

یادگیری ساختارهای کنترلی  
آشنایی با ساختارهای دو و چند انتخابی  
آشنایی با ساختارهای تکرار و یادگیری مفهوم حلقه  
ترکیب ساختارهای کنترلی به صورت تو در تو و پشت‌پشتی

---

## دستورهای کنترلی



# الگوریتم

- روش حل گام به گام یک مسئله الگوریتم نامیده می شود.
- الگوریتم: هر رویه‌ای که برای حل مسئله مشخص کند
  1. عمل‌هایی که باید اجرا شوند
  2. ترتیب این عمل‌ها
- الگوریتم برای جمع دو عدد:
  1. اطلاع به کاربر برای ورود عدد صحیح اول
  2. دریافت و ذخیره‌ی عدد صحیح اول
  3. اطلاع به کاربر برای ورود عدد صحیح دوم
  4. دریافت و ذخیره‌ی عدد صحیح دوم
  5. جمع عدد اول و دوم و ذخیره‌ی نتیجه
  6. نمایش نتیجه در خروجی
- معمولا از نمودارهایی (مانند فلوچارت) برای شرح الگوریتم‌ها استفاده می شود.



## کنترل برنامه

- **کنترل برنامه:** مشخص کردن ترتیب اجرای دستورها در یک برنامه
- توجه به ترتیب اجرای دستورها بسیار اهمیت دارد:

$$x = 5 * y + 4 * x;$$

$$y = 3 * x * x - 7;$$

- **اجرای دنباله‌ای:** دستورها با همان ترتیبی که نوشته شده‌اند، یکی پس از دیگری اجرا می‌شوند.

- دستورها همواره به صورت دنباله‌ای اجرا می‌شوند اما می‌توان با استفاده از **دستورهای کنترلی** اجرای دنباله‌ای را تغییر داد و ترتیب اجرای دستورها را جا به جا کرد.

- دستورهای کنترلی:

– دستورهای انتخابی (تصمیم‌گیری):

– دستورهای تکرار (حلقه)



## دستورهای انتخاب

- گاهی لازم از بسته به شرایطی از بین چند دستور یکی را انتخاب و اجرا کرد.
- دستور `if` (دستور یک انتخابی): اگر ارزش یک شرط (گزاره) درست باشد، مجموعه‌ای از دستورها را اجرا می‌کند و در غیر این صورت از اجرای این دستورها صرف نظر می‌کند.
- دستور `if ... else` (دستور دو انتخابی): اگر ارزش یک شرط (گزاره) درست باشد، مجموعه‌ای از دستورها را اجرا می‌کند و در غیر این صورت مجموعه‌ای دیگر از دستورها اجرا می‌شود.
- دستور `switch` (دستور چند انتخابی): با توجه به مقدار یک عبارت صحیح، یکی از چند مجموعه‌ی دستور مختلف را اجرا می‌کند.



# دستورهای تکرار

- گاهی لازم است یک گروه از دستورها به طور مکرر اجرا شوند.
- دستورهای تکرار باعث می‌شوند مجموعه‌ای از دستورها (بدنه‌ی حلقه) تا زمانی که ارزش شرط خاصی (شرط ادامه‌ی تکرار حلقه) درست است به شکل مکرر اجرا شوند.

- دستورهای تکرار:

- `while`

- `do ... while`

- `for`

- چند دستورهای کنترلی را می‌توان با هم ترکیب کرد:
  - به صورت پشت‌پشتی: پایان یک دستور کنترلی، آغاز دستور کنترلی بعدی است.
  - به صورت تو در تو: یک دستور کنترلی درون دستور کنترلی دیگر قرار دارد.



## مثال ۴: استفاده از دستور `if ... else`

```
#include <iostream>
using namespace std;

int main()
{
    int grade;
    cout << "Please enter the grade: ";
    cin >> grade;

    if (grade >= 60)
        cout << "Passed" << endl;
    else
        cout << "Failed" << endl;

    return 0;
}
```



## شرح مثال ۴

- مجموعه‌ای از دستورها که درون یک جفت آکولاد باز و بسته قرار می‌گیرند، یک بلوک از دستورها یا یک دستور مرکب نامیده می‌شوند.
- اگر بیش از یک دستور در بدنه‌های دستور `if . . . else` داشتید، باید از بلوک‌ها استفاده کنید:

```
if ( شرط )
```

```
{
```

```
    if بدنه‌ی دستور
```

```
}
```

```
else
```

```
{
```

```
    else بدنه‌ی دستور
```

```
}
```

```
if ( studentGrade >= 60 )
    cout << "Passed.\n";
else
{
    cout << "Failed.\n";
    cout << "You must take this course again.\n";
}
```



## شرح مثال ۴

- عملگر شرطی **?:**
- این عملگر تنها عملگر سه تایی (با سه عملوند) در C++ است که ارتباط نزدیکی با دستور `if ... else` دارد.
- شکل کلی استفاده از این عملگر به صورت زیر است:  
(عملوند سوم) : (عملوند دوم) ? (عملوند اول)
- عملوند اول این عملگر یک شرط است: اگر درست باشد، آن گاه عملوند دوم آن ارزش (مقدار) کل عبارت شرطی است و اگر نادرست باشد، آن گاه عملوند سوم آن ارزش کل عبارت شرطی است.

```
cout << ( grade >= 60 ? "Passed" : "Failed" );
```

```
grade >= 60 ? cout << "Passed" : cout << "Failed";
```





## مثال ۵: استفاده از دستور `if ... else` تو در تو

```
#include <iostream>
using namespace std;

int main()
{
    int grade;
    cout << "Please enter the grade: ";
    cin >> grade;
    if (grade >= 90)
        cout << "A" << endl;
    else
        if (grade >= 80)
            cout << "B" << endl;
        else
            if (grade >= 70)
                cout << "C" << endl;
```



## مثال ۵: استفاده از دستور `if ... else` تو در تو

```
else
    if (grade >= 60)
        cout << "D" << endl;
    else
        cout << "F" << endl;

return 0;
}
```



## شرح مثال ۵

- می توان دستورهای شرطی تو در تو در مثال ۵ را به شکل زیر نیز نوشت

```
if ( studentGrade >= 90 ) // 90 and above gets "A"
    cout << "A";
else if ( studentGrade >= 80 ) // 80-89 gets "B"
    cout << "B";
else if ( studentGrade >= 70 ) // 70-79 gets "C"
    cout << "C";
else if ( studentGrade >= 60 ) // 60-69 gets "D"
    cout << "D";
else // less than 60 gets "F"
    cout << "F";
```

- کامپایلر همواره `else` در برنامه را با نخستین `if` قبل از آن مرتبط و متصل می کند، مگر آن که با قرار دادن آکولادهای مناسب در برنامه، خلاف آن از کامپایلر خواسته شود.



## شرح مثال ۵

- مسئله‌ی else سرگردان:

```
if ( x > 5 )
    if ( y > 5 )
        cout << "x and y are > 5";
    else
        cout << "x is <= 5";
```

- دستورهای بالا منجر به نمایش پیغام درست نمی‌شوند. در دستورهای پایین این مشکل اصلاح شده است:

```
if ( x > 5 )
{
    if ( y > 5 )
        cout << "x and y are > 5";
}
else
    cout << "x is <= 5";
```

