

BİL-142 Bilgisayar Programlama II (C/C++)

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Konular

- Giriş
- Kontrol Yapıları
- if Seçme Deyimi
- if ... else Seçme Deyimi
- while Tekrar Deyimi
- Sayaç Kontrollü Tekrar
- Sentinel Kontrollü Tekrar
- İççe Kontrol Deyimleri
- for Tekrar Deyimi
- do ... while Tekrar Deyimi
- switch Çoklu Seçim Deyimi
- break ve continue Deyimleri

Giriş

- Bir problem çözümü için program yazarken hangi yapıların kullanılacağıın planlanması gerekir.
- Hangi kontrol yapılarının kullanılacağı ve hangi sırada kullanılacağı program geliştirirken belirlenmelidir.
- Program geliştirirken önce algoritma oluşturulması, sonra kabakod (pseudo code) olarak yazılması ve daha sonra uygun yapıların kullanılarak geliştirilmesi doğrudur.

3

Konular

- Giriş
- **Kontrol Yapıları**
- if Seçme Deyimi
- if ... else Seçme Deyimi
- while Tekrar Deyimi
- Sayaç Kontrollü Tekrar
- Sentinel Kontrollü Tekrar
- İççe Kontrol Deyimleri
- for Tekrar Deyimi
- do ... while Tekrar Deyimi
- switch Çoklu Seçim Deyimi
- break ve continue Deyimleri

Kontrol Yapıları

- Programlarda deyimler sıralı olarak çalışır (sequential execution).
- Programlardaki bazı komutlar çalışma sırasını değiştir (transfer of control).
- C++ programlama dili seçme deyimlerine (if, if ...else, switch) ve tekrar deyimlerine (while, for, do ... while) sahiptir.

5

Konular

- Giriş
- Kontrol Yapıları
- **if Seçme Deyimi**
- if ... else Seçme Deyimi
- while Tekrar Deyimi
- Sayaç Kontrollü Tekrar
- Sentinel Kontrollü Tekrar
- İççe Kontrol Deyimleri
- for Tekrar Deyimi
- do ... while Tekrar Deyimi
- switch Çoklu Seçim Deyimi
- break ve continue Deyimleri

if Seçme Deyimi

- if seçme verilen şartın doğruluğuna bağlı olarak belirlenen bloğun çalışmasını sağlar.

```
if ( grade >= 60 )  
    cout << "Passed";
```

- Çalıştırmak istenilen birden fazla deyim varsa block içinde { ... } ifade edilebilir.

7

Konular

- Giriş
- Kontrol Yapıları
- if Seçme Deyimi
- if ... else Seçme Deyimi
- while Tekrar Deyimi
- Sayaç Kontrollü Tekrar
- Sentinel Kontrollü Tekrar
- İççe Kontrol Deyimleri
- for Tekrar Deyimi
- do ... while Tekrar Deyimi
- switch Çoklu Seçim Deyimi
- break ve continue Deyimleri

if ... else Seçme Deyimi

- if ... else seçme deyimi verilen şartın doğruluğuna ve yanlışlığına bağlı olarak belirlenen iki ayrı bloğun çalışmasını sağlar.

```
if ( grade >= 60 )
    cout << "Passed";
else
    cout << "Failed";
```

- Çalıştırmak istenilen birden fazla deyim varsa, her iki durum içinde block içinde { } ifade edilebilir.
- Aşağıdaki yazım şeklide aynı işi yapar.

```
cout << ( grade >= 60 ? "Passed" : "Failed" );
```

9

if ... else Seçme Deyimi

İççe if ... else deyimleri

- Birden fazla if ... else seçme deyimi birlikte kullanılabilir.

```
if ( x > 5 )
    if ( y > 5 )
        cout << "x and y are > 5";
else
    cout << "x is <= 5";
```

- else ifadesi $y \leq 5$ ise çalışır ve $x > 5$ olsa bile "x is ≤ 5 " mesajı yazılır.
- Bu şekilde kullanımlarda parantez { } kullanılabilir.

```
if ( x > 5 )
{
    if ( y > 5 )
        cout << "x and y are > 5";
}
else
    cout << "x is <= 5";
```

10

Konular

- Giriş
- Kontrol Yapıları
- if Seçme Deyimi
- if ... else Seçme Deyimi
- **while Tekrar Deyimi**
- Sayaç Kontrollü Tekrar
- Sentinel Kontrollü Tekrar
- İççe Kontrol Deyimleri
- for Tekrar Deyimi
- do ... while Tekrar Deyimi
- switch Çoklu Seçim Deyimi
- break ve continue Deyimleri

while Tekrar Deyimi

- while tekrar deyimi verilen şart doğru olduğu sürece belirlenen işlerin tekrar edilmesini sağlar.

```
int product = 3;

while ( product <= 100)
    product = 3 * product;
```

- Burada 100 sayaç olarak adlandırılır ve tekrar işleminin sayısını gösterir.
- Birden fazla deyim için blok { } oluşturulmalıdır.
- Şart kısmı ve (&&), veya(||) mantıksal ifadelerle genişletilebilir.

Konular

- Giriş
- Kontrol Yapıları
- if Seçme Deyimi
- if ... else Seçme Deyimi
- while Tekrar Deyimi
- **Sayaç Kontrollü Tekrar**
- Sentinel Kontrollü Tekrar
- İççe Kontrol Deyimleri
- for Tekrar Deyimi
- do ... while Tekrar Deyimi
- switch Çoklu Seçim Deyimi
- break ve continue Deyimleri

Sayaç Kontrollü Tekrar

- while tekrar deyimi verilen şart doğru olduğu sürece (sayaç değerine ulaşıncaya kadar) belirlenen işlerin tekrar edilmesini sağlar.
- Sayaç kontrollü tekrar işleminde kontrol değişkeninin başlangıç değerinin, döngü devam şartının ve sonuçlanma değerinin verilmesi gerekir.
- Her döngüde kontrol değişkeninin değerinin değiştirilmesi (artırılması, azaltılması) gereklidir.

```
int counter = 1; // declare and initialize control variable

while ( counter <= 10 ) // loop-continuation condition
{
    cout << counter << " ";
    counter++; // increment control variable by 1
} // end while
```

