متغیرها و ثابت ها

در این بخش متغیرها و انواع مختلف متغیرهای عددی صحیح و ممیزشناور که توسط برنامه های C برای ذخیره داده ها در طی اجرا بکاربرده می شوند توضیح داده خواهد شد. همچنین نحوه تعریف ثابت های برنامه و دو نوع ثابت حقیقی و سمبلیک شرح داده خواهد شد.

متغیرها
انواع داده عددی
اعلان متغیر
مقداردهی اولیه متغیر
ثابت ها

برنامه های کامپیوتری که با انواع مختلف مقادیر داده کار می کنند به راهی برای ذخیره آنها در حافظه نیاز دارند. این مقادیر می توانند اعداد یا کاراکترها باشند. C دو روش ذخیره سازی برای مقادیر دارد؛ متغیرها (variables) و ثابت ها (constants). متغیر محل ذخیره داده است که حاوی مقداری است که در طی اجرای برنامه می توانند تغییر کند. در مقابل، ثابت مقدار ثابتی دارد که نمی تواند تغییر کند.

متغیرها

یک متغیر یک محل ذخیره داده نامگذاری شده در حافظه کامپیوتر است. شما در برنامه با استفاده از نام متغیر به داده ذخیره شده در این محل رجوع می کنید.

نام متغیر

هر متغیر دارای یک نام است. در زبان C اسامی متغیر ها باید از قوانین زیر پیروی کند:

- اسم می تواند شامل حروف، ارقام و کاراکتر زیرخط (_) باشد.
- اولین کاراکتر اسم متغیر باید یک حرف باشد. زیر خط هم می تواند در ابتدای اسم متغیر بکار برود ولی توصیه نمی شود.
- بزرگ و کوچک بودن حروف مهم است، بنابراین اسامی count و Count به دو متغیر جداگانه اشاره دارند. برنامه نویسان اغلب از حروف کوچک برای متغیر ها استفاده می کنند و اسامی با حروف بزرگ معمولا برا ثابت ها بکار می رود.
- کلمات کلیدی نمی توانند به عنوان نام متغیر استفاده شود.

مثال. چند نمونه از اسامی مجاز و غیر مجاز در جدول زیر دیده می شود:

نام متغیر	*مجاز بودن*
Percent	مجاز
y2x5__fg7h	مجاز
annual_profit	مجاز
_1990_tax	مجاز، ولی توصیه نمی شود
savings#account	غیر مجاز، کاراکتر#
double	غیرمجاز، کلمه کلیدی
9winter	غیرمجاز، اولین کاراکتر رقمی

در بعضی کامپایلرها نام متغیر می تواند ۳۱ کاراکتر (در C++ استاندارد ۱۰۲۴ کاراکتر) طول داشته باشد؛ یعنی کامپایلر تنها ۳۱ کاراکتر اول نام را درنظر می گیرد. با این طول می توان نامی برای متغیر انتخاب کرد که معنی داده ذخیره شده در آن را منعکس کند.

نام متغیر کمک به روشن شدن کاربرد آن برای کسی که source برنامه را نگاه می کند.

مثال. در برنامه ای که پرداخت های وام را محاسبه می کند مقدار بهره را می توان در متغیر interest_rate ذخیره کرد.

مرسوم است از زیرخط برای جدا کردن متغیرهای چند کلمه ای استفاده شود. یا ازروش نام گذاری کوهانی (camel notation) که حرف اول هر کلمه بزرگ است پیروی می شود.

مثال. متغیری که بهره بانکی را ذخیره می کنند می تواند به دو صورت interest_rate یا InterestRate تعریف شود.

نکته. نامهای توصیفی برای متغیرها انتخاب کنید.
نکته. به سبکی که برای نامگذاری متغیر ها انتخاب کرده اید در کل برنامه وفادار بمانید.
نکته. شروع نام متغیر با زیرخط غیر ضروری است.
نکته. نامگذاری متغیر همگی با حروف بزرگ غیر ضروری است.

