

# Programlamaya Giriş

## C++ Programlama Dili



# Konular

- ✓ C++ Ortamı Bileşenleri
- ✓ C++ Programının Gelişim Aşamaları
- ✓ C++ Programının Derlenme Aşamaları
- ✓ Büyük Resim - C++ Genel Program Yapısı
- ✓ İlk C++ Programı
- ✓ İsim Uzayı (Namespace)
- ✓ Değişkenler (Variables)
- ✓ Değişmez Tanımlama
- ✓ Setw (Manipulator)
- ✓ Tip Dönüşümü
- ✓ Aritmetik Atama, Artırma, Azaltma İşleçleri
- ✓ İlişkisel İşleçler
- ✓ İşlem Önceliği
- ✓ Kitaplık Fonksiyonları
- ✓ Sorular

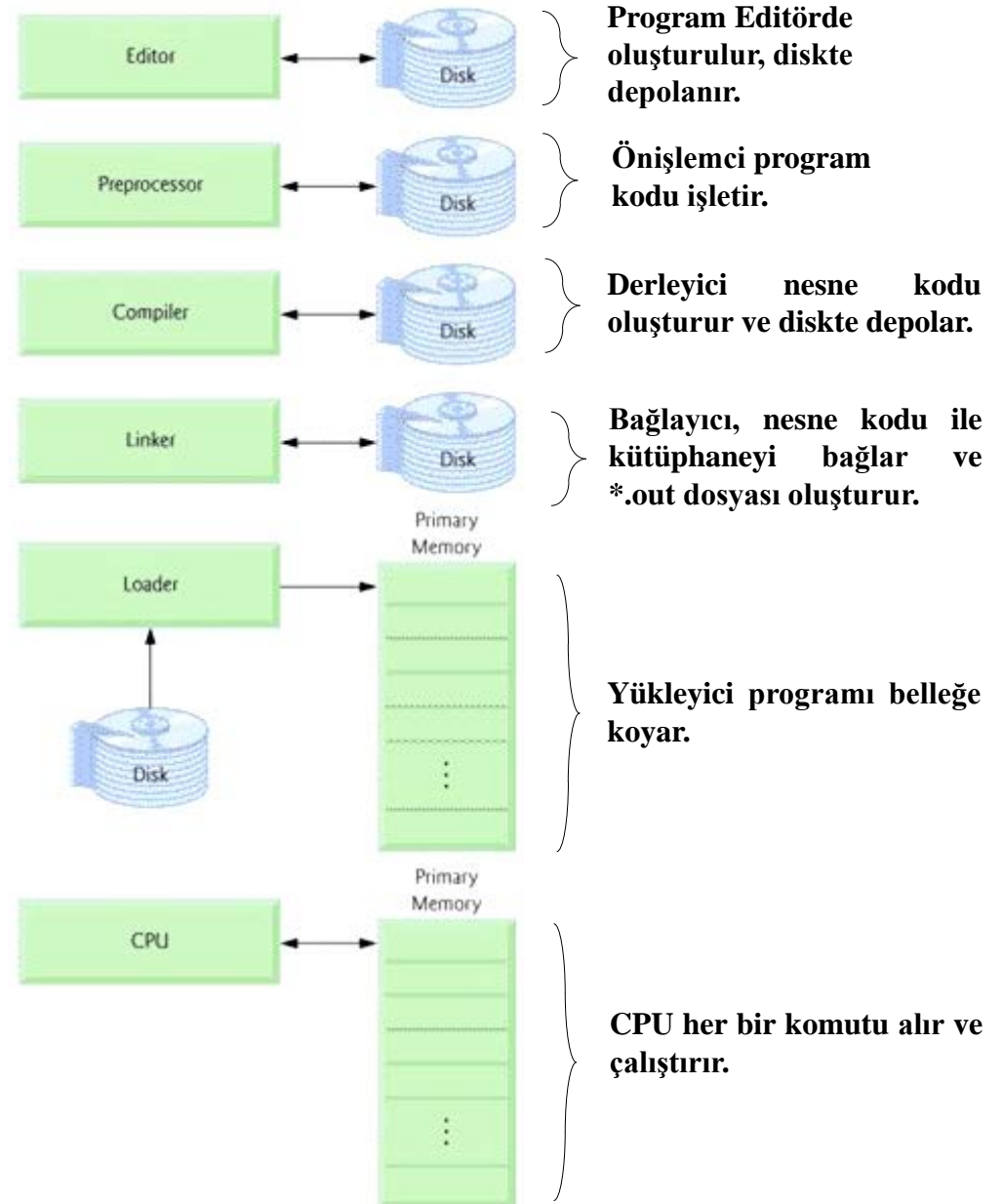
# C++ Ortamı Bileşenleri

- ✓ Program Geliştirme Ortamı
- ✓ C++ Dili
- ✓ C++ Standart Kitaplığı

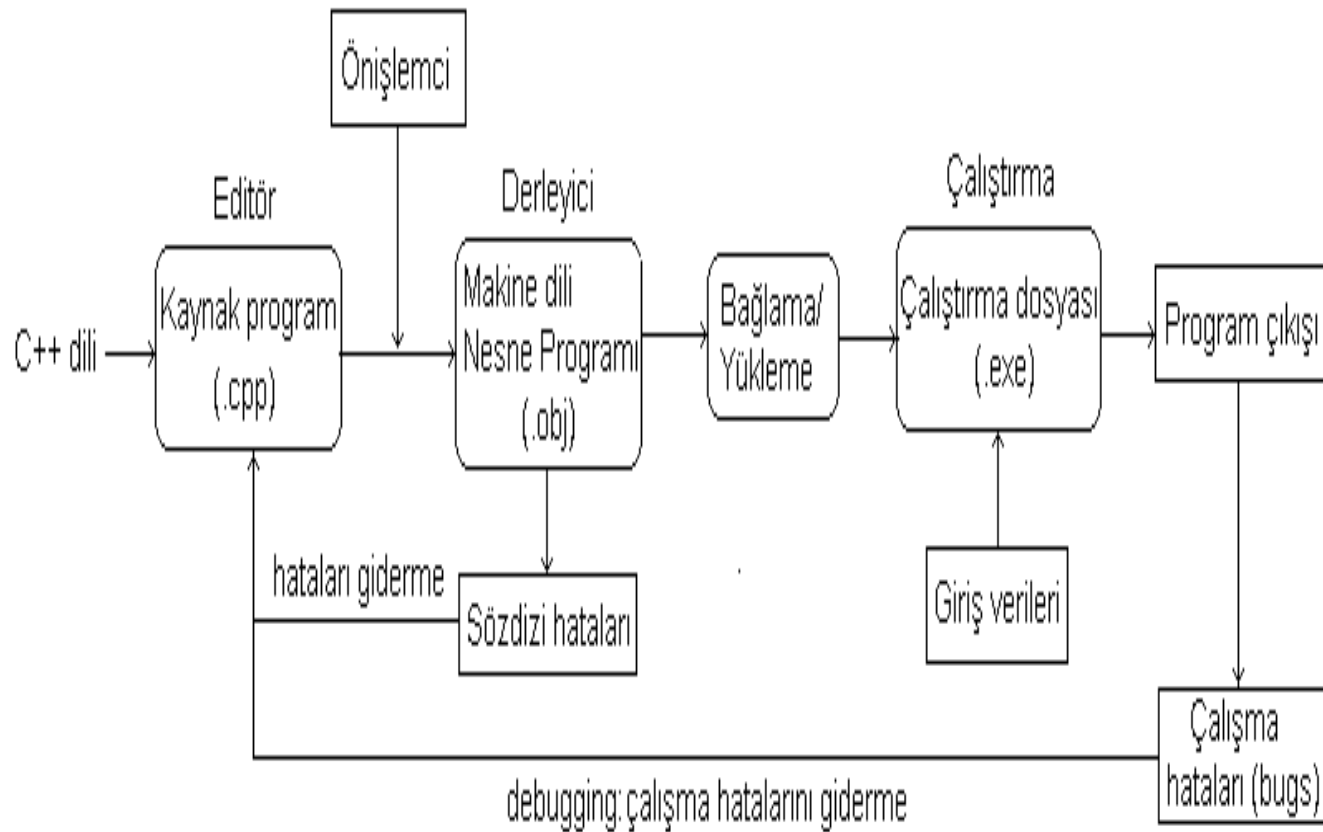
# C++ Programının Gelişim Aşamaları

## C++ Programının Gelişim Aşamaları

1. Edit
2. Preprocess
3. Compile
4. Link
5. Load
6. Execute



# C++ Bir Programının Derlenme Aşamaları



- ✓ Program ilk olarak bir **editörde** yazılır. Bu programa kaynak (source) program ve içindeki kodlara kaynak (source) kodlar denir. kaynak program, sabit diske dosya şeklinde kaydedilmelidir. Bir çok C++ derleyicisi kaynak programın .cpp (.c) uzantılı bir dosya olmasını bekler. Örneğin: ornek1.cpp
- ✓ Derlendikten sonra programın makine dilindeki versiyonu **nesne** (object) program adını taşır ve **.obj** uzantılı olur.
- ✓ Derlendikten sonra çalışmaya hazır programın uzantısı **.exe** olur ve buna exe programı denir.  
