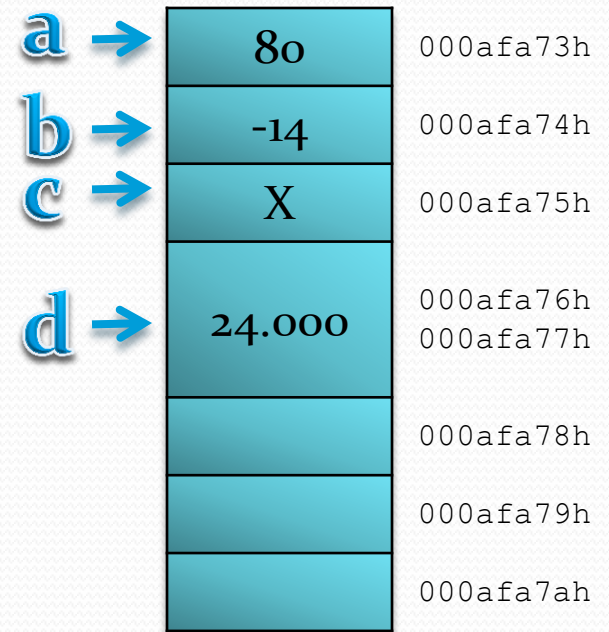


PROGRAMLAMA TEMELLERİ

Değişkenler ve C++

Değişken

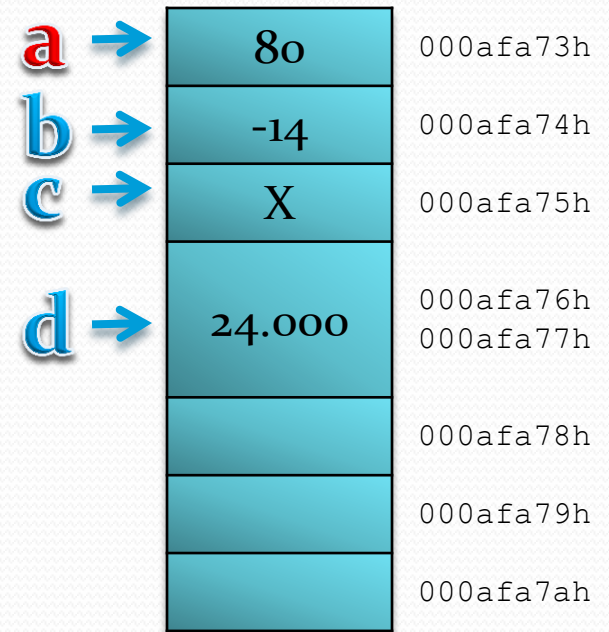
- İsimlendirilmiş bellek alanlarına değişken denir. Değişkenlerin özellikleri:
 - Ad
 - Boyut
 - Değer
 - Tip
 - Adres (Şimdilik bu konuya girmeyeceğiz!)



Bellek

Değişken

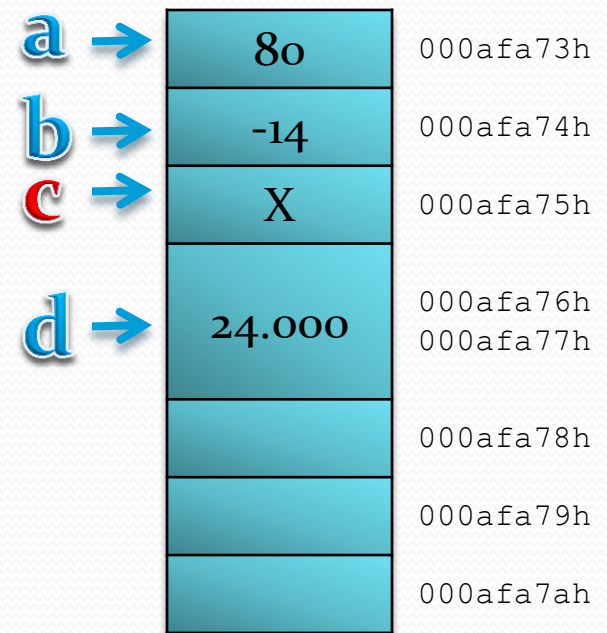
- Örnek;
 - Ad : “a”
 - Boyut : 1 Byte
 - Değer : 80
 - Tip : Pozitif Tamsayı
 - Adresi : 000AFA73h