14

Sayaç Kontrollü Tekrar

```
1 // Fig. 4.8: GradeBook.h
2 // Definition of class GradeBook that determines a class average.
3 // Member functions are defined in GradeBook.cpp
4 #include <string> // program uses C++ standard string class
5 using std::string;
6
7 // GradeBook class definition
8 class GradeBook
9 {
10 public:
11     GradeBook( string ); // constructor initializes course name
12     void setCourseName( string ); // function to set the course name
13     string getCourseName(); // function to retrieve the course name
14     void displayMessage(); // display a welcome message
15     void determineClassAverage(); // averages grades entered by the user
16 private:
17     string courseName; // course name for this GradeBook
18 }; // end class GradeBook
```

15

Sayaç Kontrollü Tekrar

```
1 // Fig. 4.9: GradeBook.cpp
2 // Member-function definitions for class GradeBook that solves the
3 // class average program with counter-controlled repetition.
4 #include <iostream>
5 using std::cout;
6 using std::cin;
7 using std::endl;
8
9 #include "GradeBook.h" // include definition of class GradeBook
10
11 // constructor initializes courseName with string supplied as argument
12 GradeBook::GradeBook( string name )
13 {
14     setCourseName( name ); // validate and store courseName
15 } // end GradeBook constructor
16
17 // function to set the course name;
18 // ensures that the course name has at most 25 characters
19 void GradeBook::setCourseName( string name )
20 {
21     if ( name.length() <= 25 ) // if name has 25 or fewer characters
22         courseName = name; // store the course name in the object
23     else // if name is longer than 25 characters
24     { // set courseName to first 25 characters of parameter name
25         courseName = name.substr( 0, 25 ); // select first 25 characters
26         cout << "Name \"" << name << "\" exceeds maximum length (25).\n"
27             << "Limiting courseName to first 25 characters.\n" << endl;
28     } // end if...else
29 } // end function setCourseName
30
```

16

Sayaç Kontrollü Tekrar

```
31 // function to retrieve the course name
32 string GradeBook::getCourseName()
33 {
34     return courseName;
35 } // end function getCourseName
36
37 // display a welcome message to the GradeBook user
38 void GradeBook::displayMessage()
39 {
40     cout << "Welcome to the grade book for\n" << getCourseName() << "\n"
41         << endl;
42 } // end function displayMessage
43
44 // determine class average based on 10 grades entered by user
45 void GradeBook::determineClassAverage()
46 {
47     int total; // sum of grades entered by user
48     int gradeCounter; // number of the grade to be entered next
49     int grade; // grade value entered by user
50     int average; // average of grades
51
52     // initialization phase
53     total = 0; // initialize total
54     gradeCounter = 1; // initialize loop counter
55
56     // processing phase
57     while ( gradeCounter <= 10 ) // loop 10 times
58     {
59         cout << "Enter grade: "; // prompt for input
60         cin >> grade; // input next grade
61         total = total + grade; // add grade to total
62         gradeCounter = gradeCounter + 1; // increment counter by 1
63     } // end while
64
```

17

Sayaç Kontrollü Tekrar

```
65 // termination phase
66 average = total / 10; // integer division yields integer result
67
68 // display total and average of grades
69 cout << "\nTotal of all 10 grades is " << total << endl;
70 cout << "Class average is " << average << endl;
71 } // end function determineClassAverage
```

18

Sayaç Kontrollü Tekrar

```
1 // Fig. 4.10: fig04_10.cpp
2 // Create GradeBook object and invoke its determineClassAverage function.
3 #include "GradeBook.h" // include definition of class GradeBook
4
5 int main()
6 {
7     // create GradeBook object myGradeBook and
8     // pass course name to constructor
9     GradeBook myGradeBook( "CS101 C++ Programming" );
10
11     myGradeBook.displayMessage(); // display welcome message
12     myGradeBook.determineClassAverage(); // find average of 10 grades
13     return 0; // indicate successful termination
14 } // end main
```

```
Welcome to the grade book for
CS101 C++ Programming

Enter grade: 67
Enter grade: 78
Enter grade: 89
Enter grade: 67
Enter grade: 87
Enter grade: 98
Enter grade: 93
Enter grade: 85
Enter grade: 82
Enter grade: 100

Total of all 10 grades is 846
Class average is 84
```

19

Konular

- Giriş
- Kontrol Yapıları
- if Seçme Deyimi
- if ... else Seçme Deyimi
- while Tekrar Deyimi
- Sayaç Kontrollü Tekrar
- **Sentinel Kontrollü Tekrar**
- İççe Kontrol Deyimleri
- for Tekrar Deyimi
- do ... while Tekrar Deyimi
- switch Çoklu Seçim Deyimi
- break ve continue Deyimleri

Sentinel Kontrollü Tekrar

- **while** tekrar deyimi verilen şart doğru olduğu sürece belirlenen işlerin tekrar edilmesini sağlar.

```
1 // Fig. 4.12: GradeBook.h
2 // Definition of class GradeBook that determines a class average.
3 // Member functions are defined in GradeBook.cpp
4 #include <string> // program uses C++ standard string class
5 using std::string;
6
7 // GradeBook class definition
8 class GradeBook
9 {
10 public:
11     GradeBook( string ); // constructor initializes course name
12     void setCourseName( string ); // function to set the course name
13     string getCourseName(); // function to retrieve the course name
14     void displayMessage(); // display a welcome message
15     void determineClassAverage(); // averages grades entered by the user
16 private:
17     string courseName; // course name for this GradeBook
18 }; // end class GradeBook
```