انواع داده عددی

دانستن انواع متغیرهای عددی که زبان برنامه نویسی در اختیار می گذارد لازم است زیرا مقادیر عددی مختلف میزان حافظه متفاوتی را اشغال می کنند و عملیات ریاضی معینی روی آنها انجام می شود. نوع متغیر بسته به طبیعت داده ای که ذخیره می کند می تواند یکی از این انواع تعریف شده باشد. با انتخاب نوع مناسب برای متغیر برنامه شما با بیشترین بازدهی ممکن اجرا می شود.

اعداد صحیح کوچک حافظه کمتری می خواهند و کامپیوتر می تواند عملیات ریاضی مانند جمع و ضرب را به سرعت روی آنها انجام بدهد. درحالیکه مقادیر صحیح بزرگ و اعداد ممیز شناور به فضا و زمان بیشتری برای محاسبات نیاز دارند.

انواع متغیر عددی در دو دسته کلی قرار می گیرند:

- متغیرهای صحیح، مقادیری را نگه می دارند که بخش کسری ندارند. متغیرهای صحیح خود دو نوع هستند: علامتدار، که می توانند مقادیر مثبت و منفی را ذخیره کنند و بدون علامت، که تنها مقادیر مثبت را نگه می دارند.
- متغیرهای ممیزشناور، مقادیری را که دارای بخش کسری هستند یعنی اعداد حقیقی را نگه می دارند.

جدول انواع داده عددی C++ و میزان حافظه هر کدام در یک ریز کامپیوتر با معماری ۱۶ بیتی

Variable Type	Keyword	Bytes Required	Range
Character	char	1	-128 to 127
Integer	int	2	-32768 to 32767
Short integer	short	2	-32768 to 32767
Long integer	long	4	-2,147,483,648 to 2,147,438,647
Unsigned character	unsigned char	1	0 to 255
Unsigned integer	unsigned int	2	0 to 65535
Unsigned short integer	unsigned short	2	0 to 65535
Unsigned long integer	unsigned long	4	0 to 4,294,967,295
Single-precision floating-point	float	4	1.2E-38 to 3.4E38 (Approximate range; precision = 7 digits)
Double-precision floating-point	double	8	2.2E-308 to 1.8E308 (Approximate range; precision = 19 digits)

دقت (precision) به معنی درستی مقادیر ذخیره شده است. مثلا نتیجه تقسیم 1/3 برای یک متغیربا دقت 7 رقم به صورت 0.3333333 ذخیره می شود.

عبارات signed، unsigned، short و long اصلاح کننده هائی هستند که برای تغییر و تبدیل انواع پایه از قبیل char، int و double اضافه می شوند

توجه کنید که نوع های int و short در جدول یکسان هستند. این دو نوع روی معماری های دیگر ممکن است متفاوت باشند برای مثال در معماری 32 بیتی short ۲ بایتی و int ۴ بایتی است.

long float وجود ندارد ولی نوع double هست که دو برابر float است.

انواع داده ها وابسته به platform کامپیوتر است و ممکن است روی کامپیوترهای مختلف متفاوت باشد، اما C با توجه به استاندارد ANSI موارد زیر را تضمین می دهد:

- یک کاراکتر همیشه یک بایت است.
- اندازه short کوچکتر یا مساوی اندازه int است.
- اندازه int کوچکتر یا مساوی اندازه long است.
- اندازه unsigned برابر با int است.
- اندازه float کوچکتر مساوی با اندازه double است.

اعداد صحیح به صورت پیش فرض علامتدار هستند و نیاز به کلمه کلیدی خاصی برای بیان آن نیست ولی درصورت تمایل می توان کلمه کلیدی signed را اضافه کرد.