Bellek

Değişken

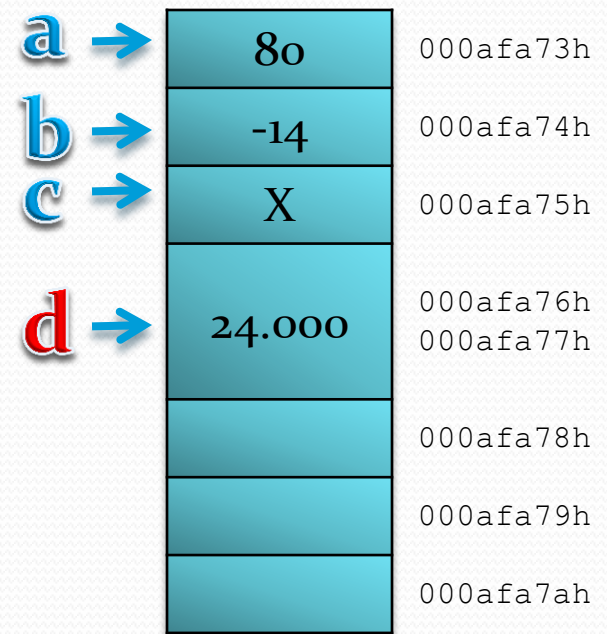
- Örnek;
 - Ad : "c"
 - Boyut : 1 Byte
 - Değer : 'X'
 - Tip : Karakter
 - Adresi : 000AFA75h



Bellek

Değişken

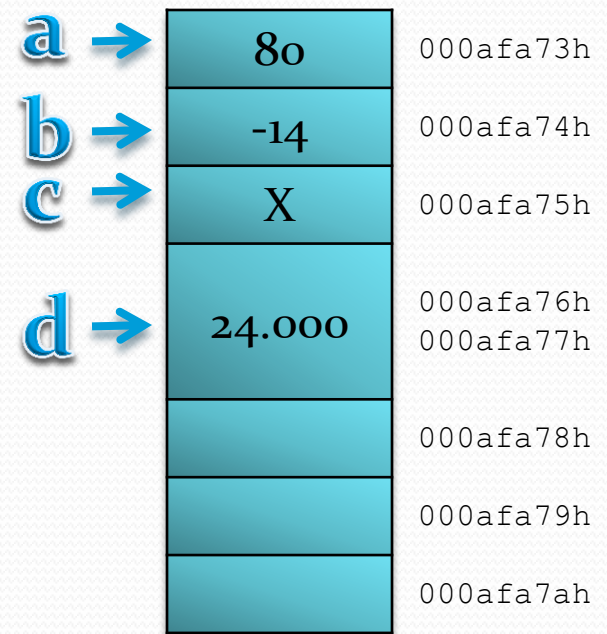
- Örnek;
 - Ad : “d”
 - Boyut : 2 Byte
 - Değer : 24 000
 - Tip : Tamsayı
 - Adres : 000AFA76h-000AFA77h



Bellek

C++'ta Değişkenler

- Buldukları blok «{ ... }» içerisinde
- Kullanılmadan önce
- Genellikler bloğun başında «{» hemen sonra tanımlanırlar.
- Lokal/Yerel değişkenler sadece buldukları blok içerisinde etkilidirler, blok dışında tanımsızdırlar.



