شروع برنامه‌نویسی ۲

مبانی برنامه‌نویسی

(۱۳۹۹-۱۳-۱۱)

جلسه‌ی هفتم

دانشگاه شهید بهشتی

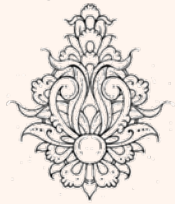
پاییز ۱۳۹۹

دانشکده‌ی مهندسی برق و کامپیوتر

احمد محمودی ازناوه

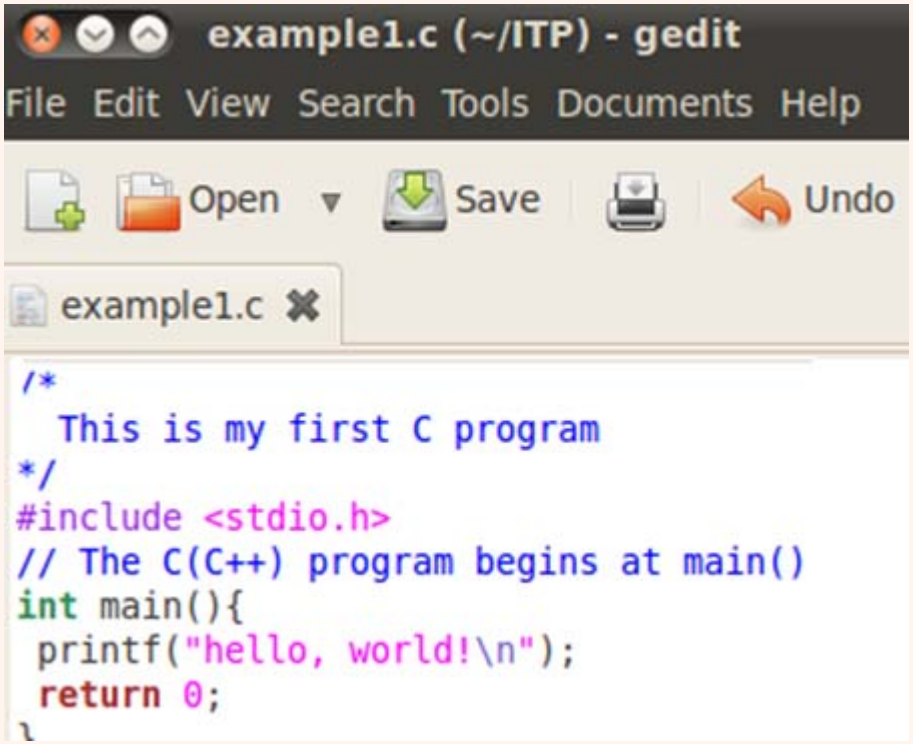
فهرست مطالب

- نخستین برنامه
- مراحل کامپایل
- انواع داده



نوشتن نخستین برنامه به زبان C

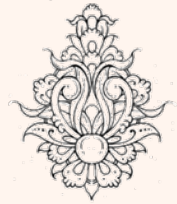
- تنها راه آموختن یک زبان نوشتن برنامه‌ای به آن زبان است، تقریبا اولین برنامه‌ای که به هر زبانی نوشته می‌شود، برنامه‌ی «hello world!» است؛ برنامه‌ای که عبارت گفته شده را چاپ کند.



توضیحات برنامه

- خط نخست: **توضیح** (comment)، متنی است که به منظور راهنمایی و درک بهتر به برنامه اضافه می‌شود و تاثیری در اجرای برنامه ندارد. کامپایلر توضیحات برنامه را قبل از اجرا حذف می‌کند. این به معنای بی‌اهمیت بودن این توضیحات نیست. یک برنامه‌ی خوب، برنامه‌ای است که علاوه بر کارایی قابلیت فهم بالایی هم داشته باشد.

```
/*
  This is my first C program
*/
#include <stdio.h>
// The C(C++) program begins at main()
int main(){
 printf("hello, world!\n");
 return 0;
}
```