برنامه نمایش اندازه نوع های مختلف متغیرها

اعلان متغیر

در زبان برنامه نویسی C هر متغیر قبل از استفاده باید اعلان شود. اعلان متغیر کامپایلر را از نام، نوع و مقدار اولیه متغیر آگاه می کند. اگر برنامه سعی کند از متغیری استفاده کند که قبلا اعلان نشده است کامپایلر پیغام خطا صادر می کند.

اعلام متغیر به فرم کلی زیر است:

```
typename varname;
```

typename نوع متغیر را مشخص می کند که باید یکی از نوع های داده ای زبان C باشد. varname نام متغیر است که باید از قواعد گفته شده تبعیت کند.

می توان در یک خط چند متغیر را از یک نوع تعریف کرد.

مثال. متغیرهای count، number و start از نوع int و متغیرهای percent و total از نوع float اعلان شده اند.

```
int count, number, start;     /* three integer variables */
float percent, total;         /* two float variables */
```

در بخش توابع درقسمت حوزه متغیرها درباره محل اعلان متغیر در برنامه توضیح داده شده است. فعلا کلیه متغیرها را در آغاز تابع main() اعلان کنید.

کلمه کلیدی typedef

کلمه کلیدیtypedef برای ایجاد نامی جدید به نوع داده موجود بکار می رود و در واقع یک مترادف برای آن نوع تولید می کند.

مثال. عبارت زیر برای نوع داده ای int مترادف integer را ایجاد می کند. بنابراین در برنامه می توان از کلمه integer برای اعلان متغیرهای از نوع int استفاده کرد.

```
typedef int integer;
integer count;
```

توجه داشته باشید که typedef نوع داده جدیدی را تولید نمی کند بلکه تنها به شما اجازه نامگذاری نوع داده ای که قبلا تعریف شده را می دهد. کاربرد زیاد آن در نوع های داده ای ترکیبی است که در بخش ساختمان مطالعه خواهید کرد.

مقداردهی اولیه متغیرهای عددی

هنگام اعلان یک متغیر، با وجودیکه مقدار آن هنوز تعیین نشده است، به کامپایلر فرمان داده می شود تا فضای لازم برای متغیر را کنار بگذارد. در این حالت مقدار متغیر می تواند صفر یا یک مقدار تصادفی باشد. قبل از استفاده از متغیر همیشه باید آن را با یک مقدار مشخص مقداردهی کرد. این کار موقع اعلان متغیر می تواند صورت بگیرد. برای اینکار بعد از اسم متغیر علامت مساوی و مقدار اولیه موردنظر را ذکر کنید.

مثال.

```
int count = 0;
double percent = 0.01, taxrate = 28.5;
```

علامت مساوی (=) عملگر واگذاری در زبان C است که در بخش عبارات توضیح داده می شود.

توجه کنید مقداراولیه متغیر خارج از محدوده مجاز نباشد. کامپایلر چنین خطائی را گیر نمی اندازد و برنامه ممکن ا ست هنگام اجرا نتایج غیرمنتظره ای تولید کند.

نکته. با تعداد بایت هائی که نوع های متغیر روی کامپیوتر شما اشغال می کند به خوبی آشنا شوید.
نکته. از typedef برای خوانا کردن برنامه خود استفاده کنید.
نکته. تاحد ممکن هنگام تعریف متغیر به آنها مقداراولیه بدهید. استفاده از متغیری که مقداردهی نشده است ممکن است نتایج غیرقابل پیش بینی تولید کند.
نکته. استفاده از متغیرهای float و double برای ذخیره اعداد صحیح باعث پایین آمدن کارائی برنامه می شود.
نکته. مقادیر بزرگ را در نوع متغیری که برای ذخیره آنها کوچک است ذخیره نکنید.
نکته. اعداد منفی را در نوع های بدون علامت ذخیره نکنید.

ثابت ها

ثابت مشابه متغیر محلی برای ذخیره داده ای است که توسط برنامه استفاده می شود. اما برخلاف متغیر، مقدار ذخیره شده در یک ثابت در طی اجرای برنامه قابل تغییر نیست.

