

انواع داده

مبانی برنامه‌نویسی

(۱۳۹-۱۳۳-۱۱)

جلسه‌ی نهم



دانشگاه شهید بهشتی

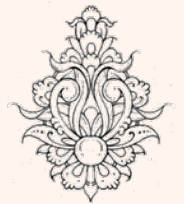
پاییز ۱۳۹۲

دانشکده‌ی مهندسی برق و کامپیوتر

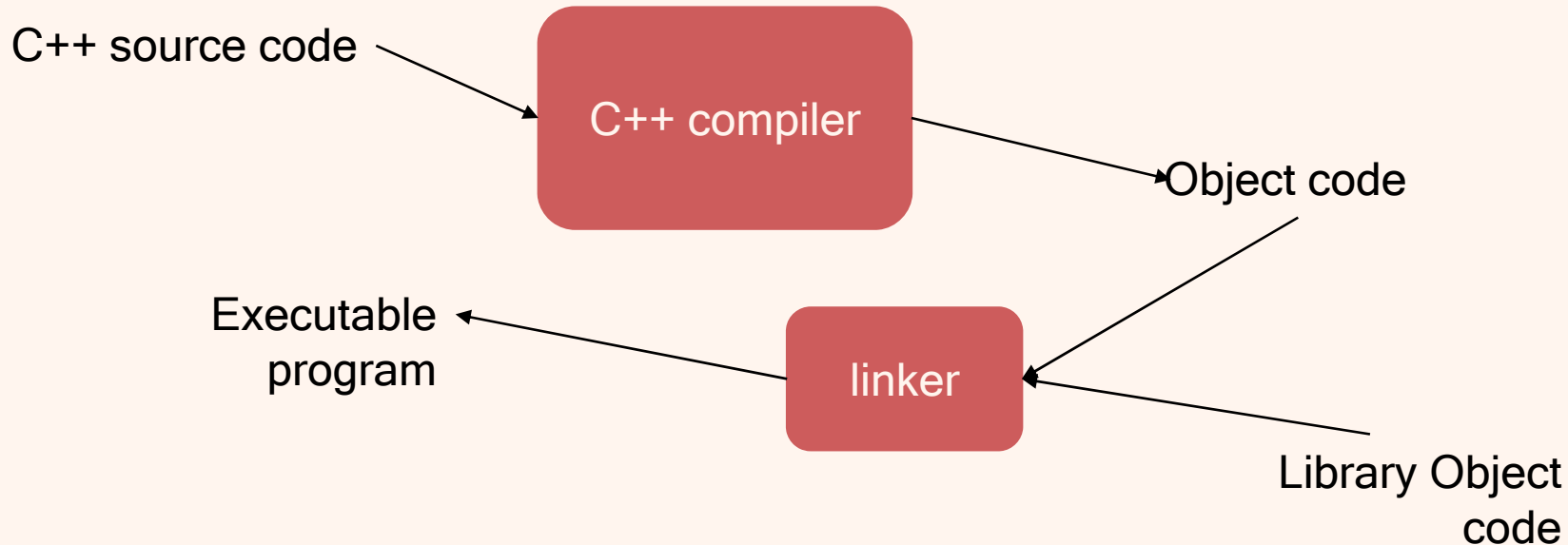
احمد محمودی ازناوه

# فهرست مطالب

- مروری بر جلسه‌ی پیش
- انواع داده
- خروجی و ورودی



# کامپایلر و پیونده

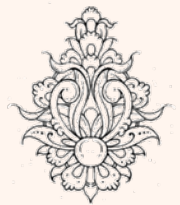