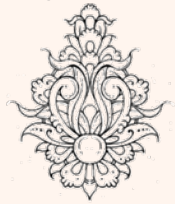
توضیحات برنامه (ادامه ...)

```c
/*
  This is my first C program
*/
#include <stdio.h>
// The C(C++) program begins at main()
int main(){
 printf("hello, world!\n");
 return 0;
}
```

با استفاده از دو علامت اسلش «//» : هر متنی که بعد از دو علامت اسلش بیاید تا پایان همان سطر یک توضیح تلقی می‌شود. این شیوه در استاندارد اولیه زبان C وجود نداشت، در زبان C++ و نهایتاً به زبان C هم اضافه شد.

هر متنی که با علامت «/*» شروع شود و با علامت «*/» پایان یابد یک توضیح تلقی می‌شود.

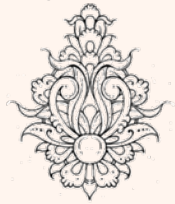
```c
//line comment

/*
block comment
*/
```



راهنمای پیش‌پردازش

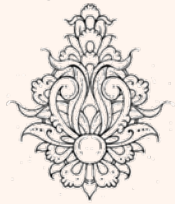
- دومین خط از کد بالا یک «راهنمای پیش‌پردازشگر» یا همان preprocessor است. در واقع پیش‌پردازش گامی است که پیش از کامپایل برنامه انجام می‌شود.
- راهنمای پیش‌پردازگر با کاراکتر «#» شروع می‌شود، که نشان می‌دهد این خط، یک راهنمای پیش‌پردازگر است. این کاراکتر باید در ابتدای همه‌ی خطوط راهنمای پیش‌پردازگر باشد.
- عبارت **include**
  - در واقع به کامپایلر اعلام می‌کند که در این برنامه از توابع کتابخانه‌ای استفاده شده است.
  - نام یک «فایل کتابخانه‌ای» میان دو علامت < > قرار می‌گیرد. به چنین فایلی فایل سرآیند می‌گویند.

```c
/*
  This is my first C program
*/
#include <stdio.h>
// The C(C++) program begins at main()
int main(){
 printf("hello, world!\n");
 return 0;
}
```



شروع برنامه

- در واقع برنامه‌ی ما با عبارت «main» آغاز می‌شود.
  - عبارت int که یک نوع عددی در C++ است. در واقع نوع خروجی تابع main را مشخص می‌کند.
  - اگر برنامه با موفقیت اجرا شود مقدار عددی صفر باز می‌گردد.
  - عبارت main که به آن «تابع اصلی» در C++ می‌گویند.
  - دو پرانتز ( ) که نشان می‌دهد عبارت main یک «تابع» است.
  - بدنه‌ی برنامه بین دو علامت { } قرار می‌گیرد.
  - تابع printf عبارت رشته‌ای محصور در " " را چاپ می‌کند، این تابع متعلق به کتابخانه‌ی **stdio** می‌باشد.

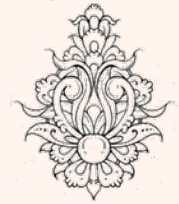


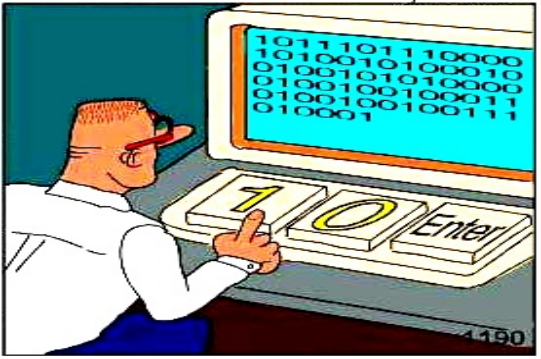
دو شیوه‌ی مختلف

- دو تصویر زیر از لحاظ کامپایلر هیچ تفاوتی ندارند. به نظر شما کدام شیوه مناسب‌تر است؟
- به طور کلی شیوه‌ی نوشتن همراه با تورفتگی توصیه می‌شود، در آینده در این مورد بیشتر صحبت می‌کنیم.

```c
// The C(C++) program begins at main()
int main(){
 printf("hello, world!\n");
 return 0;
}
```

