



کلیات و مفاهیم

مقدمه

در اوایل دهه 1970 میلادی، «دنيس ريچي» در آزمایشگاه کمپاني بل زبان C را برای برنامه‌نویسی سیستمها طراحی کرد. این زبان از دو زبان پیشین به نامهای BCPL و B منتج شده است که این دو نیز در همین آزمایشگاه نوشته شدند. زبان C تا سال 1978 منحصر به استفاده در همین آزمایشگاه بود تا اینکه «ريچي» و «کرنیه» نسخه نهایی این زبان را منتشر کردند. به سرعت مفسرهای متعددی از C تهیه شد، لیکن برای جلوگیری از ناسازگاریهای ایجاد شده و نیز حفظ قابلیت حمل زبان، استاندارد ANSI تعاریف متحدالشکلی مطرح کرد. آنچه در این کتاب مطرح می‌شود بر اساس همین استاندارد است. مفسر خود برنامه‌ای کامپوتري است که برنامه سطح بالا داده ورودی آن و برنامه ایجاد شده به زبان ماشین خروجی آن را تشکیل می‌دهد.

به طور کلی ویژگیهای مهم زبان C به اختصار به شرح زیر است:

- زبان C به طور گسترده‌ای در دسترس است. مفسرهای تجارتي آن در بیشتر کامپیوترهای شخصی، مینی‌کامپیوترها و نیز در کامپیوترهای بزرگ قابل استفاده‌اند.
 - C زبانی است همه منظوره، ساخت‌یافته سطح بالا (مانند زبان پاسکال و فورترن) و انعطاف‌پذیر که برخی از خصوصیات زبانهای سطح پایین را نیز که معمولاً در اسمبلی یا زبان ماشین موجود است داراست. در عین حال این زبان برای کاربردهای ویژه طراحی نشده و می‌توان از آن در همه زمینه‌ها، بخصوص به دلیل نزدیکی آن به زبان ماشین در برنامه‌نویسی سیستم، استفاده کرد. بنابراین C بین زبانهای سطح بالا و سطح پایین قرار دارد و در نتیجه اجازه می‌دهد که برنامه‌نویس خصوصیات هر دو گروه زبان را به کار برد. از این رو در بسیاری از کاربردهای مهندسی به طور انحصاری زبان C به کار می‌برند. (زبانهای سطح بالا، دستورالعملهایی شبیه زبان انسان و پردازش فکری او دارند، همچنین یک دستورالعمل زبان سطح بالا معادل چند دستورالعمل به زبان ماشین است.)
 - برنامه‌های نوشته شده به زبان C به طور کلی مستقل از ماشین یا نوع کامپیوتر است و تقریباً تحت کنترل هر سیستم عاملی اجرا می‌شود.
 - مفسرهای C معمولاً فشرده و کم حجم‌اند و برنامه‌های هدف ایجاد شده با آنها در مقایسه با سایر زبانهای برنامه‌سازی سطح بالا، خیلی کوچک و کارآمدند.
 - برنامه‌های C در مقایسه با سایر زبانهای برنامه‌سازی سطح بالا، به راحتی قابل انتقال‌اند. دلیل آن این است که C خیلی از ویژگیهای وابسته به نوع کامپیوتر را در توابع کتابخانه‌ای خود منظور داشته است. بنابراین هر نسخه از C با مجموعه‌ای از توابع کتابخانه‌ای مخصوص به خود همراه است که براساس ویژگیهای کامپیوتر میزبان مربوط نوشته شده است. این توابع کتابخانه‌ای تا حدودی استاندارد است و معمولاً هر تابع کتابخانه‌ای در نسخه‌های متعدد C به شکل یکسان در دسترس است.
 - C روش برنامه‌سازی ماژولار را پشتیبانی می‌کند. همچنین از نظر عملگرها نیز زبانی قوی است که عملگرهای گوناگونی برای دستکاری روی داده‌ها در سطح بیت داراست.
 - به طور کلی جامعیت، عمومیت، خوانایی، سادگی، کارایی و پیمانه‌ای بودن که همگی از مشخصات برنامه‌ای ایده‌آل‌اند در زبان C پیاده‌سازی می‌شوند.
- ویژگیهای فوق موجب شده زبان C یکی از قوی‌ترین و محبوب‌ترین زبانهای برنامه‌سازی در دنیا مطرح شود.