{execute: yürütme}

# Büyük Resim - C++ Genel Program Yapısı

```
// C++ genel program yapısı
```

```
#include <iostream>
```

```
#include <math.h>
```

kütüphane dosyalarının yazılması

```
using namespace std;
```

İsim uzayı

```
#define sabit 100
```

```
#define pi 3.14
```

Sabit tanımlanması

```
char kar;
```

```
char dizi[10];
```

```
int sayi=5;
```

Global değişkenlerin tanımlanması

```
void alt_program()
```

```
{
```

```
...
```

```
}
```

Alt programların yazılması

```
main()
```

```
{
```

```
int x;
```

```
float y;
```

Yerel değişkenler

Ana programın yazılması

**Komutlar**

```
.....
```

```
}
```

# İlk C++ Programı

```
//=====
// Name      : AlgoritmaveProgramlama211.cpp
// Author     : Celal Çeken
// Version    :
// Copyright   : Your copyright notice
// Description : Ekrana "Merhaba Dünya" yazan program
//=====
```

```
#include <iostream>
using namespace std;
```

Kütüphane ekleme

```
int main()
{
    cout << "!!!Merhaba Dünya!!!"<< endl; // Merhaba Dünya
    return 0;
}
```

main ve diğer tüm fonksiyon blokları { ile başlar ve } biter

Return 0 programın başarı ile sonlandırıldığını gösterir

- ✓ Anlaşılabilirliği artırmak için programların ilk satırlarında programın kısa tanıtımı, yazarı, tarihi gibi bilgiler yer almalıdır.
- ✓ Boşluklar, boş satırlar ve hiyerarşik hizalama yine anlaşılabilirliği artırma açısından son derece önemlidir.

