

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

---

Hazırlayan: M.Ali Akcayol  
Gazi Üniversitesi  
Bilgisayar Mühendisliği Bölümü

## Konular

---

- Giriş
- Sınıflar, Nesnelere, Fonksiyon ve Veri Üyeleri
- Sınıf ve Fonksiyon Üyeleri
- Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları
- Nesnelere Başlatma ve Constructor Fonksiyonu
- Sınıflar ve Ayrı Dosyalarda Kullanılması
- Interface ve Implementation
- Set Fonksiyonuyla Veri Geçerleme

## Giriş

- Bir program `main()` fonksiyonuyla bir veya daha fazla sınıftan oluşabilir.
- Her sınıf veri üyeleri ve fonksiyon üyelerine sahiptir.
- Bir sınıf kendisiyle ilişkili tüm veriler ve fonksiyonları içerir.

3

## Konular

- Giriş
- **Sınıflar, Nesnelere, Fonksiyon ve Veri Üyeleri**
- Sınıf ve Fonksiyon Üyeleri
- Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları
- Nesnelere Başlatma ve Constructor Fonksiyonu
- Sınıflar ve Ayrı Dosyalarda Kullanılması
- Interface ve Implementation
- Set Fonksiyonuyla Veri Geçerleme

## Sınıflar, Nesneler, Fonksiyon ve Veri Üyeleri

- Bir sınıf yapılan işlemleri fonksiyonları kullanılarak gerçekleştirir.
- Fonksiyonlar gerekirse veri üyelerini (data member) kullanabilir.
- Sınıf üzerinde çalışan kullanıcı fonksiyonların yaptığı işin karmaşıklığından etkilenmez.
- Kullanıcı bir sınıfın tanımlanmış arayüzüne (interface) erişir.
- Bir sınıftan birçok nesne (object) üretilebilir. Her nesne kendisine ait özellikler ve fonksiyonlara sahip olur.

5

## Konular

- Giriş
- Sınıflar, Nesneler, Fonksiyon ve Veri Üyeleri
- **Sınıf ve Fonksiyon Üyeleri**
- Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları
- Nesnelere Başlatma ve Constructor Fonksiyonu
- Sınıflar ve Ayrı Dosyalarda Kullanılması
- Interface ve Implementation
- Set Fonksiyonuyla Veri Geçerleme

## Sınıf ve Fonksiyon Üyeleri

```
1 // Fig. 3.1: fig03_01.cpp
2 // Define class GradeBook with a member function displayMessage;
3 // Create a GradeBook object and call its displayMessage function.
4 #include <iostream>
5 using std::cout;
6 using std::endl;
7
8 // GradeBook class definition
9 class GradeBook
10 {
11 public:
12     // function that displays a welcome message to the GradeBook user
13     void displayMessage()
14     {
15         cout << "Welcome to the Grade Book!" << endl;
16     } // end function displayMessage
17 }; // end class GradeBook
18
19 // function main begins program execution
20 int main()
21 {
22     GradeBook myGradeBook; // create a GradeBook object named myGradeBook
23     myGradeBook.displayMessage(); // call object's displayMessage function
24     return 0; // indicate successful termination
25 } // end main
```

Welcome to the Grade Book!

7

## Sınıf ve Fonksiyon Üyeleri

- Satır 9-17 arasında **GradeBook** adlı bir sınıf tanımlanıyor.
- Satır 13-16 arasında ekrana mesaj yazan **displayMessage()** adlı bir fonksiyon tanımlanıyor.
- **main()** fonksiyonu **GradeBook** sınıfından **myGradeBook** adlı bir örnek (instance, object) oluşturmaktadır.
- **myGradeBook** nesnesinin **displayMessage()** fonksiyonu çağırılarak ekrana mesaj yazdırılmaktadır.
- **public** anahtar kelimesi (erişim belirleyicisi-access specifier) bu üyeye herkes tarafından ulaşılacağını ifade eder. (gelecek derslerde ayrıntılı görülecektir.)