کاراکترها

زبان برنامه‌نویسی C مجموعه‌ای خاص از کاراکترها را شناسایی می‌کند. این مجموعه که در حکم مصالح اولیه جهت شکل دادن به اجزای اصلی برنامه‌اند عبارت‌اند از:

- حروف بزرگ و حروف کوچک. زبان C برخلاف زبان پاسکال بین حروف بزرگ و کوچک فرق می‌گذارد. مثلاً FOR با for یکسان نیست.

- ارقام دهدهی. شامل 0 تا 9

- جای خالی یا Blank

- کاراکترهای مخصوص. شامل ؟ * \ / = - + @ # \$ % ^ () ; ~ [] { } < > . , &

- کاراکترهای فرمت‌بندی. که برای بیان حالت‌های ویژه‌ای به کار می‌روند و عبارت‌اند از

کاراکتر Horizontal Tab یا \t، کاراکتر Vertical Tab یا \v، کاراکتر خط جدید یا \n، کاراکتر برگشت به عقب (Back Space) یا \b، کاراکتر تغذیه فرم (Form Feed) یا \f، کاراکتر ابتدای سطر یا \r، کاراکتر تهی یا \0، و جز آن که آنها را کاراکترهای فرمان نیز می‌نامند. لازم به توضیح است که هر کاراکتر فرمان بیان‌کننده تنها یک کاراکتر است هر چند که با دو یا چند کاراکتر نوشته می‌شود.

شناسه‌ها

شناسه‌ها نشانه‌های سمبولیکی‌اند که برای مراجعه به انواع داده‌ها مانند مقادیر ثابت، متغیرها، نوعها و توابع به کار می‌روند. به عبارتی دیگر شناسه نامی است که به عناصر مختلف برنامه مانند متغیرها، توابع، آرایه‌ها، اشاره‌گرها و جز آن اختصاص می‌یابد. یک شناسه C دنباله‌ای است از حروف، ارقام یا علامت زیرخط که با هر ترتیبی می‌توانند قرار گیرند. اما اولین کاراکتر باید حرفی باشد. در شناسه‌ها هر دو شکل حروف بزرگ و کوچک مجاز شناخته می‌شوند. برحسب قرارداد شناسه‌هایی که با علامت زیرخط شروع می‌شوند فقط در برنامه‌های سیستم کاربرد دارند و در برنامه‌های کاربردی قابل استفاده نیستند. در مورد تعداد کاراکترها در اسامی از نظر C محدودیتی وجود ندارد گرچه هر مفسر ویژگیها و محدودیتهای خاص خودش را به کار می‌برد. طول اسامی در زبان C استاندارد تا 31 کاراکتر مجاز است.

☞ مثال 1-1 اسامی زیر شناسه‌های معتبرند.

x1 , sum , payam_noor , maximum

☞

☞ مثال 2-1 اسامی زیر شناسه‌های غیرمعتبرند.