21

Sentinel Kontrollü Tekrar

```
1 // Fig. 4.13: GradeBook.cpp
2 // Member-function definitions for class GradeBook that solves the
3 // class average program with sentinel-controlled repetition.
4 #include <iostream>
5 using std::cout;
6 using std::cin;
7 using std::endl;
8 using std::fixed; // ensures that decimal point is displayed
9
10 #include <iomanip> // parameterized stream manipulators
11 using std::setprecision; // sets numeric output precision
12
13 // include definition of class GradeBook from GradeBook.h
14 #include "GradeBook.h"
15
16 // constructor initializes courseName with string supplied as argument
17 GradeBook::GradeBook( string name )
18 {
19     setCourseName( name ); // validate and store courseName
20 } // end GradeBook constructor
21
22 // function to set the course name;
23 // ensures that the course name has at most 25 characters
24 void GradeBook::setCourseName( string name )
25 {
26     if ( name.length() <= 25 ) // if name has 25 or fewer characters
27         courseName = name; // store the course name in the object
28     else // if name is longer than 25 characters
29     { // set courseName to first 25 characters of parameter name
30         courseName = name.substr( 0, 25 ); // select first 25 characters
31         cout << "Name \"<name>\" exceeds maximum length (25).\n"
32             << "Limiting courseName to first 25 characters.\n" << endl;
33     } // end if...else
34 } // end function setCourseName
35
```

22

Sentinel Kontrollü Tekrar

```
36 // function to retrieve the course name
37 string GradeBook::getCourseName()
38 {
39     return courseName;
40 } // end function getCourseName
41
42 // display a welcome message to the GradeBook user
43 void GradeBook::displayMessage()
44 {
45     cout << "Welcome to the grade book for\n" << getCourseName() << "!\n"
46         << endl;
47 } // end function displayMessage
48
49 // determine class average based on 10 grades entered by user
50 void GradeBook::determineClassAverage()
51 {
52     int total; // sum of grades entered by user
53     int gradeCounter; // number of grades entered
54     int grade; // grade value
55     double average; // number with decimal point for average
56
57     // initialization phase
58     total = 0; // initialize total
59     gradeCounter = 0; // initialize loop counter
60
61     // processing phase
62     // prompt for input and read grade from user
63     cout << "Enter grade or -1 to quit: ";
64     cin >> grade; // input grade or sentinel value
65
```

23

Sentinel Kontrollü Tekrar

```
66 // loop until sentinel value read from user
67 while ( grade != -1 ) // while grade is not -1
68 {
69     total = total + grade; // add grade to total
70     gradeCounter = gradeCounter + 1; // increment counter
71
72     // prompt for input and read next grade from user
73     cout << "Enter grade or -1 to quit: ";
74     cin >> grade; // input grade or sentinel value
75 } // end while
76
77 // termination phase
78 if ( gradeCounter != 0 ) // if user entered at least one grade...
79 {
80     // calculate average of all grades entered
81     average = static_cast< double >( total ) / gradeCounter;
82
83     // display total and average (with two digits of precision)
84     cout << "\nTotal of all " << gradeCounter << " grades entered is "
85         << total << endl;
86     cout << "Class average is " << setprecision( 2 ) << fixed << average
87         << endl;
88 } // end if
89 else // no grades were entered, so output appropriate message
90     cout << "No grades were entered" << endl;
91 } // end function determineClassAverage
```

24

Sentinel Kontrollü Tekrar

```
1 // Fig. 4.14: fig04_14.cpp
2 // Create GradeBook object and invoke its determineClassAverage function.
3
4 // include definition of class GradeBook from GradeBook.h
5 #include "GradeBook.h"
6
7 int main()
8 {
9     // create GradeBook object myGradeBook and
10    // pass course name to constructor
11    GradeBook myGradeBook( "CS101 C++ Programming" );
12
13    myGradeBook.displayMessage(); // display welcome message
14    myGradeBook.determineClassAverage(); // find average of 10 grades
15    return 0; // indicate successful termination
16 } // end main
```

```
Welcome to the grade book for
CS101 C++ Programming

Enter grade or -1 to quit: 97
Enter grade or -1 to quit: 88
Enter grade or -1 to quit: 72
Enter grade or -1 to quit: -1

Total of all 3 grades entered is 257
Class average is 85.67
```

25

Konular

- Giriş
- Kontrol Yapıları
- if Seçme Deyimi
- if ... else Seçme Deyimi
- while Tekrar Deyimi
- Sayaç Kontrollü Tekrar
- Sentinel Kontrollü Tekrar
- **İççe Kontrol Deyimleri**
- for Tekrar Deyimi
- do ... while Tekrar Deyimi
- switch Çoklu Seçim Deyimi
- break ve continue Deyimleri

İççe Kontrol Deyimleri

- Birden fazla kontrol deyimi birlikte ve iççe kullanılabilir.

```
1 // Fig. 4.16: Analysis.h
2 // Definition of class Analysis that analyzes examination results.
3 // Member function is defined in Analysis.cpp
4
5 // Analysis class definition
6 class Analysis
7 {
8 public:
9     void processExamResults(); // process 10 students' examination results
10 }; // end class Analysis
```

27

İççe Kontrol Deyimleri

```
1 // Fig. 4.17: Analysis.cpp
2 // Member-function definitions for class Analysis that
3 // analyzes examination results.
4 #include <iostream>
5 using std::cout;
6 using std::cin;
7 using std::endl;
8
9 // include definition of class Analysis from Analysis.h
10 #include "Analysis.h"
11
12 // process the examination results of 10 students
13 void Analysis::processExamResults()
14 {
15     // initializing variables in declarations
16     int passes = 0; // number of passes
17     int failures = 0; // number of failures
18     int studentCounter = 1; // student counter
19     int result; // one exam result (1 = pass, 2 = fail)
20 }
```