## İlk Programın Çalıştırılması

- ✓ Derleyici (GNU C Compiler (windows sistemler için mingw))
- ✓ Geliştirme ortamı (NotPad ++, Eclipse CDT, DevC++)
- ✓ makefile dosyasının oluşturulması



## İsim Uzayı (Namespace)

- ✓ Program boyutları büyüdükçe tüm verilere farklı isimler bulmak zorlaşır.
- ✓ Nesne ya da fonksiyon arşivlerinin kullanımında, arşivlerdeki değişkenler ile kendi yazdığınız programınız içerisindeki değişkenlerin isimleri çakışabilir. Bu durumda derleme hatası ile karşılaşabilirsiniz.
- ✓ C++ bu problemi **isim uzayı (namespace)** ile çözmüştür.
- ✓ Her programcı kendi tanımlamalarını kendine ait bir isim uzayı içerisinde yapar.
- ✓ Bir isim uzayındaki verinin ismi başka bir uzaydaki ile aynı olsa dahi çakışma olmaz.

```
namespace programci1
{ //programci1'in isim uzayı
    int tamsayi; // programci1'e ait tamsayi
    void fonksiyon(int);
    : // Diğer değişkenler
} // programci1 isim uzayının sonu
```

```
namespace programci2
{ // programci2'nin isim uzayı
    int tamsayi; // programci2'ye ait tamsayi
    : // Diğer değişkenler
} // İsim uzayının sonu
```

# İsim Uzayı (Namespace)

İsim uzaylarındaki tanımlayıcılara erişilmesi:

```
programci1::tamsayi =3; //programci1'in uzayındaki tamsayi  
programci2::tamsayi=-345; //progamci2'nin uzayındaki tamsayi  
programci1::f(6); //progamci1'in uzayındaki f fonksiyonu
```