book-5 , 4s , \$tax , "p" , number one

☞

متغیرها

متغیرها در زبان C شناسه‌هایی‌اند که محل‌هایی از حافظه را به خود اختصاص می‌دهند. به عبارت دیگر متغیر شناسه‌ای است که در میان بخشی از برنامه برای نسبت دادن نوع تعیین شده‌ای از اطلاعات مورد استفاده قرار می‌گیرد. در ساده‌ترین حالت هر متغیر جانشین یک قلم داده می‌شود. متغیر ترکیبی است از ارقام، حروف و علامت زیرخط (_). لازم به ذکر است متغیرهایی که با علامت زیرخط شروع می‌شوند برای متغیرهای داخلی سیستم رزرو شده‌اند. طول هر متغیر در استاندارد ANSI تا 31 کاراکتر است، ولی در بعضی از مفسرهای قدیمی محدود به 8 کاراکتر است. تمامی متغیرهایی که در برنامه‌های C به کار می‌روند باید تعریف یا اعلان شوند. به یک متغیر می‌توان داده‌های مختلفی در محل‌های گوناگون برنامه نسبت داد. بنابراین مقدار متغیر در طول اجرای برنامه ممکن است تغییر کند.

بعضی از شناسه‌های زبان C کلمات رزرو شده یا کلیدی‌اند. یعنی معنی و مفهوم آن از قبل در زبان تعریف و پیش‌بینی شده است. بنابراین شناسه‌های تعریف شده برنامه‌نویس نیستند. متداول‌ترین کلمات کلیدی زبان C در زیر نشان داده شده است.

main	int	float	Char	If	else	goto	for
double	do	while	default	signed	return	register	enum
const	continue	short	Case	auto	struct	static	sizeof
break	long	switch	Void	typedef	extern	unsigned	union

البته در بعضی از مفسرهای زبان C ممکن است کلمات کلیدی دیگری نیز وجود داشته باشد که باید به کتاب راهنمای مربوط مراجعه کرد. همچنین توجه داشته باشید که همه کلمات کلیدی با حروف کوچک نوشته می‌شوند. پس کلمه کلیدی است درحالی که Main کلمه کلیدی نیست زیرا حرف اول آن بزرگ است. همین طور void کلمه کلیدی است اما VOID کلمه کلیدی نیست.

علامت توضیح

در زبان C هر عبارتی که بین دو علامت /* و */ قرار گیرد صرفاً توضیح محسوب می‌شود. مثلاً اگر بخواهیم در مورد دستوری توضیح دهیم که چه کاری انجام می‌دهد در هر جای برنامه که فضای خالی مجاز باشد می‌توان برای توضیح از علامت فوق استفاده کرد. (در اغلب نسخه‌ها علامت // هم مجاز است.)

☞ مثال 3-1 در برنامه زیر از علامت توضیح استفاده شده است.

```
#include<stdio.h>
```

```
main ()
```

```
{
    int j, k;
    for (j =1 ; j <= 10 ; j+ +) /* outer loop */
    {
        printf("%5d ", j);
        for (k=1; k<=10; k+ +) /* inner loop */
            printf("%5d", j * k);
        printf("\n");
    }
}
```

```

}
}

```

ساختار برنامه‌های C

در زبان C برنامه‌ها با فرمت خاصی نوشته می‌شوند. همه برنامه‌های C شامل يك يا چندین تابع اند که فقط یکی از آنها تابع اصلی یا main نامیده می‌شود. بنابراین هر برنامه فقط يك تابع اصلی دارد. برنامه همیشه با اجرای تابع اصلی آغاز می‌گردد. معرفی توابع دیگر ممکن است قبل و یا بعد از تابع اصلی قرار گیرد. به طور کلی می‌توان گفت که هر برنامه به زبان C حداقل دارای اجزای مقدماتی به ترتیب زیر است.

main()	تابع اصلی
{	شروع تابع اصلی
variables declaration ;	تعریف متغیرها
program statements ;	دستورات برنامه
}	پایان تابع اصلی

دستورالعملهای اجرایی

در هر برنامه دستورالعملهای اجرایی باید بعد از تعریف متغیرها درج شوند. دستوری قابلیت اجرا دارد که در پایان آن دستور، علامت سمیکولون (;) نوشته شود. برای فهم بهتر این موضوع به مثال زیر توجه کنید.

