر داده ای، دلیل وظیفه سنگینی بر عهده ی برنامه نویس بود. در این زبان هر عنصر داده ای یک کلمه از حافظه، معادل 2 بایت، را اشغال می کرد.

دو سال بعد، یعنی در سال 1972، آقای "دنيس ریچی" زبان جدیدی را بر پایه زبان B طراحی نمود. از آنجاییکه این زبان پس از زبان B طراحی شده بود، زبان C نام گرفت. زبان C بسیاری از ویژگی ها و مفاهیم زبانهای BCPL و B را دارا بود و ویژگی های دیگری، از جمله انواع داده ای، را نیز به آنها افزوده بود

زبان C مستقل از سخت افزار است، به همین دلیل با طراحی خوب و دقیق نرم افزار می توان برنامه هایی نوشت که دارای قابلیت حمل در اکثر کامپیوترها باشند.

قدرت زبان C باعث شد که شرکت های مختلفی در صدد طراحی کامپایلرهای C برآیند.

این موضوع مشکل جدیدی برای طراحان برنامه هایی بود که می خواستند نرم افزارهای قابل حملی بنویسند که روی کامپیوترهای مختلفی اجرا

شود. 19 سال پس از طراحی زبان C، "کمیته استاندارد ملی آمریکا"، ANSI، نسخه ی استاندارد زبان C را تصویب نمود.

گامی به عقب :

در اواخر دهه 1970 هنگامی که "بیارنه استراس تروپ" بر روی تز دکترایش در دانشگاه کمبریج انگلستان کار می کرد مجبور بود برنامه های خود را به زبان Simula بنویسد. Simula از Algol مشتق شده و برای اولین بار مفاهیم کلاس به عنوان اشیا داده ای در این زبان (Simula) مطرح شده بود.

کلاس در Simula، جرقه ی طراحی و ساخت کامپایلر C++ را در ذهن "بیارنه" شعله ور نمود. او پس از مهاجرت از انگلستان به آمریکا و کار در شرکت AT&T تعمیم و گسترش زبان C را آغاز کرد. زبان جدید در سال 1980 طراحی شده و در 1982 با استقبال فراوانی روبرو شده بود.

این زبان با عناوین متفاوتی از جمله "New C" و "C With Classes" و "C84" مشهور بود! یکی از همکاران "استراس تروپ" بنام Rick Mascitti نام C++ را برای آن برگزید.

17 سال بعد از طراحی C++ یعنی در نوامبر سال 1997 C++ نیز استاندارد شد.

C++ زبانی دو رگه است! چنانچه این قابلیت را دارد که هم به صورت "پیمانه ای" و هم بصورت "شی گرا" نرم افزارهای قابل اعتمادی را ایجاد نماید. چند سال بعد زبان جاوا نیز بر پایه ی C و C++ طراحی شد.

- کتابخانه استاندارد C++

برنامه های C++ از قطعاتی به نام کلاس ها و توابع تشکیل می شود. شما در عین حال که برای برنامه نویسی به این قطعات از پیش ساخته شده نیاز دارید، می توانید این قطعات را نیز خودتان بسازید "کتابخانه استاندارد C++" دارای کلاس های و توابع زیادی می باشد که مجموعه ای غنی، جهت برنامه نویسی را در اختیار برنامه نویسان قرار می دهد.

بنابراین برای یادگیری زبان C++ باید به دو امر مهم همت گماشت:

1- یادگیری ساختار و نحو زبان C++

2- یادگیری کلاس ها و توابع از پیش ساخته شده، در "کتابخانه ی استاندارد C++".

تذکر : برای آموختن دقیقتر این توابع و کلاس ها می توانید به کتاب `The Draft Standard C++ Library` مراجعه نمایید.

- پایه و اساس یک محیط نمونه برنامه نویسی C++

طراحی برنامه در C++ اغلب شامل 5 مرحله می باشد :

1- ویرایشگر : شامل تایپ کدهای برنامه در ویراستار C++ و ذخیره بر روی هارد دیسک با پسوند *.C و یا C++. [Editor]

2- پیش پردازنده : افزودن فایل های متنی مورد نیاز، در فایلی که قرار است کامپایل شود

[PreProcessor]

- 3- کامپایلر : ترجمه ی کدهای برنامه به زبان ماشین توسط کامپایلر و ساختن فایل *.Obj [Compiler]
- 4- پیوند دهنده : ساختن فایل اجرایی (*.Exe) و افزودن تمام چیزهایی که برای اجرای برنامه بدون حضور به کامپایلر لازم است.[Linker]
- 5- بار کننده : انتقال کدهای زبان ماشین از روی دیسک سخت در حافظه ی اصلی کامپیوتر. [Loader]