**using Bildirimi:**

```
using programci1::tamsayi; //İsim uzayındaki bir değişken için geçerlidir  
tamsayi = 3; // programci1'deki tamsayi  
programci2::tamsayi = -345; //Diğer tamsayi için isim uzayını belirtmek gerekli  
programci1::f(6); // f fonksiyonu için isim uzayını belirtmek gerekli
```

Bu bildirim istenirse tek bir veri yerine isim uzayının tamamı için geçerli yapılabilir. Bunun için using namespace sözcükleri birlikte kullanılır.

```
using namespace programci1; // İsim uzayının tamamı için geçerli  
tamsayi = 3; // programci1'deki tamsayi  
f(6); // programci1'deki f  
programci2::tamsayi = -345; // Diğer isim uzayı için isim belirtmek gerekli
```

# Değişkenler (Variables)

- ✓ Üzerinde işlem yapılan verinin saklandığı bellek konumunu gösterir. Aşağıdaki özelliklere sahiptir.

## Ad (name)

harf, rakam ve \_  
rakam ile başlayamaz  
büyük/küçük harf duyarlı

## Tip (type)

## Değer (value)

## Ömür (lifetime)

Değişkenin kullanılabildiği süre

## İsim Uzayı (namespace)

Değişkene ismi ile erişilebilen program bloğu

# Değişkenler

## Değişken isimlendirme kuralları

- ✓ Değişken isimleri harf, rakam ve '\_' ifadelerinden oluşur.
- ✓ İlk karakter rakam olamaz. Türkçe karakterler kullanılamaz.
- ✓ Ayrılmış kelimeler (reserved words) kullanılamaz. If, else, for v.s.
- ✓ Büyük harf-küçük harf duyarlılığı vardır (case-sensitive).
- ✓ Değişken isminin içerdiği veri ile ilgili olması büyük kolaylıklar sağlar.

ogrenciSayisi

- ✓ Değişkenlerin ilk harfi küçük, diğer kelimelerin baş harfi büyük olmalı.

ogrencininNotOrtalamasi gibi...

## Kodlama türleri

Linux Coding Style, Linus Torvalds

Hungarian Notation,

MSDN GNU Coding Standards

Java Coding Style Guide

## Değişken Tipleri ve Değer Aralıkları

Veri Tipi	Bit	Bayt	Değer Aralığı
Bit	1		0 to 1
Bool		1	true or false
Signed char	8	1	-126 to +127
Unsigned char	8	1	0 to 256
Enum	16	2	-32768 to +32767
Signed short	16	2	-32768 to +32767
Unsigned short	16	2	0 to 65535
Signed int	16	2	-32768 to +32767
Unsigned int	16	2	0 to 65535
Signed long	32	4	-2147483648 to +2147483647
Unsigned long	32	4	0 to 4294967295
Float	32	4	+/-1.7549E+38 to +/-3.402823E+38

int\_8, \_int16, \_int32, \_int32, \_int64

int tipi short tipinden büyük olmasına karşın daha hızlı erişilir.

# Integer

Tamsayılar:

Bayt miktarı

int ( unsigned int ) Derleyici Bağımlı

short ( unsigned short ) 2-Bayt

```
// intvars.cpp
// demonstrates integer variables
#include <iostream>
#include <conio.h>
using namespace std;

int main()
{
    int var1;           //var1 tam sayı değişkeni
    int var2;           //var2 tam sayı değişkeni

    var1 = 20;          //assign value to var1
    var2 = var1 + 10;    //assign value to var2
    cout << "var1+10 un degeri"<<endl<<var2; //output text
    // cout << var2 << endl; //output value of var2
    char ch=getch();
    return 0;
}
```

- ✓ int veri tipi, kullanılan işletim sisteminin kelime uzunluğuna bağlıdır. Örneğin 32 bit kelime uzunluğuna sahip işletim sisteminde 4 bayt.

