

Priskyrimo operatorius

# Didinimo vienetu komanda (++)

- Užrašas **kiekis++** atitinka operaciją  
**kiekis = kiekis + 1**
- Gali būti **++kiekis** ir **kiekis++**

```
B = 10;  
A = B++;  
// A = 10, B = 11;  
Kintamojo B reikšmė priskiriama kintamajam A, tada B reikšmė padidinama
```

```
B = 11;  
A = ++B;    // A = 12, B = 12  
B reikšmė padidinama, ir gauta reikšmė priskiriama kintamajam A.
```

# Operatoriai

Operatorius	Pavyzdys	Atitikmuo
+=	a += 10	a = a + 10;
-=	a -= 10	a = a - 10;
*=	a *= 10	a = a * 10;
/=	a /= 10	a = a / 10;
%=	a %= 10	a = a % 10;

# Ciklas **for**

Ciklas **for** paprastai taikomas tada, kai jo veiksmų kartojimo skaičius žinomas.

**for** (pradinės reikšmės priskyrimas; sąlyga; padidinimas) sakiny;

```
for (pradinės reikšmės priskyrimas; sąlyga; padidinimas)
{
    sakiny;
    .....
    sakiny;
}
```

# Ciklas for (pavyzdžiai)

Sudarykime skaičių nuo 10 iki 20 kubų lentelę, keisdami skaičius reikšmes kas 1.

```
int sk, kumas;
for ( sk = 10; sk <= 20; sk++) {
    kumas = sk * sk *sk;
    cout << sk << " " << kumas << endl;
}
```

```
int sk, kumas;
for ( sk = 20; sk >= 10; sk--) {
    kumas = sk * sk *sk;
    cout << sk << " " << kumas << endl;
}
```

**Pavyzdys.** Programa išveda kas penktą skaičių.

```
#include<iostream>
using namespace std;
int main(void)
{
    int skaicius;
    for (skaicius =1; skaicius <=25; skaicius +=5) // skaicius = skaicius+5
        cout << skaicius << " ";
    return 0;
}
```

C:\ "C:\Documents and Settings\bendras\Desktop\Kursai\bin\Debug\Kursai.exe"

```
1 6 11 16 21
Process returned 0 (0x0)   execution time : 0.016 s
Press any key to continue.
```

# Užduotis (1)

Mokslininkai nustatė, kad norint sužinoti, kiek laiko reikia miegoti žmogui, ne vyresniam kaip 18 metų, reikia naudotis formule  $t = 8 + (18 - a)/2$ ; čia  $t$  – laikas valandomis,  $a$  – amžius metais. Sukursime programą, kuri apskaičiuos, kiek laiko turi miegoti žmogus nuo kūdikystės iki 18 metų. Skaičiavimų rezultatus pateiksime lentelėje.

# Sprendimas

```
// Miegas
#include <iostream>
#include <iomanip>
using namespace std;
int main ()
{
    setlocale (LC_CTYPE, "");
    int a;
    double t;
    cout << "Žmogaus amžius, m Laikas miegui, val." << endl;
    for (a = 0; a <= 18; a++) {
        t = 8 + (double) (18 - a) / 2;
        cout << setw (9) << a << setw (20) << fixed << setprecision
(1) << t << endl;
    }
    return 0;
}
```



## Užduotis (2)

Žinomo kartojimų skaičiaus ciklą labai patogu taikyti, kai reikia apskaičiuoti funkcijos reikšmes, kai argumentai yra sveikieji skaičiai ir kinta intervale  $[x_1; x_2]$  žingsniu lygiu 1.

Sukurkite programą, skaičiuojančią funkcijos  $y = 7x^2 + 5x - 3$  reikšmes, kai  $x$  kinta  $[-10; 10]$  žingsniu lygiu 1

# Sprendimas

```
// Vieta programos vardui įrašyti
# include <iostream>
# include <iomanip>
using namespace std;
int main ()
{
    setlocale (LC_CTYPE, "");
    int x, y;
    cout << "X _____ Y" << endl;
    cout << " | " << endl;
    for (x = -10; x <= 10; x++) {
        y = 7 * x * x + 5 * x - 3;
        cout << setw (4) << x << " | " << setw (6) << fixed << y << endl;
    }
    return 0;
}
```

# Užduotis (3)

Kiek kartų atliekamas ciklas ir kokios bus galutinės kintamųjų reikšmės?

```
b = 9;  
for (a = 5; a <= 7; a++) b = b + 3;
```

Ciklas atliekamas 3 kartus,  
a = 8, b = 18.

```
m = 6; n = 9; b = 0;  
for (a = m; a <= n; a++) b = b + a;
```

Ciklas atliekamas 4 kartus,  
a = 10, b = 30, m = 6, n = 9.

```
m = 4; n = 6; b = 2;  
for (a = m; a <= n; a++) b = b - a;  
b = b + 3;
```

Ciklas atliekamas 3 kartus,  
a = 7, b = -10, m = 4, n = 6

```
w = 5; t = 4; s = 4; m = 2;  
for (a = t; a <= w; a++) {  
    s = s + m;  
    m = m + a; }  
}
```

Ciklas atliekamas 2 kartus,  
a = 6, s = 12, m = 11, w = 5,  
t = 4