مثال 4.1 برنامه‌ای بنویسید که مساحت مستطیلی به طول 6 و عرض 3 را محاسبه و چاپ کند.

```

#include<stdio.h>
main()
{
    int length , width , S ; /* variable declaration */
    length = 6 ;
    width = 3 ;
    S = length * width ;
    printf ("area = %d", S) ;
}

```

پس از اجرای برنامه، خروجی برنامه به صورت زیر نمایش داده می‌شود.

```
area = 18
```

توضیح. برنامه با حروف کوچک تایپ شده است. توضیحات نیز معمولاً با حروف کوچک تایپ می‌شوند. در زبان C حروف بزرگ و کوچک معادل یکدیگر نیستند. خط اول برنامه اعلان می‌کند که کتابخانه مربوط به توابع ورودی و خروجی برای دستیابی به توابع آن آماده شود. در واقع ارجاع به فایل است که شامل اطلاعاتی است که هنگام تفسیر باید در برنامه قرار گیرند. C یکی از زبانهای است که به لحاظ داشتن توابع توکار یا از پیش فرض شده بسیار غنی است. هر مجموعه از توابع که عملیات ویژه‌ای را انجام می‌دهد در مجموعه‌ای با عنوان کتابخانه یا library قرار می‌گیرد. توابعی که عملیات ورودی و خروجی را انجام می‌دهند در کتابخانه‌ای به نام stdio.h قرار دارند که در آن stdio به معنی standard input output (ورودی و خروجی استاندارد) و h نیز معرف header یا عنوان است. تابع printf نیز یکی از توابع خروجی است.

حال به توضیح #include می‌پردازیم. برنامه‌های نوشته شده به زبان C قبل از اینکه به وسیله مفسر ترجمه شوند در اختیار برنامه دیگری با عنوان پیش‌پردازنده یا preprocessor قرار می‌گیرند. یکی از کاربردهای اصلی این برنامه آن است که کتابخانه‌های مورد نیاز برنامه منبع را یعنی کتابخانه‌هایی را که توابع به کار رفته در برنامه منبع را شامل است برای استفاده آماده می‌کند. این کار با دستور include که در ابتدای آن علامت # و به دنبال آن نام کتابخانه در داخل علامت <> یا " " می‌آید انجام می‌گیرد. اولین عبارت در برنامه بالا همین کار را برای ما انجام می‌دهد.

در خط بعد تابع اصلی تعریف شده است. برانتز خالی به دنبال نام تابع بیان می‌کند که این

تابع آرگومانی ندارد. چند خط بعدی، متن برنامه اصلی را تشکیل می‌دهند که از پنج دستور ساده تشکیل شده است. پایان هر دستور را علامت سمیکولون (;) مشخص می‌نماید. چون متن برنامه بیش از یک دستور است مجموع آنها که دستور مرکب یا بلاک را تشکیل می‌دهند در داخل یک زوج آکولاد قرار می‌گیرند. در واقع هر آکولاد چپ برای مفسر C به معنی شروع بلاک و هر آکولاد راست معرف پایان آن است. البته در بعضی مواقع برحسب مورد آکولادها را می‌توان به صورت تودرتو نیز به کار برد.

اولین دستور در متن برنامه یا همان شروع آکولاد، توصیف متغیرهاست. سه دستور بعدی دستورات محاسباتی و جایگزینی‌اند و در آخر نیز دستور خروجی است که در آن تابع printf برای چاپ فرمت‌دار است. اولین آرگومان تابع مذکور متن داخل گیومه است که تابع آن را به همان صورت در خروجی چاپ می‌کند. البته بعضی از قسمتهای متن که شامل علامت % باشد مانند %d به مفسر اطلاع می‌دهد که اولین متغیر بعد از بسته شدن گیومه که در این مثال S است مقادیر صحیح می‌پذیرد. در اینجا d معرف decimal است و %d فرمت متغیر در خروجی را تعریف می‌کند. در فصلهای بعدی به طور کامل به بحث فرمت متغیرها خواهیم پرداخت.