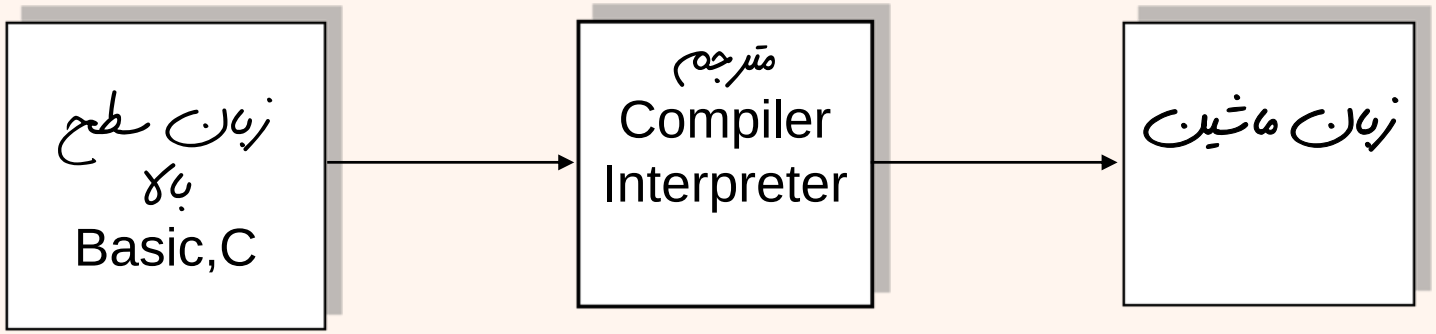
```c
int main(){  printf("hello, world!\n");  return 0; }
```





ترجمه‌ی برنامه به زبان ماشین

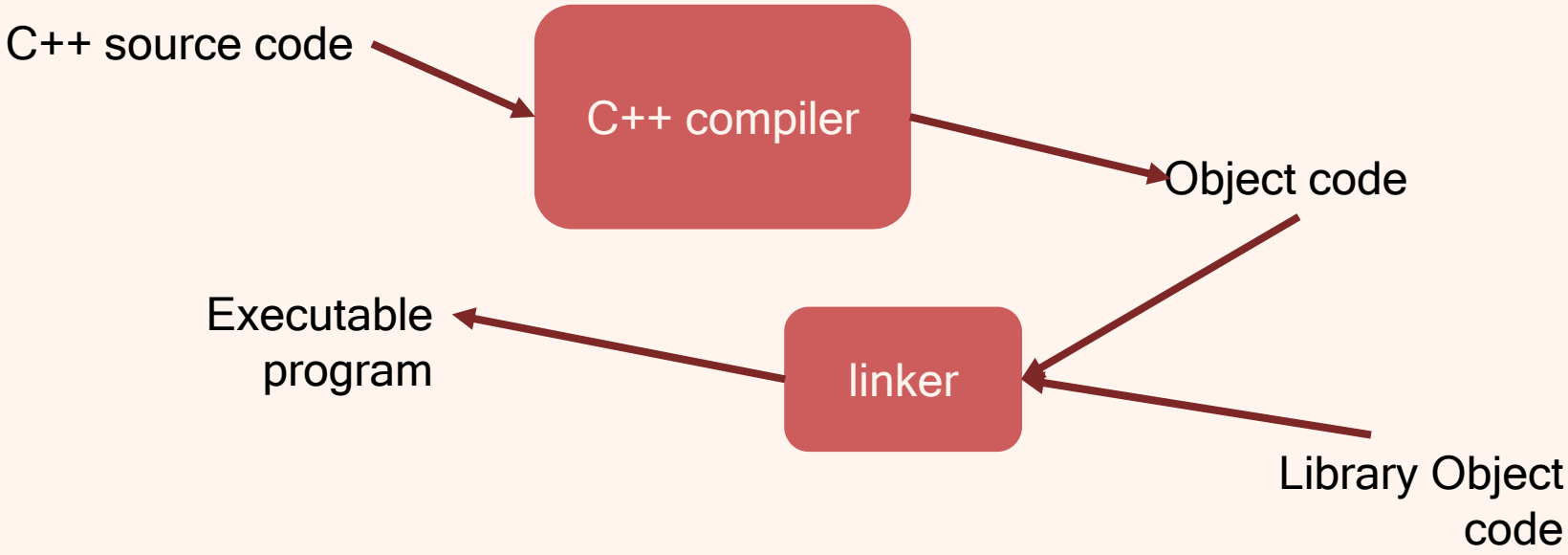
- **انواع مترجم**
  - COMPILER -
  - INTERPRETER -




```c
/*
  This is my first C program
*/
#include <stdio.h>
// The C(C++) program begins at main()
int main(){
 printf("hello, world!\n");
 return 0;
}
```

```
00000000 <main>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   83 e4 f0                and    $0xfffffff0,%esp
   6:   83 ec 10                sub    $0x10,%esp
   9:   c7 04 24 00 00 00 00    movl   $0x0,(%esp)
  10:   e8 fc ff ff ff          call   11 <main+0x11>
  15:   b8 00 00 00 00          mov    $0x0,%eax
  1a:   c9                      leave
  1b:   c3                      ret
ahmad@ubuntu:~/ITP$
```

