

مبانی کامپیوتر و برنامه سازی

مدرس: سعدون عزیزی

دانشکده مهندسی – گروه کامپیوتر

پاییز ۹۶

سرفصل مطالب

- ☐ آشنایی با کامپیوتر و الگوریتم
- ☐ مقدمه‌ای بر برنامه‌نویسی C
- ☐ محاسبات
- ☐ ورودی/خروجی
- ☐ حلقه‌ها
- ☐ دستورات شرطی
- ☐ توابع
- ☐ آرایه‌ها
- ☐ کاراکترها و رشته‌ها
- ☐ اشاره‌گرها
- ☐ ساختار
- ☐ فایل‌ها

تعریف متغیر

□ هر متغیر قبل از استفاده باید معرفی شود و علاوه بر نام، نوع نیز باید تعیین گردد

```
type var1, var2;
```

مثال:

```
int i, n;
```

```
float grade, sum, x;
```

```
double error;
```

```
char c1, c2, str[20];
```

متغیر

اندازه حافظه	نوع متغیر	کلمه کلیدی
۱ بایت	کاراکتر	char
۲ بایت (۴ بایت)	صحیح	int
۲ بایت (۴ بایت)	صحیح بدون علامت	unsigned int
۴ بایت	صحیح بلند	long int
۴ بایت	اعشاری	float
۸ بایت	اعشاری مضاعف	double
۱۰ بایت	اعشاری مضاعف (با بزرگی و دقت بیشتر)	long double

مقداردهی اولیه

```
type var1=value1, var2=value2;
```

مثال:

```
int i=0, n=20;
```

```
float grade, sum=0.0;
```

```
char c1='A', str[20]="Computer Science";
```

* برای متغیر رشته ای str، ۲۰ بایت اختصاص داده می شود و از اولین بایت به ترتیب کاراکترهای Computer Science قرار می گیرند و در انتهای کاراکتر تهی ('\0') اضافه می شود که بیانگر انتهای رشته است. این کار به طور خودکار صورت می گیرد.

شناسه ثابت

□ بعد از تعریف یک شناسه ثابت، نمی توان مقدار آن را تغییر داد.

```
const type var1=value1;
```

مثال:

```
const int max=100;
```

```
const float pi=3.1415;
```

عملگرهای محاسباتی

□ پنج عملگر محاسباتی در C

+ - * / %

□ اگر دو طرف عملگرهای محاسباتی صحیح باشد، حاصل محاسبه صحیح خواهد بود و در غیر این صورت حاصل اعشاری می شود ($7*4=28$ و $7.0*4=28.0$)

□ چنانچه دو طرف عملگر تقسیم مقادیر صحیح باشند، حاصل خارج قسمت تقسیم به صورت صحیح خواهد بود و ارقام بعد از ممیز حذف می شوند ($8/3=2$)

□ دو طرف عملگر % باید صحیح باشند (خطا $8.5\%3$)

□ حاصل عبارت $x\%y$ با مقدار x هم علامت است ($-10\%4=-2$ و $10\%-4=2$)

عبارت جایگزینی و عملگرهای جایگزینی

❑ `variable = expression;`

`m = 4*(x*x) + 2*x + 1;`

❑ `i+=1;` \rightarrow `i=i+1;`

❑ `j-=2;` \rightarrow `j=j-2;`

❑ `x*=a+b;` \rightarrow `x=x*(a+b);`

❑ `y/=2-t;` \rightarrow `y=y/(2-t);`

❑ `m%=n;` \rightarrow `m=m%n;`

عملگرهای یک طرفه

□ عملگر افزایشی (++)

```
i=6;  
j=++i;      → i=7 , j=7  
k=i++;      → k=7 , i=8
```

□ عملگر کاهشی (--)

```
i=6;  
j=--i;      → i=5 , j=5  
k=i--;      → k=5 , i=4
```

عملگرهای یک طرفه

□ عملگر تبدیل نوع (type): نوع یک متغیر یا یک مقدار را تغییر می دهد

```
int i;
```

```
float x=15.5;
```

```
i=(int)x%8;
```

□ عملگر اندازه (sizeof): مقدار فضای اشغال شده توسط یک عبارت یا یک متغیر را برمی گرداند.

```
int i=12345, j;
```

```
double pi=3.14;
```

```
j=sizeof i;    → j=4
```

```
j=sizeof pi;   → j=8
```

```
j=sizeof 20;   → j=4
```

توابع کتابخانه ای

□ در C بسیاری از محاسبات ریاضی، مانند محاسبه سینوس، کسینوس، قدر مطلق، لگاریتم، جزء صحیح، توان، جذر و ... از قبل در توابع مستقل نوشته شده است و در اختیار برنامه نویس قرار دارد. قبل از استفاده از این توابع، باید کتابخانه مربوطه را فراخوانی کرد

تابع	خروجی	کتابخانه مربوطه	هدف تابع
<code>abs(int)</code>	<code>int</code>	<code>stdlib.h</code>	قدر مطلق
<code>sin(double)</code>	<code>double</code>	<code>math.h</code>	سینوس
<code>pow(double, double)</code>	<code>double</code>	<code>math.h</code>	آرگومان اول به توان آرگومان دوم
<code>floor(double)</code>	<code>int</code>	<code>math.h</code>	جزء صحیح
<code>fmod(double, double)</code>	<code>double</code>	<code>math.h</code>	باقیمانده تقسیم آرگومان اول بر دوم
<code>rand()</code>	<code>int</code>	<code>stdlib.h</code>	تولید عدد تصادفی
<code>sqrt(double)</code>	<code>double</code>	<code>math.h</code>	جذر

تمرین

برنامه ای بنویسید که یک عدد تصادفی در هر یک بازه های زیر تولید کند:

الف) $[0, 20]$

ب) $[-10, 10]$

ج) $[200, 250]$