8

## Konular

- Giriş
- Sınıflar, Nesnelere, Fonksiyon ve Veri Üyeleri
- Sınıf ve Fonksiyon Üyeleri
- **Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları**
- Nesnelere Başlatma ve Constructor Fonksiyonu
- Sınıflar ve Ayrı Dosyalarda Kullanılması
- Interface ve Implementation
- Set Fonksiyonuyla Veri Geçerleme

## Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları

```
1 // Fig. 3.5: fig03_05.cpp
2 // Define class GradeBook that contains a courseName data member
3 // and member functions to set and get its value;
4 // Create and manipulate a GradeBook object with these functions.
5 #include <iostream>
6 using std::cout;
7 using std::cin;
8 using std::endl;
9
10 #include <string> // program uses C++ standard string class
11 using std::string;
12 using std::getline;
13
14 // GradeBook class definition
15 class GradeBook
16 {
17 public:
18     // function that sets the course name
```

## Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları

```
19 void setCourseName( string name )
20 {
21     courseName = name; // store the course name in the object
22 } // end function setCourseName
23
24 // function that gets the course name
25 string getCourseName()
26 {
27     return courseName; // return the object's courseName
28 } // end function getCourseName
29
30 // function that displays a welcome message
31 void displayMessage()
32 {
33     // this statement calls getCourseName to get the
34     // name of the course this GradeBook represents
35     cout << "Welcome to the grade book for\n" << getCourseName() << "!"
36         << endl;
37 } // end function displayMessage
38 private:
39     string courseName; // course name for this GradeBook
40 }; // end class GradeBook
41
```

11

## Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları

```
42 // function main begins program execution
43 int main()
44 {
45     string nameOfCourse; // string of characters to store the course name
46     GradeBook myGradeBook; // create a GradeBook object named myGradeBook
47
48     // display initial value of courseName
49     cout << "Initial course name is: " << myGradeBook.getCourseName()
50         << endl;
51
52     // prompt for, input and set course name
53     cout << "\nPlease enter the course name:" << endl;
54     getline( cin, nameOfCourse ); // read a course name with blanks
55     myGradeBook.setCourseName( nameOfCourse ); // set the course name
56
57     cout << endl; // outputs a blank line
58     myGradeBook.displayMessage(); // display message with new course name
59     return 0; // indicate successful termination
60 } // end main
```

12

## Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları

### Ekran çıktısı

```
Initial course name is:  
  
Please enter the course name:  
CS101 Introduction to C++ Programming  
  
Welcome to the grade book for  
CS101 Introduction to C++ Programming!
```

13

## Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları

- Satır 15-40 arasında `GradeBook` adlı bir sınıf tanımlanıyor.
- `public` ve `private` olarak iki kısımdan oluşmaktadır.
- `private` üyelere sadece `GradeBook` sınıfının fonksiyon üyeleri (`setCourseName`, `getCourseName`, `displayMessage`) tarafından erişilebilir.
- Private bir üyeye dışarıdan erişmek istendiğinde (`myGradeBook.courseName`) aşağıdaki hata oluşur:  

```
cannot access private member declared in class 'GradeBook'
```
- Satır 45'teki `string` değişkeni `<string>` header file içinde tanımlıdır.

14

## Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları

- Satır 54'teki `getline` fonksiyonu birden fazla boşluk içeren girişleri almak için kullanılır. (`cin` ilk boşluğa kadar alır. `cin >> nameOfCourse`)
- Satır 19, `setCourseName` fonksiyonu `courseName` değişkenine değer atar.
- Satır 25, `getCourseName` fonksiyonu `courseName` değişkeninin değerini geri döndürür.
- Satır 31, `displayMessage` fonksiyonu ekrana mesaj metni yazar.
- Satır 36, `endl` satır sonu yapmak için kullanılır.

