



درس «مبانی کامپیوتر و برنامه‌سازی»

آشنایی با زبان C

صادق علی اکبری

# سرفصل مطالب

---

- تاریخچه زبان C
- درباره زبان C++
- اولین برنامه به زبان C
- فرایند اجرای یک برنامه به زبان C

# تاریخچه زبان C

- سال ۱۹۶۷ : مارتین ریچاردز (دانشگاه کمبریج) زبان BCPL را معرفی کرد
  - برای نوشتن نرم افزارهای سیستم عامل و کامپایلر
- سال ۱۹۷۰ : کن تامپسون زبان B را بر مبنای ویژگی های زبان BCPL نوشت
  - از آن برای ایجاد اولین نسخه های سیستم عامل یونیکس در آزمایشگاه بل استفاده کرد
- سال ۱۹۷۲: زبان C توسط دنیس ریچی از روی زبان B و BCPL در آزمایشگاه بل ساخته شد
  - ریچی از این زبان برای بازنویسی سیستم عامل یونیکس استفاده کرد
  - زبان C با سرعت بسیاری گسترش یافت
  - بعدها اکثر سیستم عامل های دیگر نیز با همین زبان نوشته شدند
- چاپ کتاب The C Programming Language در سال ۱۹۷۸ توسط برایان کرنیگان و ریچی باعث رشد روزافزون این زبان در جهان شد.

# ایجاد زبان C

- تولید زبان C به تولید سیستم عامل یونیکس گره خورده است
- تامپسون و ریچی در تولید سیستم عامل یونیکس همکار بودند
- ریچی طراح اصلی زبان C است
- تامپسون بعدها (۲۰۰۷) در گوگل از طراحان زبان Go بود



Thompson (left) with Dennis Ritchie.



# نسخه‌های مختلف زبان C

- نتیجه استفاده گسترده از زبان C : نسخه‌های مختلفی از این زبان به وجود آمد
- که با یکدیگر سازگار نبودند
- سال ۱۹۸۹ در مؤسسه استانداردهای ملی آمریکا (ANSI) :
  - یک تعریف فاقد ابهام و مستقل از ماشین را از این زبان تدوین شد
  - این استاندارد تحت عنوان ANSI C به تصویب رسید
- سال ۱۹۹۰: سازمان بین‌المللی استانداردسازی (ISO) نیز این استاندارد را پذیرفت
- مستندات مشترک آنها تحت عنوان ANSI/ISO C منتشر شد

# زبان C++

- ظهور روش‌های برنامه‌نویسی شیء‌گرا
- اوایل دهه ۱۹۸۰: ارائه نسخه جدیدی از زبان C به نام C++
- توسط بی‌یارنه استراس‌تروپ (دانمارکی) در آزمایشگاه‌های بل
- در C++ علاوه بر امکانات جدید، خاصیت شیء‌گرایی به C اضافه شد
- زبان C++ ابتدا C With Classes خوانده می‌شد
  - بعدها به C++ تغییر نام داد
- این زبان برنامه‌نویسی در سال ۱۹۹۸ استاندارد شد
- ISO/IEC 14882:1998
- نسخه فعلی استاندارد این زبان (سال ۲۰۱۴)
- ISO/IEC 14882:2014 (informally known as C++14)