Bellek

C++'ta Değişkenler

Syntax / Yazım Kuralı

<veri tipi> <değişken listesi> ;

[<veri tipi> <değişken listesi> ;]

- Tamsayı tipinde bir değişken tanımlamak için:

```
int a;
```

- Karakter tipinde 3 değişken tanımlamak için:

```
char x, y, z;
```

C++'ta Değişkenler

Syntax / Yazım Kuralı

<veri tipi> <değişken adı> = <ilk değer>;

- Tamsayı tipinde bir değişken tanımlayıp ilk değer olarak sıfır atamak için:

```
int a=0;
```

- Karakter Tipinde 3 Değişken Tanımlayıp İlk Değer Olarak Sırasıyla 'a', '9' ve '!' atamak için :

```
char x='a' , y='9' , z='!' ;
```


Değişken İsimleri

- Değişken isimlerinde İngiliz alfabesindeki harfler, rakamlar ve “_” kullanılabilir.
- Değişken adları bir harf ile veya “_” ile başlamak zorundadır.
- Sonrasında harf veya rakam ile devam eder.
- Türkçeye özel Ç,ç,Ğ,ğ,ı,İ,Ö,ö,Ş,ş,Ü,ü harflerini kullanmayınız.
- Örnek
a, deg1, ort, VizeNotu2, Final, _3D

Değişken İsimleri

C++ komutlarını değişken adı olarak kullanmayınız. Aşağıdaki standart C++ komutları yeniden tanımlanamaz. (Reserved words)

<code>asm</code>	<code>else</code>	<code>operator</code>	<code>throw</code>
<code>auto</code>	<code>enum</code>	<code>private</code>	<code>true</code>
<code>bool</code>	<code>extern</code>	<code>protected</code>	<code>try</code>
<code>break</code>	<code>explicit</code>	<code>public</code>	<code>typedef</code>
<code>case</code>	<code>false</code>	<code>register</code>	<code>typeid</code>
<code>catch</code>	<code>float</code>	<code>reinterpret_cast</code>	<code>typename</code>
<code>char</code>	<code>for</code>	<code>return</code>	<code>union</code>
<code>class</code>	<code>friend</code>	<code>short</code>	<code>unsigned</code>
<code>const</code>	<code>goto</code>	<code>signed</code>	<code>using</code>
<code>const_cast</code>	<code>if</code>	<code>sizeof</code>	<code>virtual</code>
<code>continue</code>	<code>inline</code>	<code>static</code>	<code>void</code>
<code>default</code>	<code>int</code>	<code>static_cast</code>	<code>volatile</code>
<code>delete</code>	<code>long</code>	<code>struct</code>	<code>Wchar_t</code>
<code>do</code>	<code>mutable</code>	<code>switch</code>	<code>while</code>
<code>double</code>	<code>namespace</code>	<code>template</code>	
<code>dynamic_cast</code>	<code>new</code>	<code>this</code>	

Değişken İsimleri

- Değişken adları anlamlı olmalı. Kodu inceleyen kişi adından değişkenin niçin kullanıldığını anlamalıdır.
- C++ değişkenleri büyük-küçük harfe duyarlıdır.
Sayi, sayi, sAYI farklı değişkenlerdir.
- C++'ta değişken adları herhangi bir uzunlukta olabilir ama ilk 63 karakteri anlamlıdır.

Değişken İsimleri

- Değişken isimlerinde boşluk kullanılamaz. Boşluk yerine tercihen '_' kullanılabilir. Ama büyük küçük harf kombinasyonu ile bu sorun aşılabılır.
- Deve Notasyonu: ilk kelime dışındaki diğer kelimelerin baş harfleri büyük yazılır.
vizeNotuOrtalamasi, faizOrani, indirimTutari
- Pascal Notasyonu: bütün kelimelerin baş harfleri büyük yazılır.
VizeNotuOrtalamasi, FaizOrani, IndirimTutari

C++ Veritipleri

Tamsayı

- | • Tip | Aralık | Biçim |
|-----------------|-------------------------|------------------|
| • short [int] | -32768..32767 | işaretsiz 16-bit |
| • int | -2147483648..2147483647 | işaretsiz 32-bit |
| • long [int] | -2147483648..2147483647 | işaretsiz 32-bit |
| • unsigned char | 0..255 | işaretsiz 8-bit |
| • signed char | -128..127 | işaretsiz 8-bit |