C++ دو نوع ثابت واقعی و سمبلیک دارد.

ثابت های واقعی

ثابت واقعی (literal constant) مقداری است که مستقیما در کد برنامه تایپ می شود.

مثال. مقادیر 20 و 0.28 در عبارات زیر ثابت های واقعی هستند که در متغیرهای count و tax_rate ذخیره می شوند.

```
int count = 20;
float tax_rate = 0.28;
```

مقادیر ثابت برای نوع های پیش ساخته می تواند به صورت های دسیمال، اکتال، هگزادسیمال، ممیزشناور یا کاراکتر باشند.

نقطه اعشار نشان دهنده ثابت ممیزشناور است. ثابت های ممیزشناور را می توان به صورت نماد علمی هم بیان کرد.

مثال. اعداد زیر همگی ثابت های ممیزشناور هستند.

```
123.456
0.019
100.
1.23E2   یا 123
4.08e6   یا 4080000
0.85e-4   یا 0.000085
```

حروف f یا F برای ثابت های float و l یا L برای ثابت های long double را می توان به عنوان پسوند اضافه کرد در غیر اینصورت با ثابت ممیزشناور بعنوان یک عدد double برخورد می شود.

ثابتی که نقطه اعشار ندارد بعنوان ثابت صحیح تفسیر می شود. ثابت های صحیح را به ۳ صورت می توان نشان داد:

- ثابت دهدهی. اگر ثابت با هر عدد غیر صفری شروع شود بعنوان یک عدد صحیح مبنای ۱۰ تفسیر می شود.. ثابت های دهدهی شامل ارقام 0 تا 9 و علامت + یا – می توانند باشند. اگر علامت نوشته نشود ثابت مثبت درنظر گرفته می شود.
- ثابت اکتال. اگر ثابت با رقم 0 شروع شود بعنوان یک عدد صحیح مبنای ۸ تفسیر می شود. ثابت های اکتال می توانند شامل ارقام 0 تا 9 و علامت + یا – باشند.
- ثابت هگزادسیمال. اگر ثابت با 0x یا 0X شروع شود بعنوان یک ثابت مبنای ۱۶ با آن برخورد می شود. ثابت های هگزادسیمال می توانند شامل ارقام 0 تا 9 و حروف A تا F و علامت + یا – باشند.

ثابت های کاراکتری بین علامت کوتیشن (') قرارمی گیرند، مانند کاراکترهای 'A' و '0'. کاراکترهای خاص با علامت (\) نشان داده می شوند مانند '\n'(newline)، '\t'(tab)، '\\'(backslash)، '\r'(carriage return)، "\"(single quote)، '\'(double quotes)، و غیره.

ثابت های سمبلیک

ثابت سمبلیک (symbolic constant) ثابتی است که توسط یک نام در برنامه مشخص می شود. مقدار حقیقی ثابت سمبلیک تنها یکبار هنگام تعریف آن وارد می شود و این مقدار مشابه ثابت های واقعی در طول اجرای برنامه قابل تغییر نیست. هر زمان که به مقدار ثابت در برنامه نیاز باشد نام آن ذکر می شود.

مثال. برای محاسبه مساحت و محیط دایره در یک برنامه به جای نوشتن عدد 3.14 می توان ثابت PI را تعریف و استفاده کرد.

```
circumference = PI * (2 * radius);
area = PI * (radius)*(radius);
```

در این حالت هم نوشتن برنامه راحت تر است و برنامه دارای خوانائی بیشتر است هم اگر نیاز به تغییر ثابت پی به مقدار 3.14159 برای بالا بردن دقت محاسبات باشد اصلاح تنها در یک نقطه برنامه که ثابت تعریف شده است صورت می گیرد نه کل برنامه.

تعریف ثابت سمبلیک

دوروش برای تعریف ثابت های سمبلیک وجود دارد: راهنمای #define و کلمه کلیدی const.

