مبانی کامپیوتر و برنامه سازی

استاد: ساسان حریفی

مثال هایی از ساختار if

```
int x=13, y=25 ;
if (x > y)
        cout << x ;
        cout << y ;
```

- خروجی قطعه کد زیر چیست؟

خروجی: ۲۵

مثال هایی از ساختار if

```
int x=2*9, y=13+4*20;
if (x > y)
{
        cout << x ;
        cout << y ;
}
```

- خروجی قطعه کد زیر چیست؟

خروجی: ندارد

مثال هایی از ساختار if

```
int a = -5;
int b = a*a;
if (a > b) ;
        cout << a ;
```

- خروجی قطعه کد زیر چیست؟

خروجی: ۵-

مثال هایی از ساختار if

```
int a = 5;
if (a < 0) ;
        cout << a ;
else
        cout << 0 ;
```

- خروجی قطعه کد زیر چیست؟

خروجی: خطا

ساختارهای کنترلی

- ساختار switch

```
switch (expression){
case value1:
        // statement
        break;
case value2:
        // statement
        break;
…
case valueN:
        // statement
        break;
default:
        // default statement
}
```

- در سوئیچ مقایسه ها فقط از نوع برابری هستند.
- مقادیر تکراری برای caseها مجاز نیست.
- اگر برای یک case دستوری وجود نداشته باشد، از دستورات case بعدی استفاده می کند.
- جهت خروجی از دستور مربوط به یک case باید از break; استفاده شود.
- می توان برای مقادیر caseها از کاراکتر نیز استفاده کرد.

ساختارهای کنترلی

- ساختار switch

- مثال) با استفاده از ساختار سوئیچ برنامه ای بنویسید که بر اساس عدد صحیح ورودی، معادل حروفی آن چاپ شود. عدد بین ۱ تا ۵ باشد.

ساختارهای کنترلی

- ساختار switch

- مثال) فرض کنید عددی توسط کاربر وارد می شود که نشان دهنده مقدار یک تاس بوده و در متغیری به نام d ذخیره می شود. حالات مختلف زوج یا فرد بودن عدد تاس را با استفاده از ساختار سوئیچ نمایش دهید.

ساختارهای کنترلی

- ساختار switch

- مثال) مثال قبل را در حالتی که مطمئن هستیم مقادیری که کاربر وارد می کند حتما بین ۱ تا ۶ است، پیاده سازی کنید.

ساختارهای کنترلی

- ساختار switch

- مثال) با استفاده از ساختار سوئیچ، برنامه ای بنویسید که دو عدد و یکی از عملگرهای جمع، تفریق، ضرب، تقسیم را از کاربر دریافت کند و با توجه به عملگر دریافتی حاصل را محاسبه کرده و نمایش دهد.

حلقه های تکرار (حلقه for)

- حلقه for

```
for (initialization ; condition ; iteration){
        دستورات
}
```

- مثال) حلقه ای بنویسید که مقادیر ۱ تا ۵ را در خروجی چاپ کند.

```
for (int i=1; i <= 5; i++){
        cout << i;
}
```

حلقه های تکرار (حلقه for)

- خروجی قطعه کد زیر چیست؟

```
int counter;
for (counter = 1 ; counter < 5 ; counter++ )
        cout << counter;
        cout << counter;
```

خروجی:

۱

۲

۳

۴

۵

حلقه های تکرار (حلقه for)

- خروجی قطعه کد زیر چیست؟

```
for (int counter = 2 ; counter < 4 ; counter+=1 )
        cout << counter;
```

خروجی:
۲
۳

حلقه های تکرار (حلقه for)

- خروجی قطعه کد زیر چیست؟

```
for (int x = 1 ; x < 5 ; x++)
        cout << x;
        cout << x;
```

خروجی: خطا

چون متغیر x در داخل حلقه تعریف شده است، فقط می تواند در داخل حلقه استفاده شود و چون برای حلقه از بلوک استفاده نشده است، پس خط آخر ربطی به حلقه for ندارد و چون x را نمی شناسد، خطا می دهد.

حلقه های تکرار (حلقه for)

- خروجی قطعه کد زیر چیست؟

```
int value;
for (value = 3 ; value > 0 ; --value) ;
        cout << value;
```

خروجی: ۰

استفاده از ; در انتهای تعریف حلقه باعث تهی شدن حلقه می شود. این یعنی دستور بعد از حلقه، ربطی به حلقه ندارد. در اینجا علت چاپ صفر، این است که مقدار اولیه برای value تعریف نشده و به صورت پیشفرض صفر چاپ می شود. اگر ; در انتهای حلقه وجود نداشته باشد، خروجی به ترتیب ۳، ۲، ۱ می شود.