C++'ta Veritipleri

Gerçel Sayılar

• Tip	Aralık	Bas.	Alan (B)
• float	$\pm 3.4e-38.. \pm 3.4e38$	7-8	4
• double	$\pm 1.7e-308.. \pm 1.7e308$	15-16	8
• long double	$\pm 1.7e-308.. \pm 1.7e308$	15-16	8

C++'ta Veritipleri

Karakter

- Tip

Alan (B)

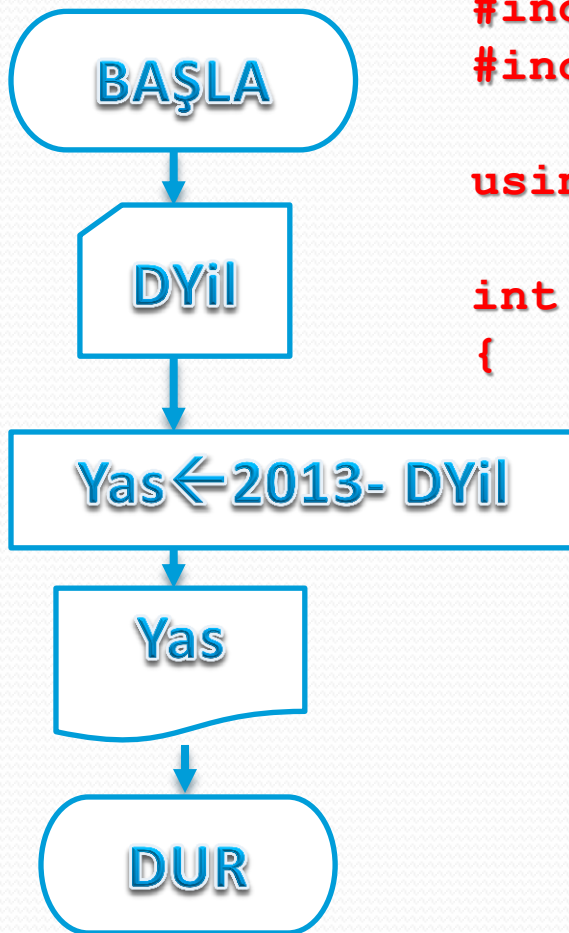
- | | | |
|-----------|-----------------------|-----|
| • char | Bir harf/rakam/işaret | 1 |
| • wchar_t | 1 geniş karakter | 2-4 |

C++'ta Veritipleri

Mantıksal

• Tip	Aralık	Alan (B)
• bool	true veya false	1

Örnek 1: Kullanıcının yaşı



```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int DYil, yas;
    cout << "Dogum yili: ";
    cin >> DYil; //Kullanıcıdan al.

    yas = 2013 - DYil;

    cout << "Yasiniz: " << yas << endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Örnek 2: Karenin alanı

```
#include <cstdlib>
#include <iostream>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
{
```

