

به نام خدا

در جلسات قبل، شما کلاس string را پیاده سازی کردید و توانستید یک رشته به طول دلخواه ایجاد کنید.

هدف تمرین این هفته آشنایی بیشتر با مفهوم operator overloading است، به همین منظور با استفاده از کلاس string که قبلاً نوشته اید عملگرهای +، *، -، =، <، > و << و >> و [] را پیاده سازی کنید.

توضیحات:

عملگر +: این عملگر دو رشته مجزا را دریافت میکند و رشته جدیدی متشکل از هر دو رشته درست میکند و مانند شبه کد زیر عمل میکند:

```
Str1 = "salam" , Str2="chetori?"
```

```
Str3 = str1 + str2
```

```
Str3 = "salamchetori?"
```

عملگر *: این عملگر یک رشته را بعنوان ورودی میگیرد و خروجی آن برابر تکرار آن رشته به تعداد عددی که در آن ضرب شده میباشد. مانند شبه کد زیر:

```
Str3 = Str1*3
```

```
Str3 = "salamsalamsalam"
```

عملگر -: این عملگر یک رشته و یک عدد را بعنوان ورودی میگیرد و رشته جدیدی درست میکند که به تعداد عدد داده شده، کاراکترهای آخر از رشته اصلی در آن نیست. مانند شبه کد زیر:

```
Str3 = str1 - 2
```

```
Str3 = "sal"
```

عملگر =: این عملگر دو رشته را با هم مقایسه میکند و خروجی آن یک boolean است. true هنگامی که دو رشته باهم برابر باشند و false وقتی که برابر نباشند.

عملگرهای << (شیفت به چپ) و >> (شیفت به راست): این عملگرهای یک رشته و یک عدد را میگیرند و به تعداد عدد کاراکترهای رشته را به چپ یا به راست شیفت میدهند.

در شیفت به چپ، همه ی کاراکترها به اندازه عدد داده شده به چپ شیفت داده میشوند و کاراکترهای ابتدایی (چپ ترین) به انتها رشته شیفت داده میشوند.

در شیفت به راست >، همه ی کاراکترها به اندازه عدد داده شده به راست شیفت داده میشوند و کاراکترهای انتهایی (راست ترین) به ابتدا رشته شیفت داده میشوند. مانند شبه کد زیر:

```
Str3 = str1>3
```

```
Str3 = "lamsa"
```

```
Str4 = Str1<3
```

```
Str4 = "amsal"
```

عملگرهای << و >>: این عملگرها را برای دریافت (insertion) و چاپ (extraction) اور لود (overload) کنید.

عملگر []: این عملگر امکان دسترسی به آرمین کاراکتر از رشته را به ما میدهد. مثل شبه کد زیر:

```
Char c = st1[2]
```

```
c = 'l'
```