28

İççe Kontrol Deyimleri

```
21 // process 10 students using counter-controlled loop
22 while ( studentCounter <= 10 )
23 {
24     // prompt user for input and obtain value from user
25     cout << "Enter result (1 = pass, 2 = fail): ";
26     cin >> result; // input result
27
28     // if...else nested in while
29     if ( result == 1 ) // if result is 1,
30         passes = passes + 1; // increment passes;
31     else // else result is not 1, so
32         failures = failures + 1; // increment failures
33
34     // increment studentCounter so loop eventually terminates
35     studentCounter = studentCounter + 1;
36 } // end while
37
38 // termination phase; display number of passes and failures
39 cout << "Passed " << passes << "\nFailed " << failures << endl;
40
41 // determine whether more than eight students passed
42 if ( passes > 8 )
43     cout << "Raise tuition " << endl;
44 } // end function processExamResults
```

29

İççe Kontrol Deyimleri

```
1 // Fig. 4.18: fig04_18.cpp
2 // Test program for class Analysis.
3 #include "Analysis.h" // include definition of class Analysis
4
5 int main()
6 {
7     Analysis application; // create Analysis object
8     application.processExamResults(); // call function to process results
9     return 0; // indicate successful termination
10 } // end main
```

```
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 2
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Passed 9
Failed 1
Raise tuition
```

```
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 2
Enter result (1 = pass, 2 = fail): 2
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 2
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 1
Enter result (1 = pass, 2 = fail): 2
Passed 6
Failed 4
```

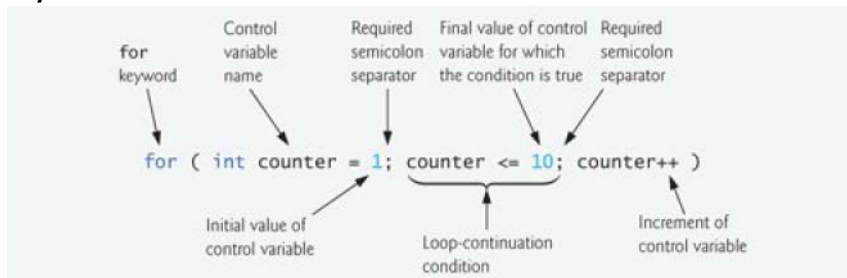
30

Konular

- Giriş
- Kontrol Yapıları
- if Seçme Deyimi
- if ... else Seçme Deyimi
- while Tekrar Deyimi
- Sayaç Kontrollü Tekrar
- Sentinel Kontrollü Tekrar
- İççe Kontrol Deyimleri
- **for Tekrar Deyimi**
- do ... while Tekrar Deyimi
- switch Çoklu Seçim Deyimi
- break ve continue Deyimleri

for Tekrar Deyimi

- Sayaç kontrollü tekrar işlemi yapar.
- Başlangıç durumu/durumları, döngü devam şartı/şartları ve her döngüde yapılacak işlem/işlemler başlık kısmında verilir.
- Her döngüde yapılacak işlemler blok içerisinde yazılır.



32

for Tekrar Deyimi

- Sayaç kontrollü tekrar işlemi for ve while deyimleriyle yapılabilir.

```
for ( initialization; loopContinuationCondition; increment )
    statement

initialization;

while ( !loopContinuationCondition )
{
    statement
    increment;
}
```

33

for Tekrar Deyimi

- **for** deyimi içinde birden fazla başlangıç durumu, döngü devam şartı ve her döngüde yapılacak işlem tanımlanabilir.

```
1 #include <iostream>
2
3 using std::cout;
4 using std::endl;
5
6 int main()
7 {
8     int a=0, b=20, i; // başlangıç değerleri
9
10    // i, a başlangıç değerleri aldı
11    // iki döngü devam şartı verildi
12    // her döngüde iki artırım yapılıyor
13    for (i=0, a=10; (i<10) && (a<15); i++, a++)
14    {
15        // ekrana yazdırma işlemi yapılıyor
16        cout << a+b << endl;
17    }
18
19    system("PAUSE");
20    return 0;
21 }
```


```
C:\Dev-Cpp\ddd.exe
30
31
32
33
34
Press any key to continue . . .
```

34

for Tekrar Deyimi

- Integer sayıların toplanması için örnek.

```
1 // Fig. 5.5: fig05_05.cpp
2 // Summing integers with the for statement.
3 #include <iostream>
4 using std::cout;
5 using std::endl;
6
7 int main()
8 {
9     int total = 0; // initialize total
10
11     // total even integers from 2 through 20
12     for ( int number = 2; number <= 20; number += 2 )
13         total += number;
14
15     // display results
16     cout << "Sum is " << total << endl;
17     system("PAUSE");
18     return 0; // successful termination
19 } // end main
```



35

for Tekrar Deyimi

- for için örnek.

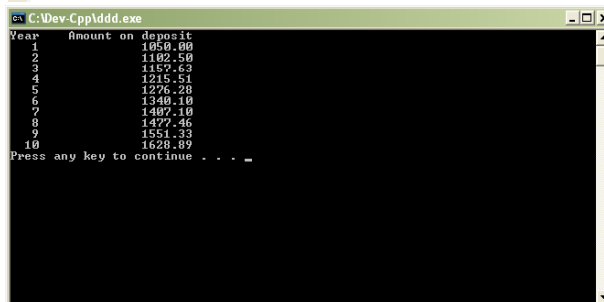
```
1 // Fig. 5.6: fig05_06.cpp
2 // Compound interest calculations with for.
3 #include <iostream>
4
5 using std::cout;
6 using std::endl;
7 using std::fixed;
8
9 #include <iomanip>
10 using std::setw; // enables program to set a field width
11 using std::setprecision;
12
13 #include <cmath> // standard C++ math library
14 using std::pow; // enables program to use function pow
15
16 int main()
17 {
18     double amount; // amount on deposit at end of each year
19     double principal = 1000.0; // initial amount before interest
20     double rate = .05; // interest rate
21
22     // display headers
23     cout << "Year" << setw( 21 ) << "Amount on deposit" << endl;
24
25     // set floating-point number format
26     cout << fixed << setprecision( 2 );
27 }
```