☞

☞ مثال 5.1 برنامه زیر طول و عرض مستطیلی را از طریق ورودی استاندارد می‌خواند و با فراخواندن تابعی به نام Rectangle مساحت آن را محاسبه می‌کند و سپس طول و عرض و مساحت را در دستگاه استاندارد خروجی نمایش می‌دهد. طول با a و عرض با b و مساحت با area مشخص می‌شود.

```
# include <stdio. h>
main ()
{
    int a , b , area ;
    int Rectangle (int a , int b) ;
    scanf ("%d %d" , &a , &b) ;
    area = Rectangle (a , b);
    printf ("\n length = %d width = %d area = %d" , a , b , area) ;
}
int Rectangle (int a , int b) ;
{
    int s ;
    s = a * b ;
    return (s) ;
}
```

اگر $a = 5$ و $b = 4$ ، خروجی برنامه مذکور به صورت زیر خواهد بود.

```
length = 5 width = 4 area = 20
```

توضیح. خط اول ارجاع به فایل کتابخانه‌ای، خط دوم تعریف تابع اصلی، خط سوم شروع تابع اصلی، و خط چهارم اعلان متغیرهاست. در خط پنجم تابع فرعی Rectangle اعلان شده است که مقدار صحیح برمی‌گرداند و آرگومانهای آن نیز a و b اند که مقادیر صحیح‌اند. در خط ششم، تابع ورودی استاندارد scanf به کار رفته است. این تابع که جزء کتابخانه stdio.h است، اطلاعات را از طریق ورودی استاندارد که صفحه کلید است دریافت می‌کند. فرمت و ساز و کار این گونه توابع را در فصلهای بعد بررسی می‌کنیم. در اینجا یادآور می‌شویم که فرم کلی تابع مزبور به صورت زیر است.

```
scanf (control string , arg1 , arg2 , ... , argn) ;
```

که در آن رشته کنترل که در داخل گیومه (" ") می‌آید، اطلاعات مورد نیاز درباره فرمت اقلام داده‌های ورودی را شامل می‌شود و عناصر arg1 , arg2 , ... , argn نیز آرگومانهایی‌اند که اقلام داده‌های ورودی را معرفی می‌کنند. در این دستور کاراکتر "&" عملگر آدرس است. در رشته کنترلی نیز که در داخل گیومه قرار دارد، از چپ به راست %d اول معرف فرمت اولین آرگومان یا داده ورودی به عنوان عدد صحیح است و به دنبال آن %d دوم نیز معرف فرمت دومین آرگومان به عنوان عدد صحیح است. علامت \n در رشته کنترلی تابع printf، موجب انتقال به سطر جدید می‌گردد. بنابراین اطلاعات بعدی در سطر جدید چاپ خواهد شد.

☞

خودآزمایی 1

1. کدام یک از اسامی زیر مجاز است به عنوان نام متغیر در برنامه به کار رود؟

Integer , -19, Lesson four , Unit_25, define , Loop2, Star565void , Please?, Payam_noor , C+ +, S#, Five\$

2. برنامه زیر مساحت مربعی به ضلع 5 سانتی متر را محاسبه و چاپ می کند. قسمتهای مختلف آن را شرح دهید.

```
#include<stdio.h>
main()
{
    int x , S ;
    x = 5 ;
    S = x * x ;
    printf ("area = %d", S) ;
}
```

3. ویژگیهای برنامه ایده آل را شرح دهید.

4. چند نمونه از زبانهای سطح بالا، سطح پایین و سطح میانی را نام ببرید.

شبکه آموزشی - پژوهشی مادیج
با هدف بهبود پیشرفت علمی
و دسترسی راحت به اطلاعات
برای جامعه بزرگ علمی ایران
ایجاد شده است



madsg.com
مادیج

IRan Education & Research NETWORK
(IRERNET)