```
    double kenar, alan;
```

```
    cout << "Karenin kenari: ";
    cin >> kenar;
```

```
    alan = kenar * kenar;
```

```
    cout << "Karenin alanı: " << alan << endl;
```

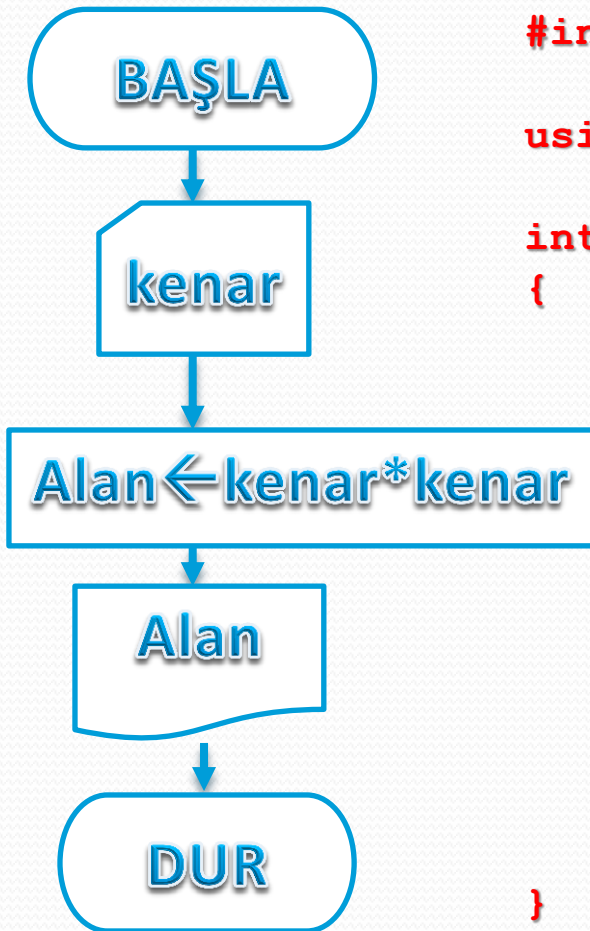
```
    system("PAUSE");
```

```
    return EXIT_SUCCESS;
```

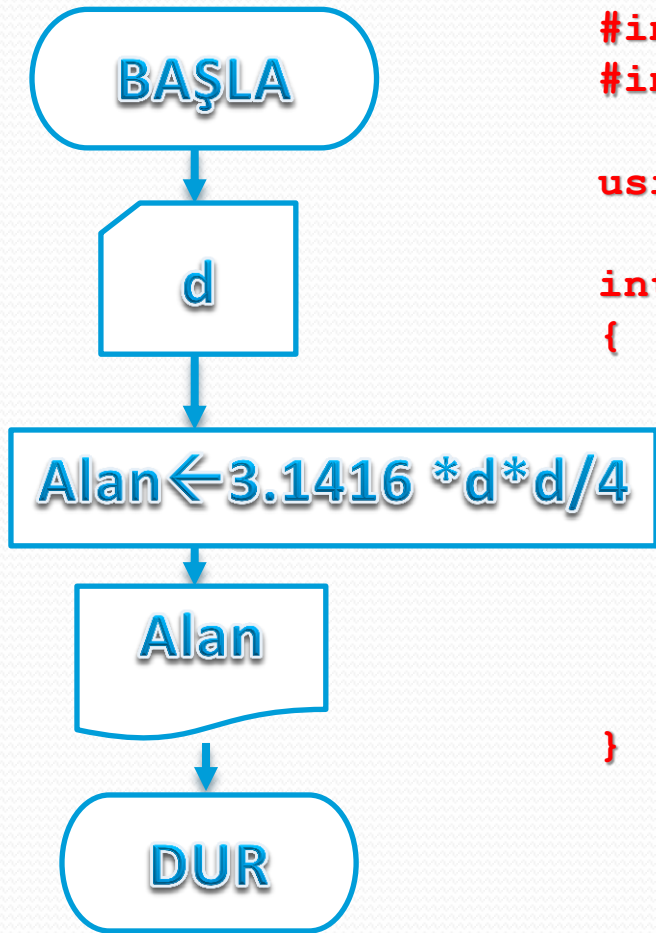
```
}
```

$$A = a^2$$

a: Karenin kenarı



Örnek 3: Dairenin alanı



```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    double d, alan;
    cout << "Dairenin capi: ";
    cin >> d;
    alan = 3.1416 * d * d / 4;
    cout << "Dairenin alanı: " << alan << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