15

## Konular

- Giriş
- Sınıflar, Nesnelere, Fonksiyon ve Veri Üyeleri
- Sınıf ve Fonksiyon Üyeleri
- Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları
- Nesnelere Başlatma ve Constructor Fonksiyonu
- Sınıflar ve Ayrı Dosyalarda Kullanılması
- Interface ve Implementation
- Set Fonksiyonuyla Veri Geçerleme



## Nesneleri Başlatma ve Constructor Fonksiyonu

- Bir sınıftan oluşturulan nesneye ilk tanımlandığında başlangıç değeri atanabilir.
- Başlangıç değeri atamak için kullanılan fonksiyonun (constructor) adı sınıf adıyla aynıdır
- Constuctor bir veya birden fazla **data member** veya **function member** çağırabilir.

17

## Nesneleri Başlatma ve Constructor Fonksiyonu

```
1 // Fig. 3.7: fig03_07.cpp
2 // Instantiating multiple objects of the GradeBook class and using
3 // the GradeBook constructor to specify the course name
4 // when each GradeBook object is created.
5 #include <iostream>
6 using std::cout;
7 using std::endl;
8
9 #include <string> // program uses C++ standard string class
10 using std::string;
11
12 // GradeBook class definition
13 class GradeBook
14 {
15 public:
16     // constructor initializes courseName with string supplied as argument
17     GradeBook( string name )
18     {
19         setCourseName( name ); // call set function to initialize courseName
20     } // end GradeBook constructor
21
22     // function to set the course name
23     void setCourseName( string name )
24     {
25         courseName = name; // store the course name in the object
26     } // end function setCourseName
27
28     // function to get the course name
29     string getCourseName()
```

18

## Nesneleri Başlatma ve Constructor Fonksiyonu

```
30     {
31         return courseName; // return object's courseName
32     } // end function getCourseName
33
34     // display a welcome message to the GradeBook user
35     void displayMessage()
36     {
37         // call getCourseName to get the courseName
38         cout << "Welcome to the grade book for\n" << getCourseName()
39              << "\n" << endl;
40     } // end function displayMessage
41 private:
42     string courseName; // course name for this GradeBook
43 }; // end class GradeBook
44
45 // function main begins program execution
46 int main()
47 {
48     // create two GradeBook objects
49     GradeBook gradeBook1( "CS101 Introduction to C++ Programming" );
50     GradeBook gradeBook2( "CS102 Data Structures in C++" );
51
52     // display initial value of courseName for each GradeBook
53     cout << "gradeBook1 created for course: " << gradeBook1.getCourseName()
54          << "\ngradeBook2 created for course: " << gradeBook2.getCourseName()
55          << endl;
56     return 0; // indicate successful termination
57 } // end main
```

19

## Nesneleri Başlatma ve Constructor Fonksiyonu

### Ekran çıktısı

```
gradeBook1 created for course: CS101 Introduction to C++ Programming
gradeBook2 created for course: CS102 Data Structures in C++
```

20

## Nesneleri Başlatma ve Constructor Fonksiyonu

- Satır 17-20 **GradeBook** sınıfının constructor'ını tanılamaktadır.
- Yeni bir nesne oluşturulduğunda constructor'da belirlenen parametreler verilir (satır 49-50).
- Constructor 'lar herhangi bir değer döndürmezler. O yüzden **return** kullanılmaz.

21

## Konular

- Giriş
- Sınıflar, Nesnelere, Fonksiyon ve Veri Üyeleri
- Sınıf ve Fonksiyon Üyeleri
- Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları
- Nesnelere Başlatma ve Constructor Fonksiyonu
- **Sınıflar ve Ayrı Dosyalarda Kullanılması**
- Interface ve Implementation
- Set Fonksiyonuyla Veri Geçerleme

## Sınıflar ve Ayrı Dosyalarda Kullanılması

- Bir sınıfı, `main()` fonksiyonu ile birlikte oluşturduğumuzda başka programlarda include yaparak kullanamayız.
- Derleyici iki tane `main()` fonksiyonu olduğu için hata mesajı verir.
- Header file tanımlayarak başka programlarda include ederek kullanabiliriz.
- Bir header file `.h` uzantılıdır ve `main()` fonksiyonu bulundurmaz.
- Bir header file run yapılmaya çalışılırsa derleyici hata aşağıdaki gibi bir mesaj verir.