عملگرها و متغیرها

- عملیات ریاضی :
در زبان C++ بیش از هفتاد عملگر وجود دارد!
تعدادی از این عملگرها ماهیت ریاضی دارند. در این قسمت به بررسی این عملگرها خواهیم پرداخت.
تعدادی از عملگرهای زبان C++ مربوط به عملگرهای مقایسه ای می شود. این نوع عملگرها تقریباً در تمام زبانهای برنامه نویسی یافت می شوند.
- عملگرهای درج در جریان و استخراج از جریان:
در زبان C++ دو عملگر خاص وجود دارند که جهت کنترل صفحه نمایش بکار می روند.
این عملگرها برای نوشتن داده ها در صفحه نمایش و خواندن داده ها از کاربر بکار می رود.
- انواع داده ای و تعریف متغیر:
متغیر چیست؟ فضا ای از حافظه ی اصلی، که در اختیار برنامه نویس قرار می گیرد . کامپایلر از این فضا محافظت می کند که سایر برنامه های در حال اجرا مقدار آن را تغییر ندهند
تذکر : برای کسب اطلاعات بیشتر در مورد انواع متغیرها، می توانید به راهنمای کامپایلر خود مراجعه نمایید.
تعریف کردن یک متغیر در زبان C++ بسیار ساده تر از آن است که فکرش را می کنید!
ابتدا نام نوع را می نویسیم و سپس نام متغیر!

int a ;

float f ;

double d ;

char ch ;

چهار متغیر به ترتیب از نوع «اعداد صحیح»، «اعداد اعشاری»، «اعداد حقیقی بزرگ» و «کاراکتری» تعریف شده است.

ساختار یک برنامه ی C++

در صورتی که یک برنامه ی C++ به طرز صحیحی طراحی و پیاده سازی شود، قابلیت خوانایی آن با برنامه هایی که با سایر زبان ها نوشته می شوند قابل مقایسه نیست ! شاید مهمترین مزیت C++ نسبت به C و سایر زبانهای برنامه نویسی خاصیت شی گرایی آن باشد . این ویژگی که انقلابی در عرصه ی تولید نرم افزار تولید کرده است، در عین سادگی برای مبتدیان پیچیدگی خاصی دارد. شاید اصلی ترین دلیل پیچیدگی، همان سادگی اش باشد!

حقیقت این است که برای یادگیری C++ باید ابتدا C را آموخت. در واقع می توان گفت که :

Classes C++

با یادگیری زبان C و سپس افزودن مباحث مربوط به کلاس ها به معلومات خویش می توان به C++ مسلط شد . برای یادگیری زبان قدرتمند C++ ، به ناچار باید با معلومات جامع و کاملی از C++ و ترجیحا یک زبان ویژوال داشت و سپس سراغ این اعجوبه قرن رفت.

بررسی ساختار یک برنامه به زبان

در این قسمت می خواهیم ساختار عمومی یک برنامه به زبان C++ را بیاموزیم.

قطعا با معلوماتی که تا به اینجا آموختید درصد قابل قبولی از ساختار یک برنامه ی C++ را درک خواهید کرد:

- 1) `#include <iostream.h>`
- 2) **Functions Prototype**
- 3) `void mian()`
- 4) `{`
- 5) **The Programme Body (Declaration Variable, Call Functions & Objects)**
- 6) `};`
- 7) **Implementation Functions**

تذکر : شماره گذاری ها برای درک ساده تر می باشد. بدیهی است که در یک برنامه شماره گذاری خطوط منجر به خطای دستوری (Error Syntax) خواهد شد.

حال به بررسی مختصر کتابخانه در این برنامه می پردازیم.

```
#include <iostream.h>
```

این خط (که اولین خط برنامه ی ++C هم می باشد) در واقع یک دستور پیش پردازنده است!

در واقع تمام خطوطی که با # شروع می شوند یک دستور پیش پردازنده هستند

این دستور به "پیش پردازنده" می گوید که :

"محتویات فایل `iostream.h` را به برنامه اضافه کن."

در داخل فایل `iostream.h` دستورات مربوط به ورودی و خروجی قرار دارد. در واقع اگر برنامه ما بخواهد از دستورات ورودی و خروجی استفاده نماید باید ابتدا فایل `iostream.h` را به کمک دستور پیش پردازنده به بدنه برنامه اضافه نماید و آنگاه بتواند از توابع درون آن استفاده نماید.

این سبک عملکرد دقیقاً مانند به کار بردن توابع API در زبان ویژوال بیسیک می باشد. در زبان ویژوال بیسیک ابتدا باید به مفسر بیسیک بگوییم که از کدام فایل های DLL می خواهیم استفاده نماییم پس از آن است که می توانیم در بدنه برنامه از توابع آن فایل DLL استفاده کنیم.

نکته : به فایل هایی مثل `iostream.h` «سرفایل» یا «هدر فایل» می گویند.

پایان