İmleçin bir sonraki satıra geçmesini sağlar. Std içerisinde tanımlanmıştır. std::endl

# Char

```
// charvars.cpp
// demonstrates character variables
#include <iostream>      //for cout, etc.
#include <conio.h>
using namespace std;

int main()
{
    char charvar1 = 'A'; //define char variable as character
    char charvar2 = '\t'; //define char variable as tab
    cout << charvar1;    //display character
    cout << charvar2;    //display character
    charvar1 = 'B';      //set char variable to char constant
    cout << charvar1 ;   //display character
    cout << '\n';        //display newline character
    char ch=getch();
    return 0;
}
```

```
cout << "Öğretmen\"dersimiz matematik \" dedi
```

\a :bip  
\b :geri silme  
\f :ileri sarma (yazıcı)  
\n :Sonraki (bir alt) satır başına git  
\r :Geçerli satır başına git  
\t :Yatay sekme (tab)  
\v :Dikey sekme  
\ \ :Ters bölü karakteri  
\? :Soru işareti karakteri  
\' :Tek tırnak karakteri  
\" : Çift tırnak karakteri

\n yerine endl komutu da kullanılır



```
// fahrenheit.cpp
// demonstrates cin, newline
#include <iostream>
using namespace std;

int main()
{
    int ftemp; //for temperature in fahrenheit

    cout << "Enter temperature in fahrenheit: ";
    cin >> ftemp;
    int ctemp = (ftemp-32) * 5 / 9;
    cout << "Equivalent in Celsius is: " << ctemp << '\n';
    return 0;
}
```

# Float

```
// circarea.cpp
// demonstrates floating point variables
#include <iostream>           //for cout, etc.
using namespace std;

int main()
{
    float rad;                //variable of type float
    const float PI = 3.14159F; //type const float

    cout << "Dairenin yari capi: "; //prompt
    cin >> rad;                      //get radius
    float alan = PI * rad * rad;     //find area
    cout << "alan degeri: " << alan << endl; //display answer
    return 0;
}
```

Kayan nokta sayılar üstel de yazılabilir: 1.55E12

Virgülden sonra istenen sayıda basamak yazdırma  
(örneğin 2 basamak)  
float PI= 3.1415  
**cout.precision(2);**  
cout<<PI;

## Değişmez (sabit) Tanımlamaları

- ✓ Tüm program boyunca aynı kalan ve değiştirilmesi mümkün olmayan değişken veya tanımlamalardır.
- ✓ `#define pi 3.14`
- ✓ `#define OCAK 1`
- ✓ `#define uyarı "İkaz var"`
- ✓ `const float g=9.81;`
- ✓ `const int x=5;`

## Setw Sabiti (Manipulator)

- ✓ **setw (değer)** // belirtilen değer kadar ekranda yer açar. Tablo şeklinde görüntüler oluşturmak için kullanılır

### setw neden ihtiyaç duyulur

```
#include <iostream>
using namespace std;
int main()
{
    long A=1234567, B=123, C=2345;
    cout << "A" << A << endl
         << "B" << B << endl
         << "C" << C << endl;
    system("pause");
    return 0;
}
```

```
A1234567
B123
C2345
Devam etmek için bir tuşa basın . . .
```

```
#include <iostream>
#include <iomanip> // for setw
using namespace std;
int main()
{
    long A=1234567, B=123, C=2345;
    cout << setw(4)<<"A"<<setw(10)<< A << endl
         << setw(4)<<"B"<<setw(10)<< B << endl
         << setw(4)<<"C"<<setw(10)<< C << endl;
    system("pause");
    return 0;
}
```

```
A    1234567
B      123
C      2345
Devam etmek için bir tuşa basın . . .
```

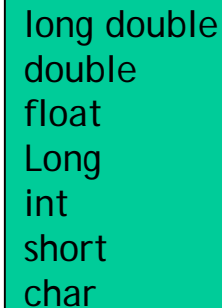
## Tip Dönüşümü (Otomatik)

```
// mixed.cpp
// shows mixed expressions
#include <iostream>
using namespace std;

int main()
{
    int count = 7;
    float avgWeight = 155.5F;

    double totalWeight = count * avgWeight;
    cout << "totalWeight=" << totalWeight << endl;
    return 0;
}
```

### Tiplerin Dönüşüm Sırası



```
long double
double
float
Long
int
short
char
```

## Tip Dönüşümü (Otomatik)

long double

double

float

unsigned long int (synonymous with unsigned long)

long int (synonymous with long)

unsigned int (synonymous with unsigned)

int

unsigned short int (synonymous with unsigned short)

short int (synonymous with short)

unsigned char

char

bool

## Tip Dönüşümü (Açık)

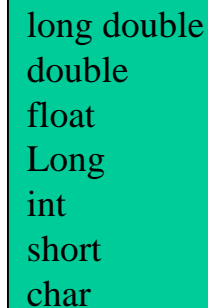
```
#include <iostream>
#include <conio.h>
using namespace std;

int main()
{
    char ch1=getch();

    cout << ch1 <<"nin ASCII degeri="<<static_cast<int>(ch1)<<
endl;
    char ch=getch();
    return 0;
}
```