36

for Tekrar Deyimi

■ for için örnek.

```
28 // calculate amount on deposit for each of ten years
29 for ( int year = 1; year <= 10; year++ )
30 {
31     // calculate new amount for specified year
32     amount = principal * pow( 1.0 + rate, year );
33
34     // display the year and the amount
35     cout << setw( 4 ) << year << setw( 21 ) << amount << endl;
36 } // end for
37
38 system("PAUSE");
39 return 0; // indicate successful termination
40 } // end main
```



```
C:\Dev-Cpp\ddd.exe
Year   Amount on deposit
1      1050.00
2      1102.50
3      1157.63
4      1215.51
5      1276.28
6      1340.10
7      1407.10
8      1477.46
9      1551.33
10     1528.89
Press any key to continue . . . _
```

37

Konular

- Giriş
- Kontrol Yapıları
- if Seçme Deyimi
- if ... else Seçme Deyimi
- while Tekrar Deyimi
- Sayaç Kontrollü Tekrar
- Sentinel Kontrollü Tekrar
- İççe Kontrol Deyimleri
- for Tekrar Deyimi
- do ... while Tekrar Deyimi
- switch Çoklu Seçim Deyimi
- break ve continue Deyimleri

do ... while Tekrar Deyimi

- do ... while tekrar deyiminde şart kontrolü sonda yapılır.
- do ... while tekrar deyiminde **döngü içindeki deyimler en az bir kez çalışır.**

```
do
    statement
while ( condition );

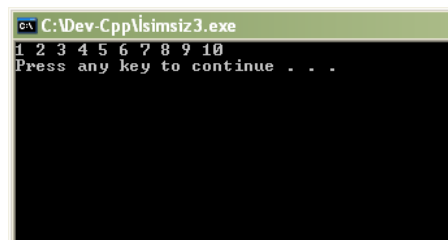
do
{
    statement
} while ( condition );
```

39

do ... while Tekrar Deyimi

- do ... while tekrar deyimi için örnek.

```
1 // Fig. 5.7: fig05_07.cpp
2 // do...while repetition statement.
3 #include <iostream>
4
5 using std::cout;
6 using std::endl;
7
8 int main()
9 {
10     int counter = 1; // initialize counter
11
12     do
13     {
14         cout << counter << " "; // display counter
15         counter++; // increment counter
16     } while ( counter <= 10 ); // end do...while
17
18     cout << endl; // output a newline
19     return 0; // indicate successful termination
20 } // end main
```



```
C:\Dev-Cpp\isimsiz3.exe
1 2 3 4 5 6 7 8 9 10
Press any key to continue . . .
```

40

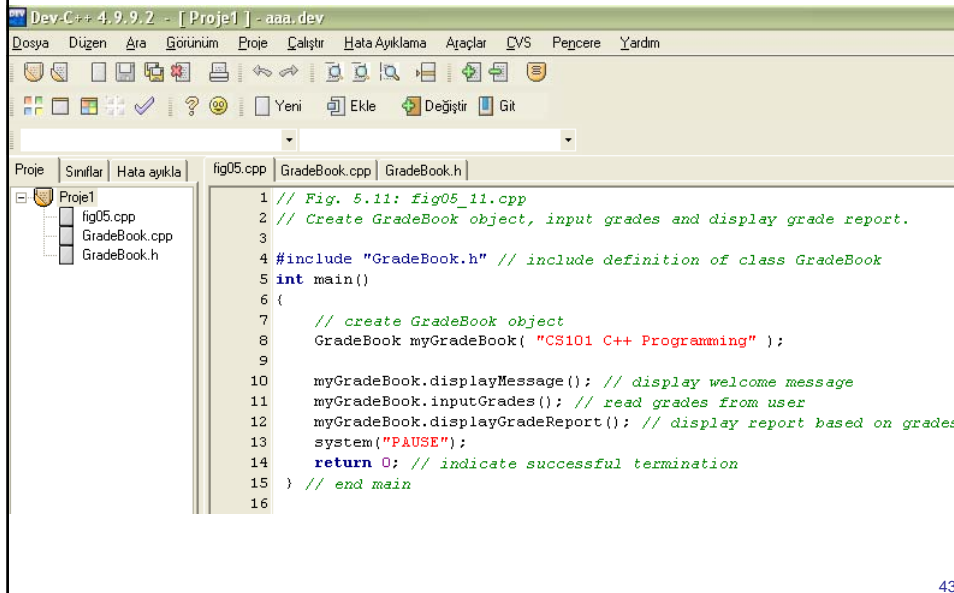
Konular

- Giriş
- Kontrol Yapıları
- if Seçme Deyimi
- if ... else Seçme Deyimi
- while Tekrar Deyimi
- Sayaç Kontrollü Tekrar
- Sentinel Kontrollü Tekrar
- İççe Kontrol Deyimleri
- for Tekrar Deyimi
- do ... while Tekrar Deyimi
- **switch Çoklu Seçim Deyimi**
- break ve continue Deyimleri

switch Çoklu Seçim Deyimi

- switch çoklu seçim deyimi if deyimine benzer şekilde çalışır.
- Belirlenen şartlara göre bir veya birden fazla blok çalıştırılır.
- Her blok sonunun break deyimiyle bitirilmesi gerekir.
- Hiçbir şartın olmaması durumu için default deyimi kullanılır.

switch Çoklu Seçim Deyimi



The screenshot shows the Dev-C++ IDE with a project named 'Proje1'. The file explorer on the left shows 'fig05.cpp', 'GradeBook.cpp', and 'GradeBook.h'. The main editor window displays the code for 'fig05.cpp'.

```
1 // Fig. 5.11: fig05_11.cpp
2 // Create GradeBook object, input grades and display grade report.
3
4 #include "GradeBook.h" // include definition of class GradeBook
5 int main()
6 {
7     // create GradeBook object
8     GradeBook myGradeBook( "CS101 C++ Programming" );
9
10    myGradeBook.displayMessage(); // display welcome message
11    myGradeBook.inputGrades(); // read grades from user
12    myGradeBook.displayGradeReport(); // display report based on grades
13    system("PAUSE");
14    return 0; // indicate successful termination
15 } // end main
16
```