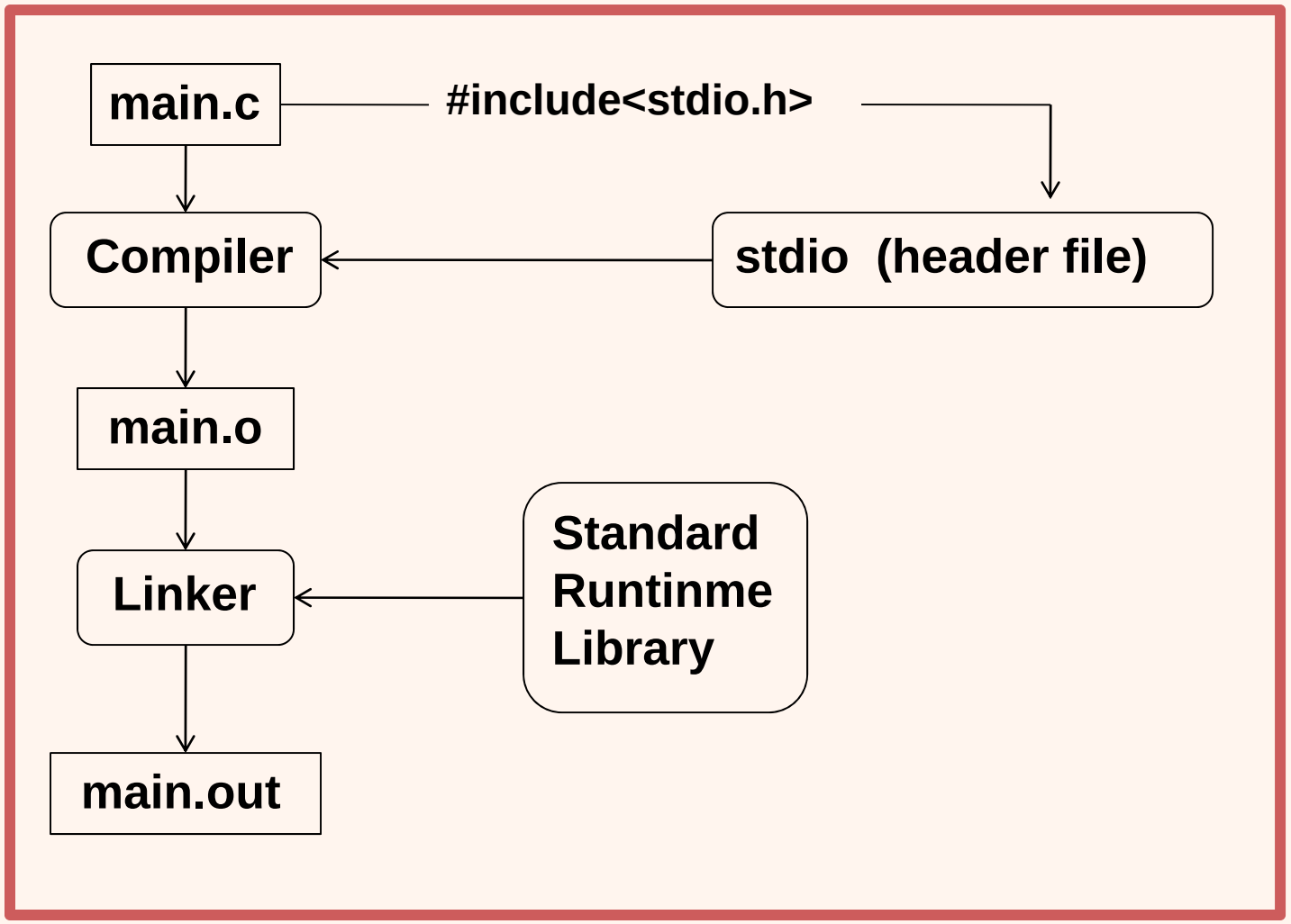
کامپایلر و پیونده



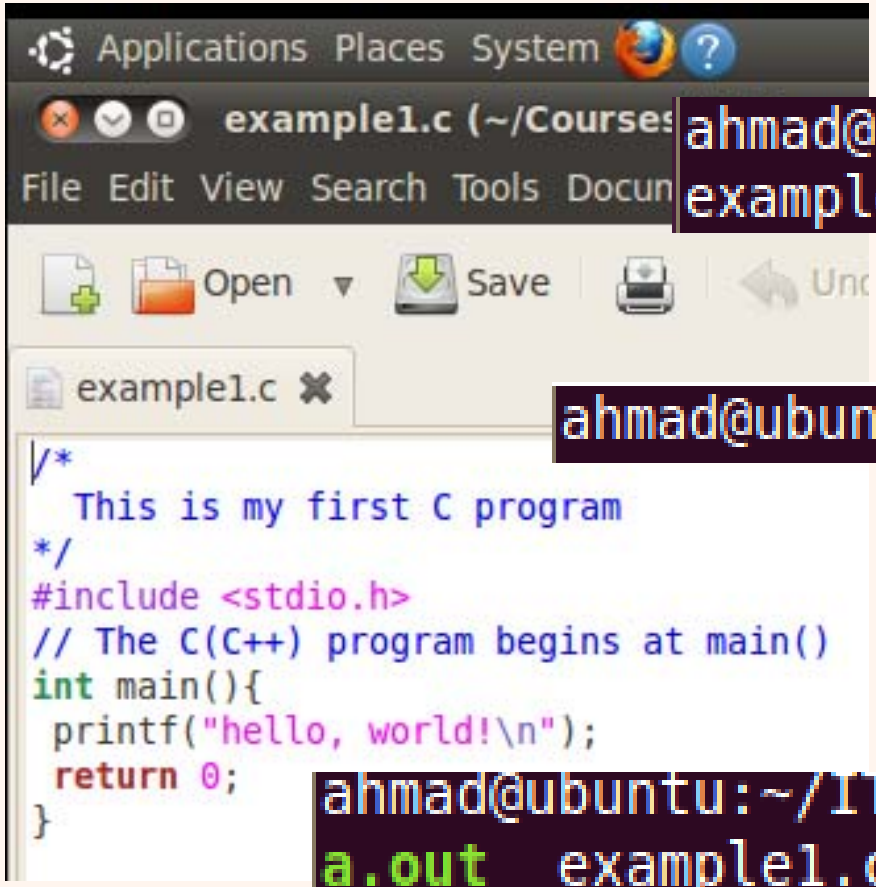
- نوشتن کد C++
- کامپایلر آن چه را که به زبان قابل درک برای انسان نوشته شده است را به Object Code ترجمه می کند.
- Linker کد شما را به کدهایی که برای اجرا لازم است متصل کرده و برنامه را به صورتی که قابلیت اجرا داشته باشد در می‌آورد.
- نتیجه‌ی نهایی یک برنامه‌ی قابل اجراست.