The cast operator can be used to convert between fundamental numeric types, such as int and double, and between related class types (as we discuss in Chapter 13, Object-Oriented Programming: Polymorphism). Casting to the wrong type may cause compilation errors or runtime errors

### Tiplerin Dönüşüm Sırası



long  
double  
double  
float  
Long  
int  
short  
char

## Aritmetik İşleçler (operators)

C++ İşlemi	Aritmetik İşleç	Mat. Gösterim	C++
Toplama	+	$a+b$	$a+b$
Çıkarma	-	$a-b$	$a-b$
Çarpma	*	$ab$	$a*b$
Bölme	/	$a/b$ veya $\frac{a}{b}$	$a/b$
Mod	%	$a \bmod b$	$a \% b$



## Aritmetik Özel İşleçler

İşleç	C++ Kullanımı	Anlamı
<code>+=</code>	<code>a+=3;</code>	<code>a=a+3;</code>
<code>-=</code>	<code>a-=3;</code>	<code>a=a-3;</code>
<code>*=</code>	<code>a*=3;</code>	<code>a=a*3;</code>
<code>/=</code>	<code>a/=3;</code>	<code>a=a/3;</code>
<code>++</code>	<code>a++;</code> veya <code>++a;</code>	<code>a=a+1;</code>
<code>--</code>	<code>a--;</code> veya <code>--a;</code>	<code>a=a-1;</code>

## Aritmetik Atama İşleçleri - Örnek

```
// assign.cpp
// demonstrates arithmetic assignment operators
#include <iostream>
using namespace std;

int main()
{
    int ans = 27;

    ans += 10;           //same as: ans = ans + 10;
    cout << ans << ", ";
    ans -= 7;           //same as: ans = ans - 7;
    cout << ans << ", ";
    ans *= 2;           //same as: ans = ans * 2;
    cout << ans << ", ";
    ans /= 3;           //same as: ans = ans / 3;
    cout << ans << ", ";
    ans %= 3;           //same as: ans = ans % 3;
    cout << ans << endl;
    return 0;
}
```

*Programmers can write programs a bit faster and compilers can compile programs a bit faster when the abbreviated assignment operators are used. Some compilers generate code that runs faster when abbreviated assignment operators are used.*

## Artırma Azaltma İşleçleri - Örnek

```
// increm.cpp
// demonstrates the increment operator
#include <iostream>
using namespace std;

int main()
{
    int count = 10;

    cout << "count=" << count << endl;    //displays 10
    cout << "count=" << ++count << endl;    //displays 11 (prefix)
    cout << "count=" << count << endl;    //displays 11
    cout << "count=" << count++ << endl;    //displays 11 (postfix)
    cout << "count=" << count << endl;    //displays 12
    return 0;
}
```

## İlişkisel İşleçler

İşleç	Kullanımı	Anlamı
==	$a==b$	a, b'ye eşit mi?
!=	$a!=b$	a, b'den farklı mı?
>	$a>b$	a, b'den büyük mü?
<	$a<b$	a, b'den küçük mü?
>=	$a>=b$	a, b'den büyük eşit mi?
<=	$a<=b$	a, b'den küçük eşit mi?

## Mantıksal ve Bitsel İşleçler

İşleç	Kullanımı	Anlamı	Açıklama
&&	a&&b	a ve b	Sonuç DOĞRU(1) veya YANLIŞ (0) tır
	a  b	a veya b	Sonuç DOĞRU(1) veya YANLIŞ (0) tır
!	!a	a DEĞİL	a'nın mantıksal değili alınır
&	a&b	a AND b	Her iki değer bitsel VE yapılır
	a b	a OR b	Her iki değer bitsel VEYA yapılır
~	~a	NOT a	a'nın tüm bitleri terslenir
^	a^b	a XOR b	Her iki değer bitsel XOR yapılır

# İlişkisel İşleçler

```
// relat.cpp
// demonstrates relational operators
#include <iostream>
using namespace std;

int main()
{
    int numb;

    cout << "Enter a number: ";
    cin >> numb;
    cout << "numb<10 is " << (numb < 10) << endl;
    cout << "numb>10 is " << (numb > 10) << endl;
    cout << "numb==10 is " << (numb == 10) << endl;
    return 0;
}
```