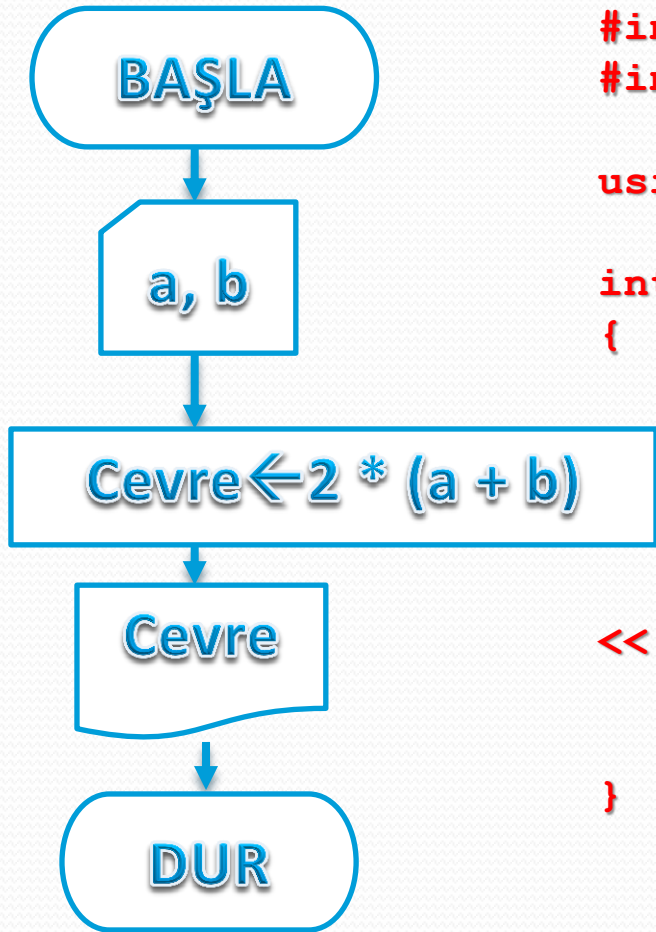
$$A = \frac{\pi d^2}{4}$$

d: dairenin çapı

Ödev

- 1. Kullanıcının gireceği kenar uzunluğundan karenin çevresi hesaplayan programın
 - A) Akış diyagramı
 - B) C++ kodu
- 2. Yarıçapı kullanıcı tarafından girilecek dairenin çevresini hesaplayan programın
 - A) Akış diyagramı
 - B) C++ kodu
- 3. Taban uzunluğu ve bu tabana ait yüksekliği kullanıcı tarafından girilecek olan üçgenin alanını hesaplayan programın
 - A) Akış diyagramı
 - B) C++ kodu

Ödev 1: Dikdörtgenin çevresi



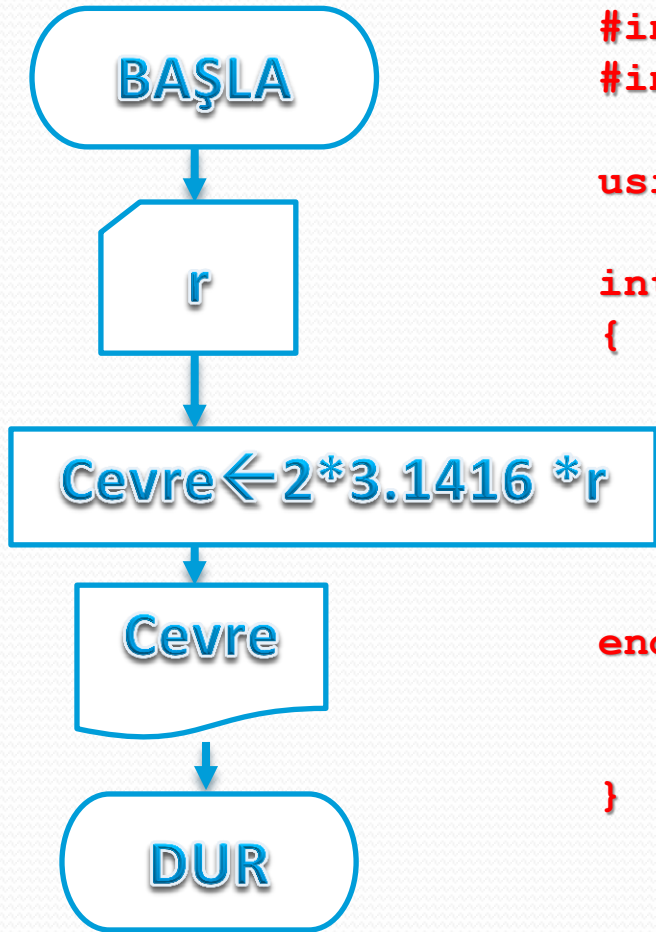
```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    double a, b, cevre;
    cout << «Dikdortgenin kenarlari: »;
    cin >> a >> b;
    cevre = 2 * (a + b);
    cout << «Dikdortgenin cevresi: » << cevre
<< endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

$$Ç = 2(a+b)$$

Ödev 2: Dairenin çevresi



