

برنامه نویسی مقدماتی

بررسی مفاهیم الگوریتم و فلوچارت

به کارگیری زبان ++C و C جهت پیاده سازی الگوریتم ها

مدرس : علی نیک فرجام

توابع

(لیست پارامترها) نام تابع نوع خروجی تابع

{

دستورات

...

}

```
int factorial ( int n )
{
    int i , f ;
    for ( i = 1 , f = 1 ; i <= n ; i ++ )
        f = f * i ;
    return f ;
}
```

فراخوانی تابع

```
int fact ;  
fact = factorial ( 12 ) ;
```

در صورتی که پارامتری نداشته باشد ذکر () الزامی است. ✍️

```
ret = f ( ) ;
```

✍ برای تابعی که خروجی ندارد از کلمه `void` استفاده می‌شود.

```
void f ( int i , float j )
```

در برنامه C همه توابع در یک سطح هستند به این معنی که در داخل یک تابع نمی‌توان تابع دیگری تعریف کرد.

نکاتی در مورد نوشتن توابع

بدون پرداختن به جزئیات پیاده‌سازی، پارامترها و خروجی را طراحی کنید.

تابع باید فقط به آنچه نیاز دارد دسترسی داشته باشد (Information hiding)

برای ارتباط با تابع از پارامترها استفاده کنید.

پارامترهای تابع

: Call by value ✍️

تغییر پارامتر در داخل تابع تاثیری بر فراخواننده ندارد. ✍️

: Call by reference ✍️

تغییر پارامتر در داخل تابع بر فراخواننده تاثیر دارد. ✍️

شکل کلی یک برنامه C

include section
Global Variables

تابع ۱

⋮

تابع n

main تابع

نکات مربوط به متغیرها

✍ محدوده شناسایی متغیر (Scope)

✍ طول عمر متغیر (Life time)

انواع متغیرها

✍️ **عمومی: خارج از توابع تعریف می‌شوند.**

✍️ **محدوده: از محل تعریف تا انتهای برنامه**

✍️ **طول عمر: از شروع اجرای برنامه تا پایان آن**

✍️ **محلی: در داخل یک تابع تعریف می‌شوند.**

✍️ **محدوده: در داخل تابعی که تعریف شده‌اند.**

✍️ **طول عمر: با شروع اجرای تابع، ایجاد و با پایان اجرای آن از**

بین می‌روند.

مسألهٔ همنام بودن متغیرها

✍️ نزدیکترین تعریف در نظر گرفته می‌شود.
✍️ (ارجحیت تعریف محلی به عمومی)

```
#include <stdio.h>
int i , j ;
int f1 ( int j )
{
    j = j + i ;
    return j * j ;
}
```

مثال

```
int k ;
void f2 ( void )
{
    int n ;
    scanf ("%d" , &n);
    j = f1 (n) + k ;
}
void main ()
{
    int i ;
    for (i = 0 , k = 10 ; i < 10 ; i ++ )
    {
        f2 () ;
        printf ("j=%d" , j) ;
    }
}
```

متغیرهای استاتیک محلی

`static` نوع متغیر نام متغیر;
`static int s ;`

ویژگی‌های متغیرهای استاتیک محلی

✍ طول عمر:

✍ با اولین اجرای تابع ایجاد شده و تا پایان اجرای برنامه باقی می‌مانند.

✍ در نتیجه: با خروج از تابع مقدار خود را حفظ می‌کنند

✍ متغیرهای استاتیک محلی فقط یک بار مقدار اولیه می‌گیرند.

✍ `static int s = 0 ;`