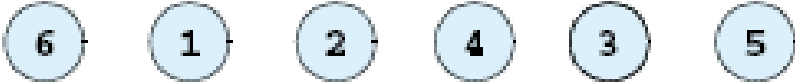
## İşleç Öncelikleri

İşleçler	Yön	İşleç Grubu
()	soldan sağa	parantez
++ --	sağdan sola	tekli
* / %	soldan sağa	çoklu çarpma
+ -	soldan sağa	çoklu toplama
< <= > >=	soldan sağa	ilişkisel
== !=	soldan sağa	eşitlik
&&	soldan sağa	ilişkisel
	soldan sağa	ilişkisel
= += -= *= /= %=	sağdan sola	atama

# İşlem Önceliği

Algebra:  $z = pr\%q + w/x - y$

Java: `z = p * r % q + w / x - y;`



Using redundant parentheses in complex arithmetic expressions can make the expressions clearer.



## Kitaplık Fonksiyonları

```
// sqrt.cpp
// demonstrates sqrt() library function
#include <iostream>           //for cout, etc.
#include <cmath>              //for sqrt()
using namespace std;

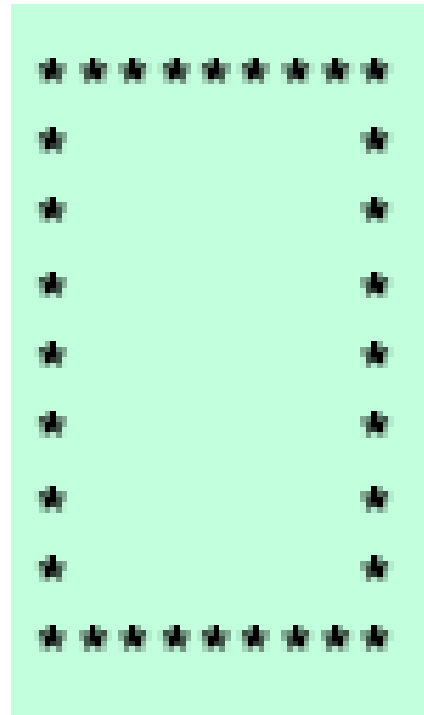
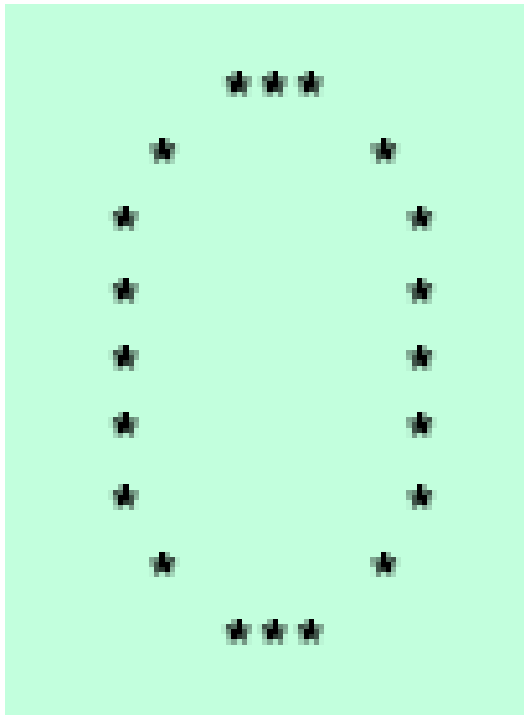
int main()
{
    double number, answer;    //sqrt() requires type double

    cout << "Enter a number: ";
    cin >> number;            //get the number
    answer = sqrt(number);    //find square root
    cout << "Square root is "
    << answer << endl;        //display it
    return 0;
}
```