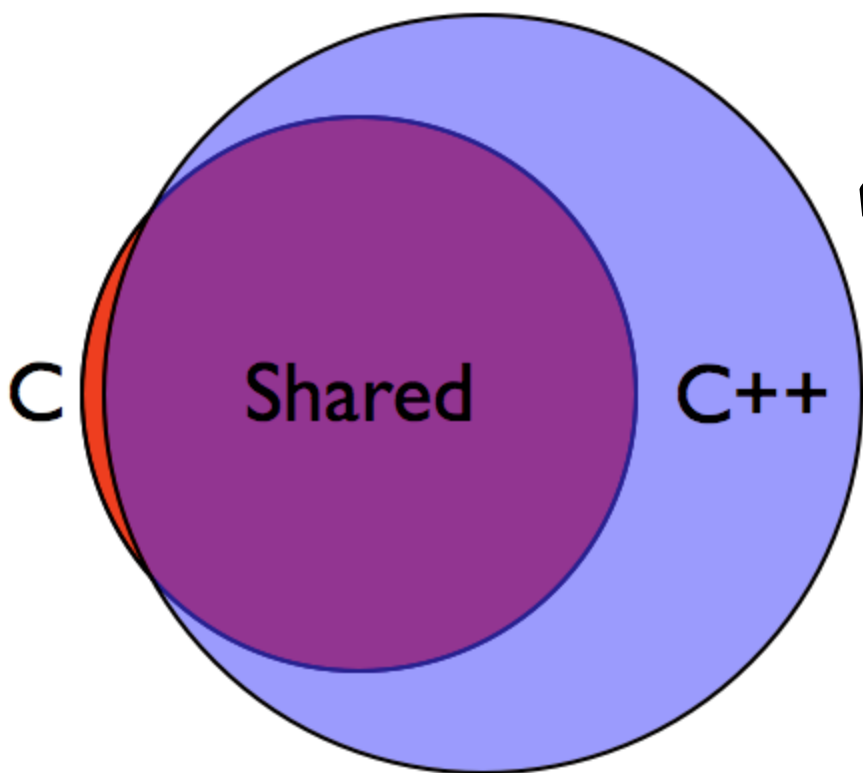
Bjarne Stroustrup, the creator of C++

# استانداردهای زبان C++

Year	C++ Standard	Informal name
1998	ISO/IEC 14882:1998 <sup>[13]</sup>	C++98
2003	ISO/IEC 14882:2003 <sup>[14]</sup>	C++03
2007	ISO/IEC TR 19768:2007 <sup>[15]</sup>	C++07/TR1
2011	ISO/IEC 14882:2011 <sup>[5]</sup>	C++11
2014	ISO/IEC 14882:2014 <sup>[16]</sup>	C++14
2017	to be determined	C++17

# موضوع این درس

- معمولاً از بخشی از زبان C++ استفاده می‌کنیم که مشترک با C است
- از ابزارها و کامپایلرهای C++ استفاده می‌کنیم
- ولی امکانات و مفاهیم شیء‌گرا را مطالعه نمی‌کنیم





# اهمیت زبان C و C++

1	^	Java
2	v	C
3	^	C++
4	^	C#
5	^	Python
6	^	PHP
7	^	JavaScript
8	^	Visual Basic .NET
9	^	Perl
10	v	Objective-C

- زبانهای مهم و پرستفاده

- الهامبخش بسیاری از زبانهای مهم بعدی

- Java, C#, PHP, JavaScript, ...

- ساختار و نحو این زبانها مشابه C و C++ است

# سطح زبان C و C++

- زبان‌های سطح پایین (مثل زبان Assembly)

- نزدیک به زبان ماشین

- برنامه‌ها، مستقیماً یا تقریباً به زبان ماشین (صفر و یک) نوشته شوند

- زبان‌های سطح بالا (مثل جاوا و C و C++ و C#)

- نزدیک به زبان انسان

- برخی، زبان‌های C و C++ را در سطح میانی می‌دانند (Middle level)

- هم قدرت زبانهای سطح پایین را دارند و هم امکانات زبان‌های سطح بالا را پشتیبانی می‌کنند

# اولین برنامه در زبان C

# اولین برنامه به زبان C

- فایلی (مثلاً با نام hello.c) بسازید
- برنامه‌های C معمولاً پسوند .c دارند
- برنامه‌های C++ معمولاً پسوند .cpp دارند
- متن زیر را در این فایل کپی کنید
- نتیجه اجرای برنامه:
- پیغام زیر نمایش داده می‌شود:

```
#include <stdio.h>
int main() {
    printf("Hello World!");
    return 0;
}
```

Hello World!

- یک «برنامه کامپیوتری»، مجموعه‌ای از دستورالعمل‌ها است

- که توسط کامپیوتر اجرا می‌شود

```
#include <stdio.h>
int main()
{
    printf("Salam.");
    printf("Khoobi?");
    printf("Khodanegahdar");
    return 0;
}
```

- برای نوشتن یک برنامه C به این موارد احتیاج داریم:

- یک «ویرایشگر متنی» برای نوشتن متن برنامه

- مثلاً Notepad یا VIM

- یک «کامپایلر» برای ترجمه برنامه به زبان ماشین (تا قابل اجرا شود)

- مثلاً gcc یا Visual C++ (Microsoft Visual Studio)

```
#include <stdio.h>
int main()
{
    printf("Hello World!");
    return 0;
}
```