• نوشتن کد C++

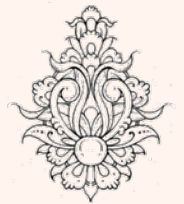
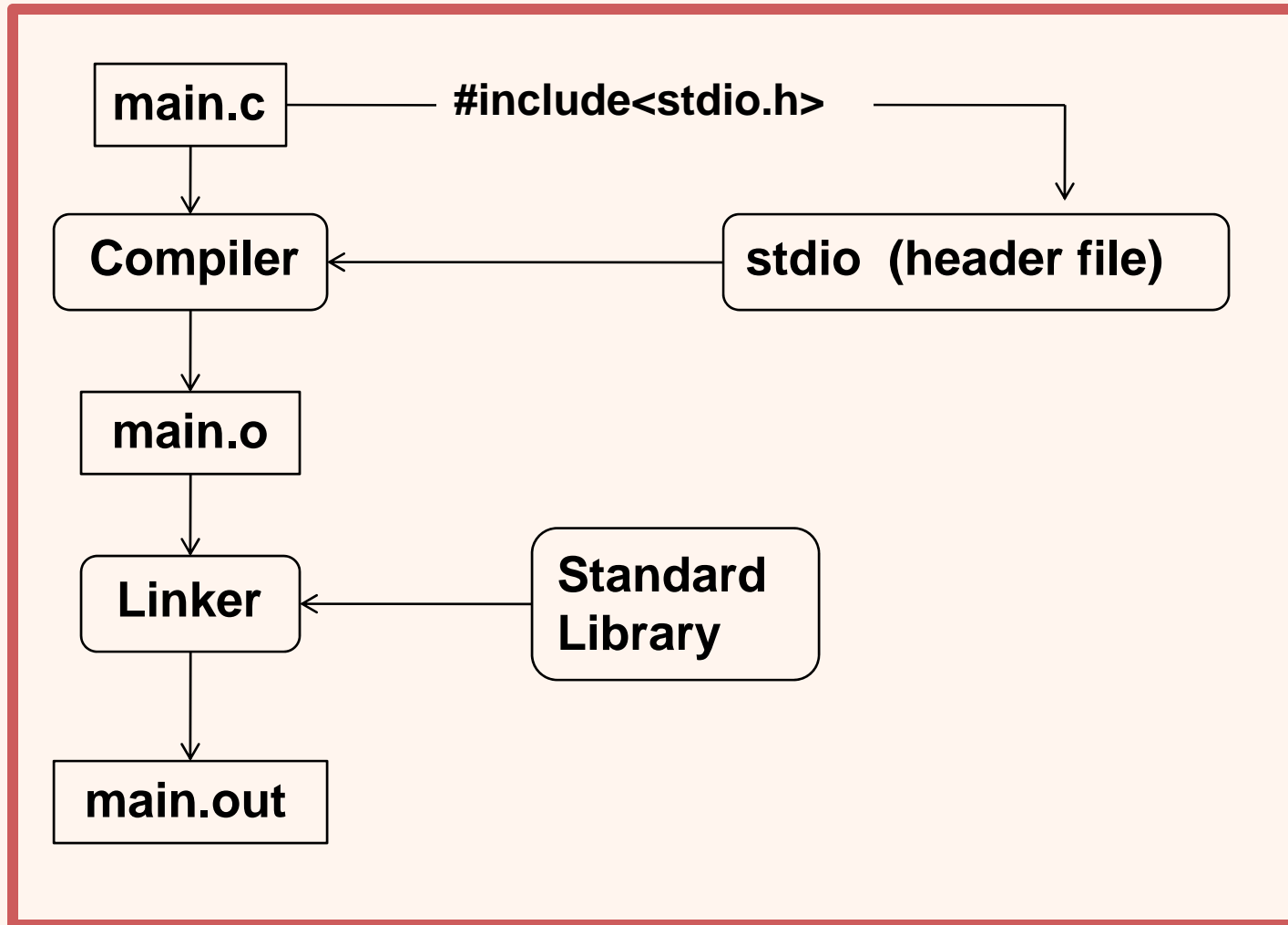
• کامپایلر آن چه را که به زبان قابل درک برای انسان نوشته شده است را به Object Code ترجمه می کند.

• Linker کد شما را به کدهایی که برای اجرا لازم است متصل کرده و برنامه را به صورتی که قابلیت اجرا داشته باشد در می آورد.