راهنمای #define در قسمت پیش پردازنده ها بیشتر توضیح داده خواهد شد. فرم کلی آن به صورت زیر است:

```
#define CONSTNAME literal
```

CONSTNAME نام ثابت و literal مقدار واقعی آن است. راهنمای #define به کامپایلر دستور می دهد که درسرتاسر کد برنامه *CONSTNAME* را با *literal* جایگزین کند.

نام ثابت از همان قواعد نامگذاری متغیرها تبعیت می کند. مرسوم است که کلیه حروف ثابت را بزرگ می نویسند که باعث تشخیص راحت تر آن از متغیرها که با حروف کوچک هستند می شود.

مثال. ثابت PI با مقدار 3.14159 تعریف شده است.

```
#define PI 3.14159
```

دقت کنید که خط #define به سمیکولن(;) ختم نمی شود. #define می تواند در هر کجای برنامه باشد ولی معمولا همگی در قسمتی از ابتدای کد برنامه و قبل از تابع ()main هستند.

روش دوم تعریف یک ثابت سمبلیک استفاده از کلمه کلیدی const است. const را می توان به اول هر اعلان متغیری اضافه کرد. در این صورت متغیر در طول اجرای برنامه قابل تغییر نخواهد بود. اگر برنامه سعی به تغییر آن کند خطای کامپایلر صادر خواهد شد.

مثال.

```
const int count = 100;
const float pi = 3.14159;
const long debt = 12000000, float tax_rate = 0.21;
```

const روی کلیه متغیرهای خط اعلان تاثیر می گذارد.

تفاوت ثابت هائی که با #define و const تعریف می شوند در حوزه آنهاست. وقتی از #define استفاده می کنید کنترل ثابت ها را از حوزه کامپایلر خارج می کنید؛ type checking روی اسم آن انجام نمی گیرد و آدرس آنرا قابل بازیابی نیست و اشاره گریا ارجاعی به آن ممکن نیست و نمی تواند از نوع user-defined باشد. اما اگر ثابت توسط const تعریف شده باشد می تواند از هر نوع داده استاندارد یا user-defined باشد. آدرس آن هم قابل بازیابی است و مانند یک متغیر دارای حوزه دسترسی است بنابراین ثابتی که درون یک تابع تعریف می شود در سایر نقاط برنامه شناخته شده نیست.

درحالیکه const به کامپایلر می گوید که مقداری تغییر نمی کند و باعث می شود کامپایلر بهینه سازی اضافی انجام بدهد، کلمه کلیدی volatile باعث می شود که مانع هرگونه عمل بهینه سازی از جانب کامپایلر روی متغیر شود. این کلمه وقتی می خواهید مقداری خارج از کنترل کد برنامه خود را بخوانید مانند ثباتی در یک سخت افزار ارتباطی استفاده کنید. متغیر volatile هرزمان که مقدارش نیاز باشد خوانده می شود حتی اگر خط خواندن آن خط قبلی باشد.

اگر عدد ممیز داری به یک متغیر صحیح داده شود بخش اعشار آن حذف می شود.

اگر عددی در متغیری با نوع کوچکتر ذخیره شود بخشی از داده ممکن است حذف شود.

1. In what variable type would you best store the following values?
- **a.** A person's age to the nearest year.
- **b.** A person's weight in pounds.
- **c.** The radius of a circle.
- **d.** Your annual salary.
- **e.** The cost of an item.
- **f.** The highest grade on a test (assume it is always 100).
- **g.** The temperature.
- **h.** A person's net worth.
- **i.** The distance to a star in miles.

2. Determine appropriate variable names for the values in exercise 1.

3. Write declarations for the variables in exercise 2.

4. Which of the following variable names are valid?
- **a.** 123variable
- **b.** x
- **c.** total_score
- **d.** Weight_in_#s
- **e.** one.0
- **f.** gross-cost
- **g.** RADIUS
- **h.** Radius
- **i.** radius
- **j.** this_is_a_variable_to_hold_the_width_of_a_box