# معنای دستورالعمل‌های اولین برنامه

- `#include <stdio.h>`
- یک دستور پیش‌پردازشی
- نشان می‌دهد که می‌خواهیم از توابع `stdio` استفاده کنیم
- `int main()`
- یک تابع معرفی می‌کند
- بدنه هر تابع با «آکولاد باز» شروع می‌شود و با «آکولاد بسته» پایان می‌پذیرد
- تابع اصلی که اجرای برنامه از آن آغاز می‌شود، `main` نام دارد
- `printf("Hello World!");`
- یک دستور ساده که یک «رشته» را در خروجی نمایش می‌دهد
- دقت: بیشتر دستورات زبان C با سمیکالون پایان می‌پذیرند
- `return 0;`
- پایان اجرای تابع `main` را اعلام می‌کند
- خروجی صفر به معنای اجرای موفقیت‌آمیز برنامه است

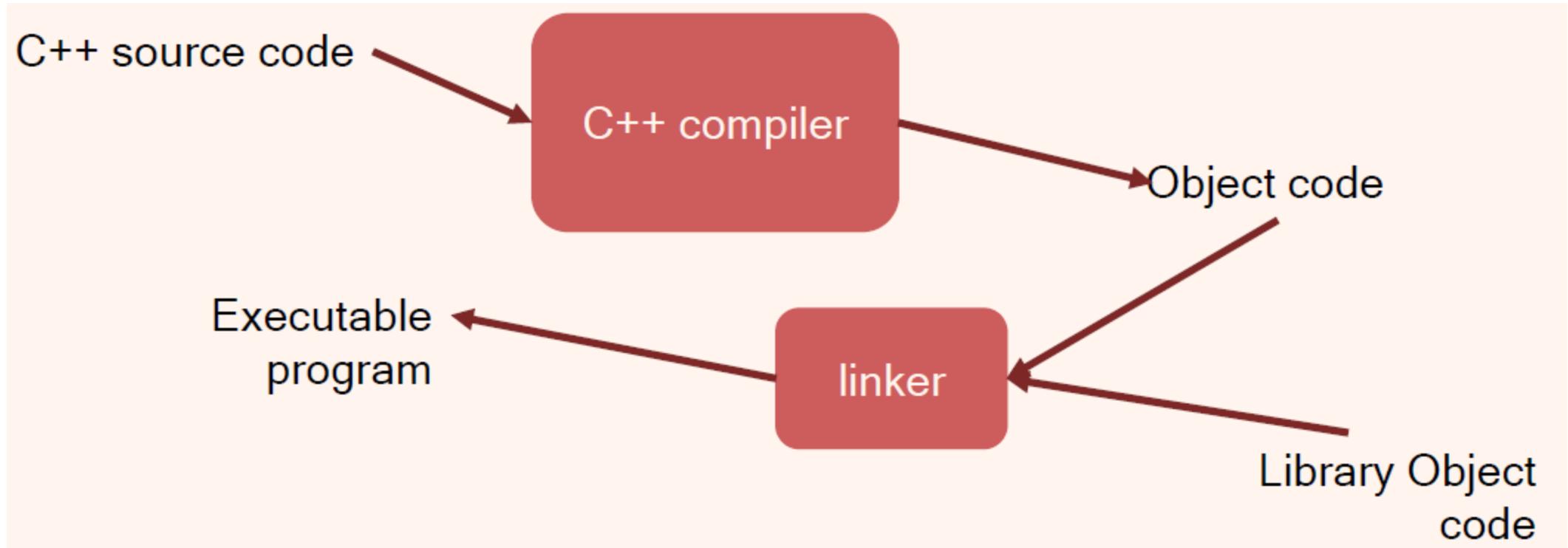
- این دو برنامه، در اجرا هیچ فرقی با هم ندارند:

```
#include <stdio.h>
int main()
{
    printf("Hello World!");
    return 0;
}
```

```
#include <stdio.h> int main() { printf("Hello World!"); return 0; }
```