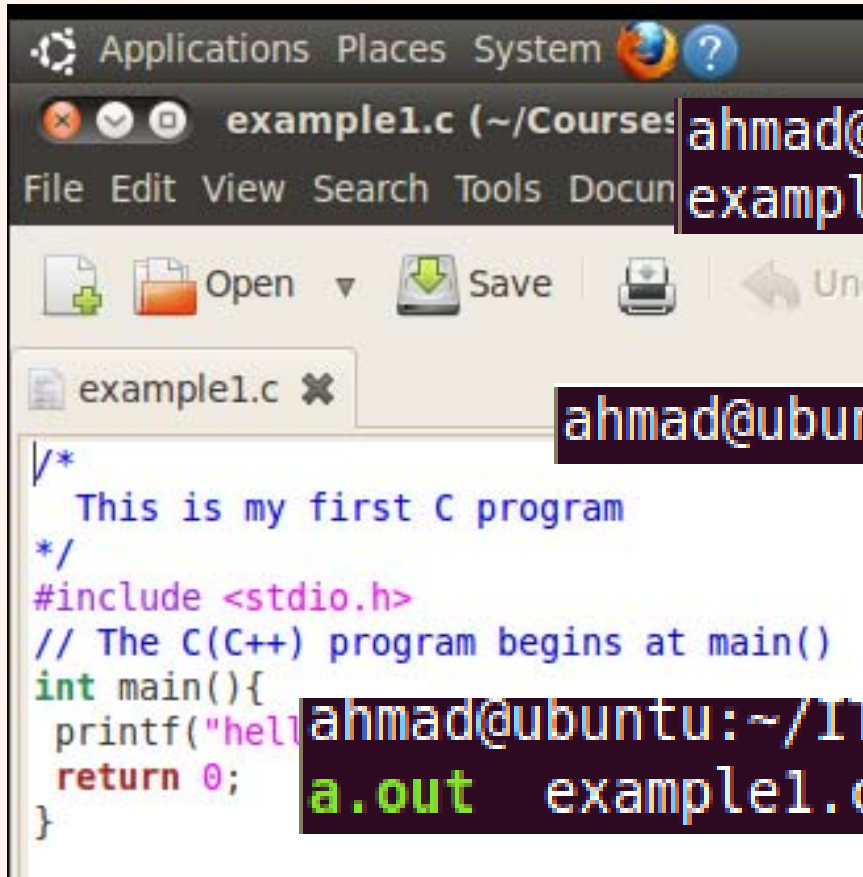
• نتیجه نهایی یک برنامه‌ی قابل اجراست.



# فایل سرآیند



# آشنایی مقدماتی با نحوه‌ی کامپایل



The screenshot shows a terminal window with the following content:

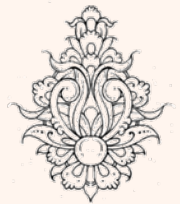
```
Applications Places System
example1.c (~/.Courses)
File Edit View Search Tools Document
example1.c
/*
  This is my first C program
*/
#include <stdio.h>
// The C(C++) program begins at main()
int main(){
  printf("hello, world\n");
  return 0;
}
```