43

switch Çoklu Seçim Deyimi

```
1 // Fig. 5.9: GradeBook.h
2 // Definition of class GradeBook that counts A, B, C, D and F grades.
3 // Member functions are defined in GradeBook.cpp
4 #include <string> // program uses C++ standard string class
5 using std::string;
6 // GradeBook class definition
7 class GradeBook
8 {
9     public:
10    GradeBook( string ); // constructor initializes course name
11    void setCourseName( string ); // function to set the course name
12    string getCourseName(); // function to retrieve the course name
13    void displayMessage(); // display a welcome message
14    void inputGrades(); // input arbitrary number of grades from user
15    void displayGradeReport(); // display a report based on the grades
16     private:
17    string courseName; // course name for this GradeBook
18    int aCount; // count of A grades
19    int bCount; // count of B grades
20    int cCount; // count of C grades
21    int dCount; // count of D grades
22    int fCount; // count of F grades
23 }; // end class GradeBook
```

44

switch Çoklu Seçim Deyimi

```
1 // Fig. 5.10: GradeBook.cpp
2 // Member-function definitions for class GradeBook that
3 // uses a switch statement to count A, B, C, D and F grades.
4 #include <iostream>
5 using std::cout;
6 using std::cin;
7 using std::endl;
8
9 #include "GradeBook.h" // include definition of class GradeBook
10
11 // constructor initializes courseName with string supplied as argument;
12 // initializes counter data members to 0
13 GradeBook::GradeBook( string name )
14 {
15     setCourseName( name ); // validate and store courseName
16     aCount = 0; // initialize count of A grades to 0
17     bCount = 0; // initialize count of B grades to 0
18     cCount = 0; // initialize count of C grades to 0
19     dCount = 0; // initialize count of D grades to 0
20     fCount = 0; // initialize count of F grades to 0
21 } // end GradeBook constructor
22
```

45

switch Çoklu Seçim Deyimi

```
23 // function to set the course name; limits name to 25 or fewer characters
24 void GradeBook::setCourseName( string name )
25 {
26     if ( name.length() <= 25 ) // if name has 25 or fewer characters
27         courseName = name; // store the course name in the object
28     else // if name is longer than 25 characters
29         { // set courseName to first 25 characters of parameter name
30             courseName = name.substr( 0, 25 ); // select first 25 characters
31             cout << "Name \"" << name << "\" exceeds maximum length (25).\n"
32                 << "Limiting courseName to first 25 characters.\n" << endl;
33         } // end if...else
34 } // end function setCourseName
35
36 // function to retrieve the course name
37 string GradeBook::getCourseName()
38 {
39     return courseName;
40 } // end function getCourseName
41
42 // display a welcome message to the GradeBook user
43 void GradeBook::displayMessage()
44 {
45     // this statement calls getCourseName to get the
46     // name of the course this GradeBook represents
47     cout << "Welcome to the grade book for\n" << getCourseName() << "\n"
48         << endl;
49 } // end function displayMessage
50
```

46

switch Çoklu Seçim Deyimi

```
51 // input arbitrary number of grades from user; update grade counter
52 void GradeBook::inputGrades()
53 {
54     int grade; // grade entered by user
55
56     cout << "Enter the letter grades." << endl
57          << "Enter the EOF (CTRL+Z) character to end input." << endl;
58
59     // loop until user types end-of-file key sequence
60     while ( ( grade = cin.get() ) != EOF )
61     {
62         // determine which grade was entered
63         switch ( grade ) // switch statement nested in while
64         {
65             case 'A': // grade was uppercase A
66             case 'a': // or lowercase a
67                 aCount++; // increment aCount
68                 break; // necessary to exit switch
69
70             case 'B': // grade was uppercase B
71             case 'b': // or lowercase b
72                 bCount++; // increment bCount
73                 break; // exit switch
74
75             case 'C': // grade was uppercase C
76             case 'c': // or lowercase c
77                 cCount++; // increment cCount
78                 break; // exit switch
79
```

47

switch Çoklu Seçim Deyimi

```
80         case 'D': // grade was uppercase D
81         case 'd': // or lowercase d
82             dCount++; // increment dCount
83             break; // exit switch
84
85         case 'F': // grade was uppercase F
86         case 'f': // or lowercase f
87             fCount++; // increment fCount
88             break; // exit switch
89
90         case '\n': // ignore newlines,
91         case '\t': // tabs,
92         case ' ': // and spaces in input
93             break; // exit switch
94
95         default: // catch all other characters
96             cout << "Incorrect letter grade entered."
97                  << " Enter a new grade." << endl;
98             break; // optional; will exit switch anyway
99     } // end switch
100 } // end while
101 } // end function inputGrades
102
```

48

switch Çoklu Seçim Deyimi

```
103 // display a report based on the grades entered by user
104 void GradeBook::displayGradeReport()
105 {
106     // output summary of results
107     cout << "\n\nNumber of students who received each letter grade:"
108         << "\nA: " << aCount // display number of A grades
109         << "\nB: " << bCount // display number of B grades
110         << "\nC: " << cCount // display number of C grades
111         << "\nD: " << dCount // display number of D grades
112         << "\nF: " << fCount // display number of F grades
113         << endl;
114 } // end function displayGradeReport
```

- `cin.get()` deyimi klavyeden karakter alır ve integer değerini `grade` değişkenine aktarır.

```
cout << "The character (" << 'a' << ") has the value "
     << static_cast< int > ( 'a' ) << endl;
```

Ekran çıktısı

The character (a) has the value 97

- EOF karakteri UNIX makineler için CTRL+D, Windows tabanlı makineler için CTRL+Z karakterini ifade eder.