```
#include <cstdlib>
#include <iostream>

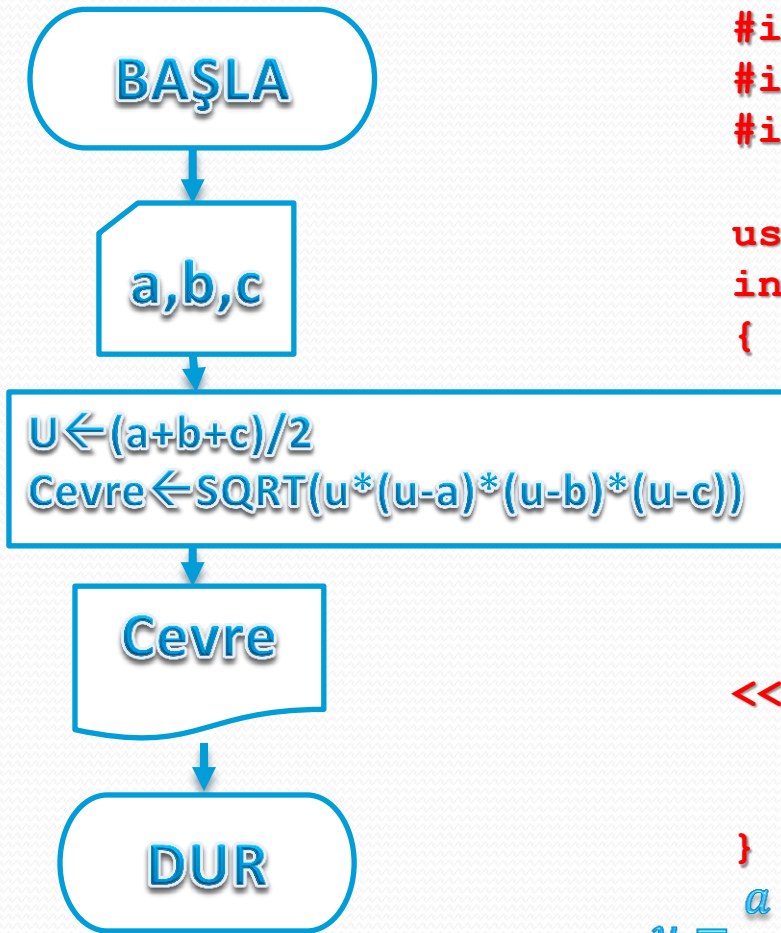
using namespace std;

int main(int argc, char *argv[])
{
    double r, cevre;
    cout << "Dairenin yaricapi: ";
    cin >> r;
    alan = 2 * 3.1416 * r;
    cout << "Dairenin cevresi: " << cevre <<
endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```



r: dairenin yarıçapı

Ödev 3: Üçgenin Alanı



```
#include <cstdlib>
#include <iostream>
#include <cmath>
```

```
using namespace std;
int main(int argc, char *argv[])
{
```

```
double a, b, c, u, cevre;
cout << «Ucgenin kenarlari: »;
cin >> a >> b >> c;
u = (a+b+c)/2;
cevre = sqrt(u*(u-a)*(u-b)*(u-c));
cout << "Ucgenin cevresi: " << cevre
<< endl;
system("PAUSE");
return EXIT_SUCCESS;
```

$$u = \frac{a+b+c}{2} \quad \zeta = \sqrt{u(u-a)(u-b)(u-c)}$$