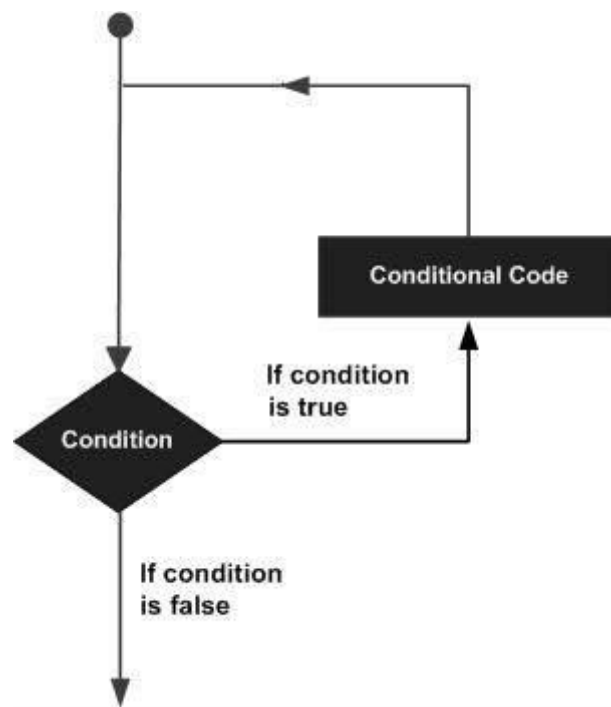


آموزش انواع / Loop حلقه در ++C

گاهی در یک برنامه لازم می شود که قطعه کدی معین را چندین بار تکرار (مجدد اجرا) نمایید. به طور کلی، دستورات به ترتیب و پی در پی اجرا می شوند به گونه ای که ابتدا اولین دستور در یک تابع اجرا می شود، سپس دومین و همین طور ادامه می یابد.

زبان های برنامه نویسی تعدادی ساختار کنترلی در اختیار توسعه دهنده قرار می دهند که امکان تغییر روند و ترتیب اجرا را برای برنامه فراهم می نماید.

دستور تکرار یا ساختار کنترلی حلقه این قابلیت را به برنامه نویس می دهد تا یک یا مجموعه ای از دستورات را چندین بار اجرا کند. در زیر شکل کلی دستور حلقه را در بیشتر زبان های برنامه نویسی مشاهده می کنید:

دستور goto در زبان ++C

دستور goto در زبان ++C، به منظور جا به جایی کنترل اجرای برنامه از یک نقطه به نقطه دیگری که با برچسب مشخص شده است، استفاده می شود.

نکته! استفاده از دستور **goto** در هر زبان برنامه نویسی بسیار دشوار است، زیرا باعث می شود تا جریان اجرای برنامه پیچیده تر شود. این موضوع موجب می شود تا درک کارکرد و اصلاح چنین برنامه هایی نیز سخت باشد. بنابراین توصیه می شود، از این دستور استفاده نکنید.

نحوه نوشتن دستور **goto**

در زیر **Syntax** یک دستور **goto** را مشاهده می کنید:

```
goto label;
..
label: statement;
```

1 **goto label;**

2 ..

3 .

4 **label:**

در **Syntax** بالا، **label** می تواند هر اسم دیگری به جزء کلمه های کلیدی ++C باشد، همچنین می توان آن را در هر جایی از برنامه قرار داد. به محض اینکه دستور **goto** اجرا شود، کنترل اجرای برنامه **label** مشخص شده می رود و از آن جا اجرا می شود.

پس **goto** یک **jump statement** غیر شرطی است. هنگامی که برنامه به این کلمه می‌رسد، به مکان مشخصی از کد که توسط **goto** مشخص شده است، پرش می‌کند.

goto سال‌ها قبل از چشم برنامه نویسان افتاد چراکه موجب می‌شد کد نویسی شما مانند اسپاگتی شود! اگرچه هیچ موقعیت برنامه نویسی به وجود نمی‌آید که به **goto** نیاز داشته باشید و در واقع سی برای این که یک زبان کامل باشد به **goto** نیاز ندارد اما هنوز هم به ندرت (و در برخی موارد به صورت مفید) استفاده می‌شود. با این تفاسیر، اگر از **goto** به صورت هوشمندانه استفاده شود می‌تواند سودمند باشد. نگرانی اصلی برنامه نویسان این است که استفاده زیاد از **goto** باعث شود برنامه ناخوانا و به هم ریخته شود اما در برخی از موارد برعکس است و به جای به هم ریختگی باعث واضح تر شدن کد خواهد شد. در هر حال استفاده از این دستور پیشنهاد نمی‌شود مگر به صورت هوشمندانه و به جا **goto**. برای انجام عملیات نیاز به یک **label** دارد **label**. یکی از شناسه‌های سی است که بعد از آن علامت دو نقطه (:): قرار می‌گیرد **label**. باید در همان بلاک و متدی باشد که **goto** قرار دارد (توضیح متد را در مقالات بعدی دنبال کنید).

تا جلسه ی بعد

خدانگهدار