49

switch Çoklu Seçim Deyimi

```
1 // Fig. 5.11: fig05_11.cpp
2 // Create GradeBook object, input grades and display grade report.
3
4 #include "GradeBook.h" // include definition of class GradeBook
5 int main()
6 {
7     // create GradeBook object
8     GradeBook myGradeBook( "CS101 C++ Programming" );
9
10    myGradeBook.displayMessage(); // display welcome message
11    myGradeBook.inputGrades(); // read grades from user
12    myGradeBook.displayGradeReport(); // display report based on grades
13    system("PAUSE");
14    return 0; // indicate successful termination
15 } // end main
```

50

switch Çoklu Seçim Deyimi

```
C:\Dev-Cpp\aaa.exe
Welcome to the grade book for
CS101 C++ Programming!

Enter the letter grades.
Enter the EOF (CTRL+Z) character to end input.
a
b
c
d
f
e
Incorrect letter grade entered. Enter a new grade.
a
a
c
d
d
d
d
h
^Z

Number of students who received each letter grade:
A: 3
B: 2
C: 2
D: 4
F: 1
Press any key to continue . . . _
```

51

Konular

- Giriş
- Kontrol Yapıları
- if Seçme Deyimi
- if ... else Seçme Deyimi
- while Tekrar Deyimi
- Sayaç Kontrollü Tekrar
- Sentinel Kontrollü Tekrar
- İççe Kontrol Deyimleri
- for Tekrar Deyimi
- do ... while Tekrar Deyimi
- switch Çoklu Seçim Deyimi
- **break ve continue Deyimleri**

break ve continue Deyimleri

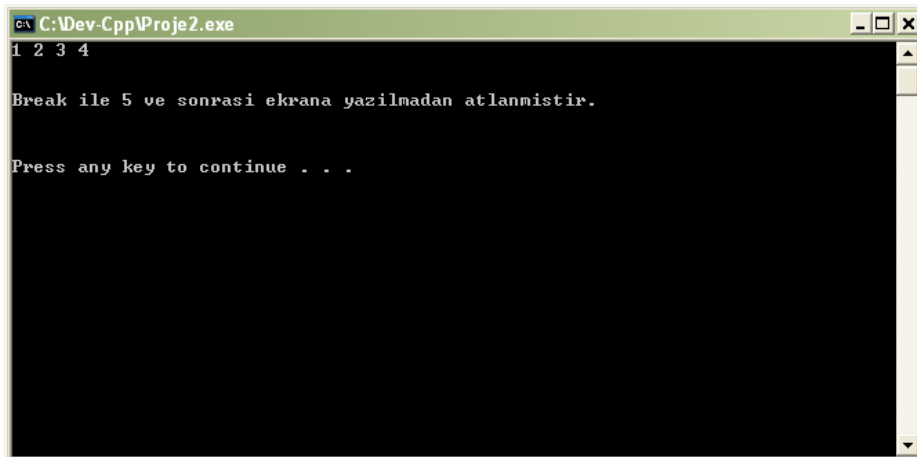
- **break** deyimi, döngünün kalan iterasyonlarını yapmadan çıkar.

```
1 // Fig. 5.14: fig05_14.cpp
2 // break statement exiting a for statement.
3 #include <iostream>
4 using std::cout;
5 using std::endl;
6
7 int main()
8 {
9     for ( int count = 1; count <= 10; count++ ) // loop 10 times
10    {
11        if ( count == 5 ) // if count is 5,
12            break; // break loop
13
14        cout << count << " ";
15    } // end for
16
17    cout << "\n\nBreak ile 5 ve sonrasi ekrana yazilmadan atlanmistir.\n\n" << endl;
18    system("PAUSE");
19    return 0; // indicate successful termination
20 } // end main
```

53

break ve continue Deyimleri

- **break** deyimi ekran çıktısı.



```
C:\Dev-Cpp\Proje2.exe
1 2 3 4

Break ile 5 ve sonrasi ekrana yazilmadan atlanmistir.

Press any key to continue . . .
```

54

break ve continue Deyimleri

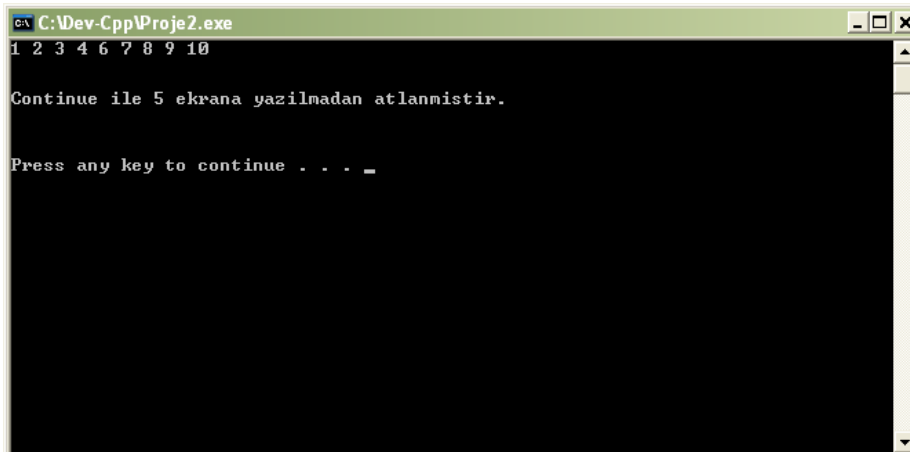
- **continue** deyimi, döngünün şimdiki iterasyonunun kalan kısmını atlar ve sonraki iterasyonla devam eder.

```
1 // Fig. 5.14: fig05_14.cpp
2 // continue statement terminating an iteration of a for statement.
3 #include <iostream>
4 using std::cout;
5 using std::endl;
6
7 int main()
8 {
9     for ( int count = 1; count <= 10; count++ ) // loop 10 times
10    {
11        if ( count == 5 ) // if count is 5,
12            continue; // skip remaining code in loop
13
14        cout << count << " ";
15    } // end for
16
17    cout << "\n\nContinue ile 5 ekrana yazılmadan atlanmıştır.\n\n" << endl;
18    system("PAUSE");
19    return 0; // indicate successful termination
20 } // end main
```