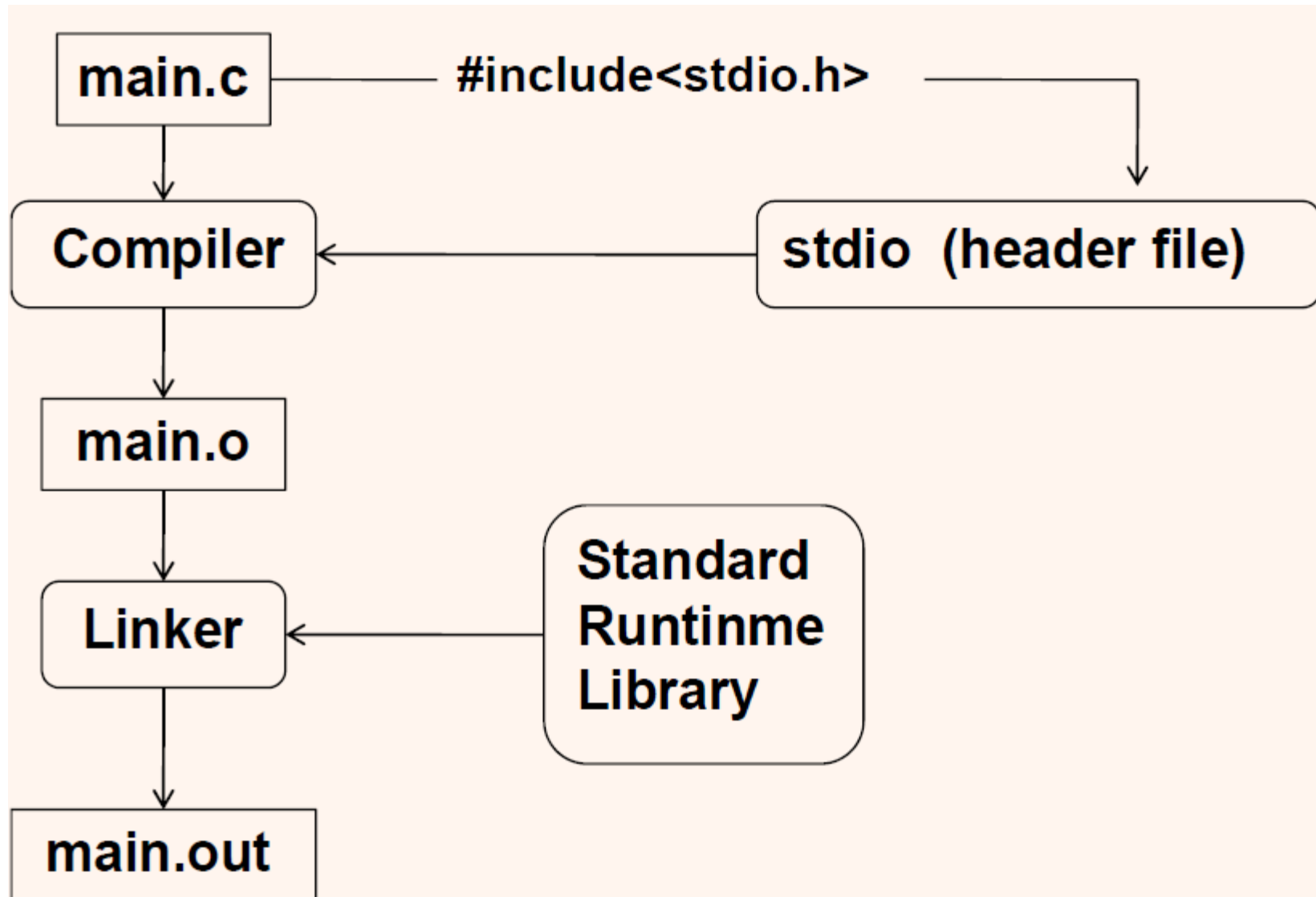
- دستورات با سمیکالون (;) از هم جدا می‌شوند (نه با فاصله یا خط جدید)
- البته برنامه اول برای ما راحت‌تر خوانده می‌شود و بهتر است

# مراحل کامپایل و تولید برنامه اجرایی





# مثال: تولید برنامه اجرایی از برنامه‌ای که نوشتیم



```
#include <stdio.h>
int main()
{
    printf("Hello World!");
    return 0;
}
```

- فرض کنید برنامه را در hello.c نوشته‌ایم
- اجرای دستور کامپایل در سیستم عامل لینوکس:

- gcc hello.c

- نتیجه: فایل اجرایی a.out ساخته می‌شود
- اگر در ویندوز کامپایل کنید، a.exe ساخته می‌شود
- نتیجه اجرای فایل a.out:

- Hello World!

- نوشتن یک برنامه ساده
- مشاهده مراحل کامپایل
- اجرای برنامه

- تاریخچه و اهمیت زبان‌های C و C++
- درباره زبان C++ و ارتباط آن با C
- نوشتن اولین برنامه به زبان C
- معنای دستورالعمل‌های اولیه زبان C
- فرایند کامپایل و اجرای یک برنامه به زبان C

پایان