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

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

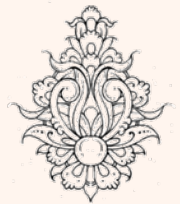
```
ahmad@ubuntu:~/ITP$ ls
a.out example1.c example2.c
```

```
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.



- در نامگذاری شناسه‌ها از کلمات کلیدی نباید استفاده شود.

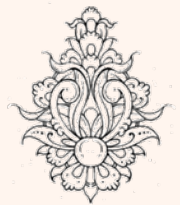
asm	default	for	private	struct	unsigned
auto	delete	friend	protected	switch	using
bool	do	goto	public	template	virtual
break	double	if	register	this	void
case	dynamic_cast	inline	reinterpret_cast	throw	volatile
catch	else	int	return	true	wchar_t
char	enum	long	short	try	while
class	explicit	mutable	signed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



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

## *Self-documenting*

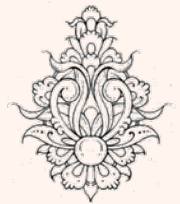
- برای نام‌گذاری از کلماتی که با دو خط زیرین شروع می‌شوند پرهیز کنید (این دست نامگذاری در کتابخانه‌های داخلی زبان استفاده می‌شود)





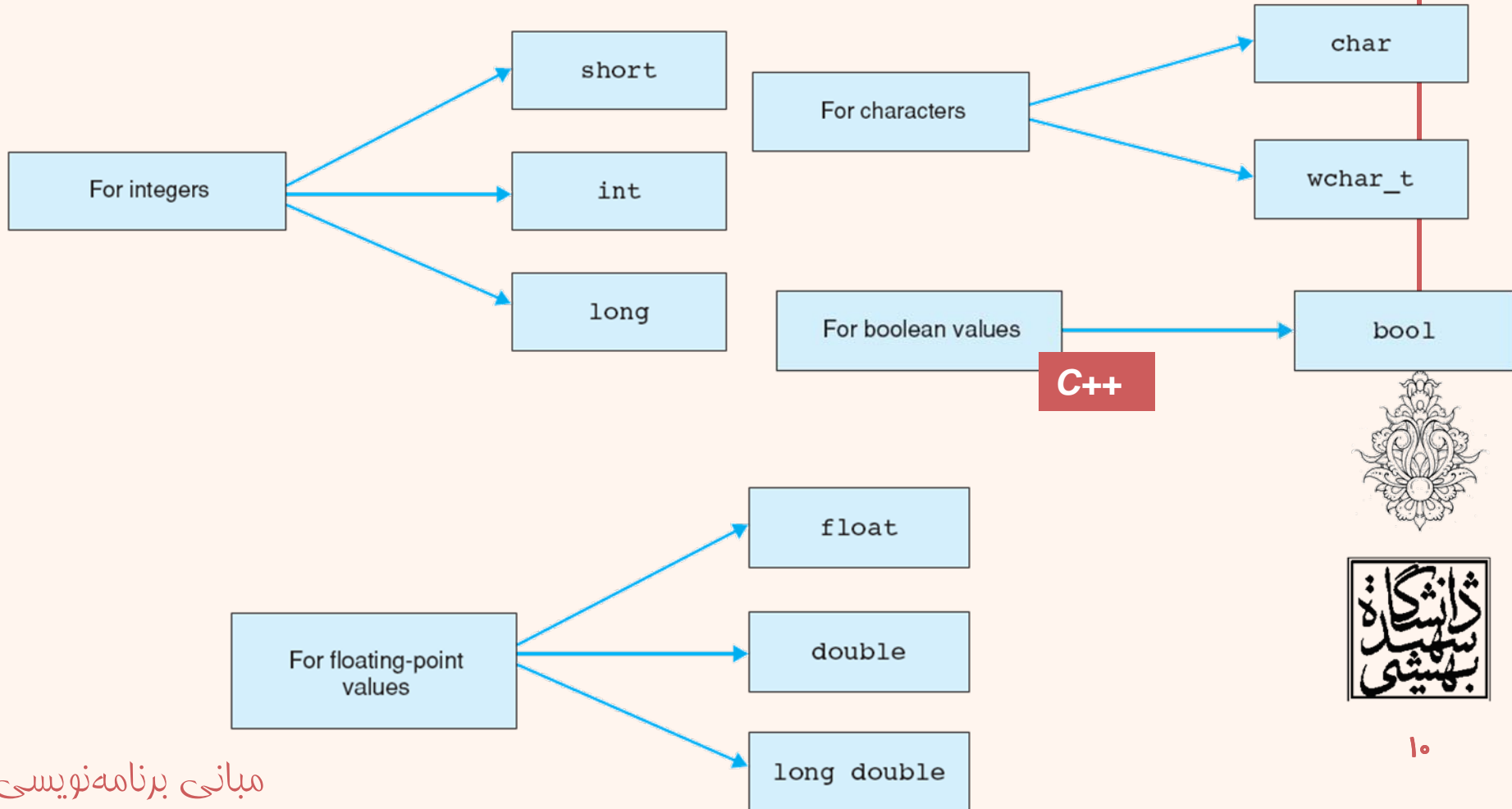
*we can do nothing or interest with a computer without storing data in memory*

- به محل‌هایی از حافظه که دارای یک آدرس هستند 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.



# انواع داده

- انواع داده‌ها بسته به مورد استفاده در برنامه می‌توانند شامل موارد زیر باشند:



# فلاصدهای از انواع اصلی داده

Type	Size	Range of Values (decimal)
char	1 byte	-128 to +127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to +127
int	2 byte resp. 4 byte	-32768 to +32767 resp. -2147483648 to +2147483647
unsigned int	2 byte resp. 4 byte	0 to 65535 resp. 0 to 4294967295
short	2 byte	-32768 to +32767
unsigned short	2 byte	0 to 65535
long	4 byte	-2147483648 to +2147483647
unsigned long	4 byte	0 to 4294967295

Type	Size	Range of Values	Lowest Positive Value	Accuracy (decimal)
float	4 bytes	-3.4E+38	1.2E-38	6 digits
double	8 bytes	-1.7E+308	2.3E-308	15 digits
long double	10 bytes	-1.1E+4932	3.4E-4932	19 digits



# اعلام و مقداردهی متغیرها

## Declaration of variables

- برای استفاده از متغیرها می‌باید ابتدا آن‌ها را اعلام نمایید:

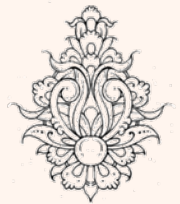
```
int a;  
float mynumber;  
int b, c;
```

- می‌توان در هنگام تعریف متغیر فرآیند مقداردهی اولیه را نیز صورت داد.

## Initialization