برگرفته از کتاب Programming Principe and Practice using c++

فایل سرآیند


main.c
#include<stdio.h>
Compiler
stdio (header file)
main.o
Linker
Standard
Runtinme
Library
main.out

آشنایی مقدماتی با نحوه‌ی کامپایل




```
ahmad@ubuntu:~/ITP$ ls
example1.c  example2.c
```

```
ahmad@ubuntu:~/ITP$ gcc example1.c
```

```
ahmad@ubuntu:~/ITP$ ls
a.out  example1.c  example2.c
```

```
ahmad@ubuntu:~/ITP$ ./a.out
hello, world!
```

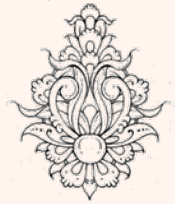
Identifier

- نام توابع، پارامترها و متغیرها را شامل می‌شود.
- C++ نسبت به حروف «حساس به حالت» است یعنی A و a را یکی نمی‌داند.
- در نام‌گذاری‌ها از حروف الفبا، ارقام و تیره‌ی زیرین (underscore) استفاده می‌شود.
- C++ برای طول شناسه‌ها محدودیت ندارد ولی چون کامپایلری خاص ممکن است محدود باشد، حداکثر طول شناسه را ۳۱ در نظر بگیرید.
- یک شناسه نباید با یک رقم شروع شود.

جدول کلمات کلیدی C++

- در نام‌گذاری شناسه‌ها از کلمات کلیدی نباید استفاده شود.

asm	default	for	private	struct	unsigned
auto	delete	friend	protected	switch	using
bool	do	goto	public	template	virtual
break	double	if	register	this	void
case	dynamic_cast	inline	reinterpret_cast	throw	volatile
catch	else	int	return	true	wchar_t
char	enum	long	short	try	while
class	explicit	mutable	signed	typedef	
const	extern	namespace	sizeof	typeid	
const_cast	false	new	static	typename	
continue	float	operator	static_cast	union	

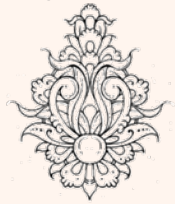


نکات

- انتخاب نام‌های معنی‌دار برای متغیرها می‌تواند در «خودمستند» بودن برنامه کمک کند.

Self-documenting

- برای نام‌گذاری از کلماتی که با دو خط زیرین شروع می‌شوند پرهیز کنید (این دست نام‌گذاری در کتابخانه‌های داخلی زبان استفاده می‌شود)



متغیرها

- به محل‌هایی از حافظه که دارای یک آدرس‌هستند و دارای اندازه‌ای مشخص می‌باشند Object گفته می‌شود.
- آدرس یک Object آدرس بایت نخست آن است.
- نام Objectها متغیر نامیده می‌شود که به واسطه‌ی آن می‌توان مقداری را ذخیره و در برنامه استفاده نمود.
- متغیرها، به منزله‌ی مکان‌هایی در حافظه است که دارای نام، اندازه و مقدار خواهد بود.