Method	Description	Example
<b>ceil( x )</b>	rounds $x$ to the smallest integer not less than $x$	ceil( 9.2 ) is 10.0 ceil( -9.8 ) is -9.0
<b>cos( x )</b>	trigonometric cosine of $x$ ( $x$ in radians)	cos( 0.0 ) is 1.0
<b>exp( x )</b>	exponential function $e^x$	exp( 1.0 ) is 2.71828 exp( 2.0 ) is 7.38906
<b>fabs( x )</b>	absolute value of $x$	fabs( 5.1 ) is 5.1 fabs( 0.0 ) is 0.0 fabs( -8.76 ) is 8.76
<b>floor( x )</b>	rounds $x$ to the largest integer not greater than $x$	floor( 9.2 ) is 9.0 floor( -9.8 ) is -10.0
<b>fmod( x, y )</b>	remainder of $x/y$ as a floating-point number	fmod( 13.657, 2.333 ) is 1.251
<b>log( x )</b>	natural logarithm of $x$ (base $e$ )	log( 2.718282 ) is 1.0 log( 7.389056 ) is 2.0
<b>log10( x )</b>	logarithm of $x$ (base 10)	log10( 10.0 ) is 1.0 log10( 100.0 ) is 2.0
<b>pow( x, y )</b>	$x$ raised to power $y$ ( $x^y$ )	pow( 2, 7 ) is 128 pow( 9, .5 ) is 3
<b>sin( x )</b>	trigonometric sine of $x$ ( $x$ in radians)	sin( 0.0 ) is 0
<b>sqrt( x )</b>	square root of $x$	sqrt( 900.0 ) is 30.0 sqrt( 9.0 ) is 3.0
<b>tan( x )</b>	trigonometric tangent of $x$ ( $x$ in radians)	tan( 0.0 ) is 0

## Çalışma Soruları

1. C ve C++ dillerinin tarihini araştırınız.
2. Kullanıcıdan istediği iki tamsayıyı okuyan ve bu değerlerin toplamını, farkını, ortalamasını, çarpımını, bölümünü ve bölümünden kalanını bulup ekrana yazdıran programı yazınız.
3. Aşağıdaki çıktıları alt alta üreten programı yazınız.



- 2.13** Given that  $y = ax^3 + 7$ , which of the following are correct Java statements for this equation?
- `y = a * x * x * x + 7;`
  - `y = a * x * x * ( x + 7 );`
  - `y = ( a * x ) * x * ( x + 7 );`
  - `y = ( a * x ) * x * x + 7;`
  - `y = a * ( x * x * x ) + 7;`
  - `y = a * x * ( x * x + 7 );`
- 2.14** State the order of evaluation of the operators in each of the following Java statements, and show the value of `x` after each statement is performed:
- `x = 7 + 3 * 6 / 2 - 1;`
  - `x = 2 % 2 + 2 * 2 - 2 / 2;`
  - `x = ( 3 * 9 * ( 3 + ( 9 * 3 / ( 3 ) ) ) );`
- 2.15** Write an application that displays the numbers 1 to 4 on the same line, with each pair of adjacent numbers separated by one space. Write the program using the following methods:
- Using one `System.out` statement.
  - Using four `System.out` statements.
- 2.16** Write an application that asks the user to enter two numbers, obtains the numbers from the user and prints the sum, product, difference and quotient (division) of the numbers. Use the techniques shown in Fig. 2.9.
- 2.17** Write an application that asks the user to enter two integers, obtains the numbers from the user and displays the larger number followed by the words “**is larger**” in an information message dialog. If the numbers are equal, print the message “**These numbers are equal.**” Use the techniques shown in Fig. 2.20.
- 2.18** Write an application that inputs three integers from the user and displays the sum, average, product, smallest and largest of the numbers in an information message dialog. Use the GUI techniques shown in Fig. 2.20. [Note: The calculation of the average in this exercise should result in an integer representation of the average. So, if the sum of the values is 7, the average should be 2, not 2.3333....]
- 2.19** Write an application that inputs from the user the radius of a circle and prints the circle’s diameter, circumference and area. Use the value 3.14159 for  $\pi$ . Use the GUI techniques shown in Fig. 2.9. [Note: You may also use the predefined constant `Math.PI` for the value of  $\pi$ . This constant is more precise than the value 3.14159. Class `Math` is defined in the `java.lang` package, so you do not need to `import` it.] Use the following formulas ( $r$  is the radius):

$$\begin{aligned} \text{diameter} &= 2r \\ \text{circumference} &= 2\pi r \\ \text{area} &= \pi r^2 \end{aligned}$$

# Kaynaklar

- ✓ Horstmann, C., Budd, T., Big C++, John Wiley & Sons, Inc.
- ✓ Deitel, C++ How To Program, Prentice Hall
- ✓ Robert Lafore, Object Oriented Programming in C++, Macmillan Computer Publishing