```
int a=0;  
float mynumber=3.67;  
int b=2, c=7;
```

```
float x(1.875, C++)
```

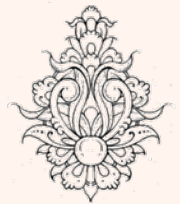


# انتساب

- فرآیند انتساب به یک متغیر را گویند.
- مقدار سمت راست ارزیابی و به سمت چپ نسبت داده می‌شود.

$x=a+b;$   
 $x=12 ;$   
 $x=y=z=28 ;$

- تفاوت مقداردهی و انتساب چیست؟



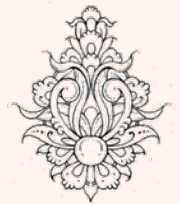
## توابع ورودی و خروجی

- برای این که بتوانیم هنگام اجرای برنامه مقادیری را چاپ کنیم، از تابع **printf** استفاده می‌کنیم.

```
printf ( "text & Argument type", Argument name ) ;
```

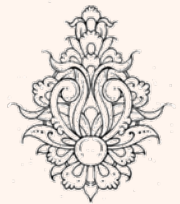
- برای این که بتوانیم هنگام اجرای برنامه مقادیری را وارد کنیم از تابع **scanf** استفاده می‌کنیم.

```
scanf ( "Argument type", &Argument name ) ;
```



# printf

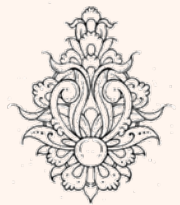
- به منظور مشاهده‌ی داده‌های به دست آمده، از این تابع استفاده می‌شود.
- این تابع می‌تواند مقادیر متغیرها را در خروجی چاپ کند.
- تعداد ورودی‌های این تابع متغیر است.
- ورودی (آرگومان) اول **رشته‌ی کنترلی** است که چگونگی چاپ متغیرها را مشخص می‌کند.
- کاراکترهای رشته‌ی کنترلی عیناً چاپ می‌شود، مگر در مواردی که با «%» همراه شده‌اند.



# printf (ادامه...)

- در واقع متناظر با هر متغیر یک علامت «%» وجود دارد که توسط کاراکتر بعد از % شیوهی نمایش را مشخص می‌کند:

کاراکتر کنترلی	مفهوم
%d	نمایش یک عدد صحیح در مبنای ده
%u	نمایش یک عدد صحیح بدون علامت در مبنای ده
%f	نمایش یک عدد اعشاری
%c	نمایش به صورت کاراکتری
%%	نمایش علامت %





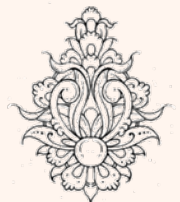
مثال

```
a=67
b=-1
c=E
x=1.200000
y=3.141500
```

```
a=C
b=4294967295
c=69
x=1073741824
y=3.141500
```

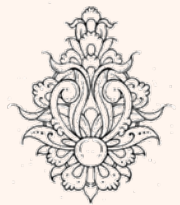
```
#include <stdio.h>
int main(){
    int a=67;
    int b=-1;
    char c=69;
    float x=1.2;
    double y=3.1415;
    printf("a=%d\n",a);
    printf("b=%d\n",b);
    printf("c=%c\n",c);
    printf("x=%f\n",x);
    printf("y=%f\n",y);
    return 0;
}
```

```
#include <stdio.h>
int main(){
    int a=67;
    int b=-1;
    char c=69;
    float x=1.2;
    double y=3.1415;
    printf("a=%c\n",a);
    printf("b=%u\n",b);
    printf("c=%d\n",c);
    printf("x=%d\n",x);
    printf("y=%f\n",y);
    return 0;
}
```



# cin - cout

- در C++، cin و cout برای این کار استفاده می‌شود هر چند می‌توان از توابع کتابخانه‌های زبان C همانند printf و scanf نیز استفاده نمود.
- چرا استفاده از این دو تابع با استفاده از cin و cout جایگزین شده است؟
- استفاده از کتابخانه‌ی iostream به دلیل type safe بودن آن راحت‌تر است.



limit.c ✕

```

#include<stdio.h>
#include<limits.h> // Definition of INT_MIN, ...
int main()
{
    printf("Range of types int & unsigned int\n\n");
    printf("Type Minimum & Maximum\n");
    printf("-----\n");
    printf("int %d %d \n",INT_MIN,INT_MAX);
    printf("unsignedint %d %u \n",0,UINT_MAX);
    return 0;
}

```

```

Range of types int & unsigned int

```

```

Type Minimum & Maximum

```

```

-----

```

```

int -2147483648 2147483647

```

```

unsignedint 0 4294967295

```