55

break ve continue Deyimleri

- **continue** deyimi ekran çıktısı.



```
C:\Dev-Cpp\Proje2.exe
1 2 3 4 6 7 8 9 10
Continue ile 5 ekrana yazılmadan atlanmıştır.
Press any key to continue . . . _
```

56

Ödev

- Klavyeden n adet öğrenciye ait numara, ad, soyad, vize, odev, katilim ve final notları tamsayı olarak girilecek.
- Veri girişi sonlandırıldıktan sonra ekrana aşağıdaki menü gelecek.
- Klavyeden 0 girilene kadar menü ekrana tekrar gelecek ve seçilen işlemin sonucunu ekrana yazacaktır.
- Her işlem sonucundan sonra ekran temizlenerek menü tekrar gelecektir. `system("CLS")` ; ile ekran temizlenebilir.
- Programda dizi kullanılmayacaktır. Tüm işlemler için bir sınıf oluşturulacaktır.

```
1- Ortalaması en yüksek öğrenci bilgileri
2- Ortalaması en düşük öğrenci bilgileri
3- Sınıf ortalaması
4- A harf notu alan öğrenci sayısı
5- B harf notu alan öğrenci sayısı
6- C harf notu alan öğrenci sayısı
7- D harf notu alan öğrenci sayısı
8- F harf notu alan öğrenci sayısı
9- Başarılı öğrenci sayısı (A+B+C)
10- Başarısız olan öğrenci sayısı (D+F)
İşlem türünü seçiniz (1...10, Çıkış için 0 giriniz) :
```

- Ortalama not hesaplamasında vize (%30), odev (%20), katilim (%10) ve final (%40) oranında alınacak.
- Harf notu için A = 90-100, B = 80-89, C = 70-79, D = 60-69 ve F = 0-59 alınacaktır.
- Öğrencilerden ortalaması en yüksek ve en düşük olanların numarası, adı ve soyadı, ortalama notunu ile harf notunu ekrana yazacaktır.

57

Ödev

- Örnek ekran çıktıları aşağıdadır.

```
1- Ortalaması en yüksek öğrenci bilgileri
2- Ortalaması en düşük öğrenci bilgileri
3- Sınıf ortalaması
4- A harf notu alan öğrenci sayısı
5- B harf notu alan öğrenci sayısı
6- C harf notu alan öğrenci sayısı
7- D harf notu alan öğrenci sayısı
8- F harf notu alan öğrenci sayısı
9- Başarılı öğrenci sayısı (A+B+C)
10- Başarısız olan öğrenci sayısı (D+F)
İşlem türünü seçiniz (1...10, Çıkış için 0 giriniz) : 1
```

```
Ortalaması en yüksek öğrenci bilgileri
*****
Numara: 90807060      Adı Soyadı: Ahmet Can      Ortalama: 97      Harf: A

Devam etmek için bir tuşa basınız...
```

58

Ödev

- 1- Ortalaması en yüksek öğrenci bilgileri
 - 2- Ortalaması en düşük öğrenci bilgileri
 - 3- Sınıf ortalaması
 - 4- A harf notu alan öğrenci sayısı
 - 5- B harf notu alan öğrenci sayısı
 - 6- C harf notu alan öğrenci sayısı
 - 7- D harf notu alan öğrenci sayısı
 - 8- F harf notu alan öğrenci sayısı
 - 9- Başarılı öğrenci sayısı (A+B+C)
 - 10- Başarısız olan öğrenci sayısı (D+F)
- İşlem türünü seçiniz (1...10, Çıkış için 0 giriniz) : 3

Sınıf Ortalaması: 67.65

Devam etmek için bir tuşa basınız...

59

Ödev

- 1- Ortalaması en yüksek öğrenci bilgileri
 - 2- Ortalaması en düşük öğrenci bilgileri
 - 3- Sınıf ortalaması
 - 4- A harf notu alan öğrenci sayısı
 - 5- B harf notu alan öğrenci sayısı
 - 6- C harf notu alan öğrenci sayısı
 - 7- D harf notu alan öğrenci sayısı
 - 8- F harf notu alan öğrenci sayısı
 - 9- Başarılı öğrenci sayısı (A+B+C)
 - 10- Başarısız olan öğrenci sayısı (D+F)
- İşlem türünü seçiniz (1...10, Çıkış için 0 giriniz) : 9

Başarılı öğrenci sayısı (A+B+C): 25

Devam etmek için bir tuşa basınız...

60

Ödev

- 1- Ortalaması en yüksek öğrenci bilgileri
 - 2- Ortalaması en düşük öğrenci bilgileri
 - 3- Sınıf ortalaması
 - 4- A harf notu alan öğrenci sayısı
 - 5- B harf notu alan öğrenci sayısı
 - 6- C harf notu alan öğrenci sayısı
 - 7- D harf notu alan öğrenci sayısı
 - 8- F harf notu alan öğrenci sayısı
 - 9- Başarılı öğrenci sayısı (A+B+C)
 - 10- Başarısız olan öğrenci sayısı (D+F)
- İşlem türünü seçiniz (1...10, Çıkış için 0 giriniz) : 4

A harf notu alan öğrenci sayısı: 3

Devam etmek için bir tuşa basınız...

61

Ödev

- 1- Ortalaması en yüksek öğrenci bilgileri
 - 2- Ortalaması en düşük öğrenci bilgileri
 - 3- Sınıf ortalaması
 - 4- A harf notu alan öğrenci sayısı
 - 5- B harf notu alan öğrenci sayısı
 - 6- C harf notu alan öğrenci sayısı
 - 7- D harf notu alan öğrenci sayısı
 - 8- F harf notu alan öğrenci sayısı
 - 9- Başarılı öğrenci sayısı (A+B+C)
 - 10- Başarısız olan öğrenci sayısı (D+F)
- İşlem türünü seçiniz (1...10, Çıkış için 0 giriniz) : 10

Başarısız olan öğrenci sayısı (D+F): 6

Devam etmek için bir tuşa basınız...

62