undefined reference to 'main'

23

## Sınıflar ve Ayrı Dosyalarda Kullanılması

```
1 // Fig. 3.9: GradeBook.h
2 // GradeBook class definition in a separate file from main.
3 #include <iostream>
4 using std::cout;
5 using std::endl;
6
7 #include <string> // class GradeBook uses C++ standard string class
8 using std::string;
9
10 // GradeBook class definition
11 class GradeBook
12 {
13 public:
14     // constructor initializes courseName with string supplied as argument
15     GradeBook( string name )
16     {
17         setCourseName( name ); // call set function to initialize courseName
18     } // end GradeBook constructor
19
20     // function to set the course name
21     void setCourseName( string name )
22     {
23         courseName = name; // store the course name in the object
24     } // end function setCourseName
25
26     // function to get the course name
27     string getCourseName()
```

24

## Sınıflar ve Ayrı Dosyalarda Kullanılması

```
28     {
29         return courseName; // return object's courseName
30     } // end function getCourseName
31
32     // display a welcome message to the GradeBook user
33     void displayMessage()
34     {
35         // call getCourseName to get the courseName
36         cout << "Welcome to the grade book for\n" << getCourseName()
37             << "!" << endl;
38     } // end function displayMessage
39 private:
40     string courseName; // course name for this GradeBook
41 }; // end class GradeBook

```

// Header file tanımı sonu.

25

## Sınıflar ve Ayrı Dosyalarda Kullanılması

```
1 // Fig. 3.10: fig03_10.cpp
2 // Including class GradeBook from file GradeBook.h for use in main.
3 #include <iostream>
4 using std::cout;
5 using std::endl;
6
7 #include "GradeBook.h" // include definition of class GradeBook
8
9 // function main begins program execution
10 int main()
11 {
12     // create two GradeBook objects
13     GradeBook gradeBook1( "CS101 Introduction to C++ Programming" );
14     GradeBook gradeBook2( "CS102 Data Structures in C++" );
15
16     // display initial value of courseName for each GradeBook
17     cout << "gradeBook1 created for course: " << gradeBook1.getCourseName()
18         << "\ngradeBook2 created for course: " << gradeBook2.getCourseName()
19         << endl;
20     return 0; // indicate successful termination
21 } // end main

```

```
gradeBook1 created for course: CS101 Introduction to C++ Programming
gradeBook2 created for course: CS102 Data Structures in C++

```

26

## Sınıflar ve Ayrı Dosyalarda Kullanılması

- Header file <> yerine "" ile gösterilir.
- Header file satırına geldiğinde derleyici önce programın source code'unun bulunduğu dizine bakar. Orda yoksa library file (<> ile gösterilen) dizinine bakar.
- Header file'ların program kaynak koduyla aynı dizinde olması veya library file dizininde yer almalıdır.
- Header file kullanımı ile **reusability** artmaktadır.

27

## Konular

- Giriş
- Sınıflar, Nesnelere, Fonksiyon ve Veri Üyeleri
- Sınıf ve Fonksiyon Üyeleri
- Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları
- Nesnelere Başlatma ve Constructor Fonksiyonu
- Sınıflar ve Ayrı Dosyalarda Kullanılması
- **Interface ve Implementation**
- Set Fonksiyonuyla Veri Geçerleme

## Interface ve Implementation

- Header file sadece bir sınıfın prototipini belirleyebilir.
- Sınıfın tüm fonksiyon üyelerinin kodu ayrı bir kaynak kodu ile oluşturulabilir.
- Header file **interface**, diğer kaynak kod ise **implementation** kısmını oluşturur.
- Kaynak kodun adı ile header file'in adı aynıdır, uzantısı farklıdır (.h, .cpp).

29

## Interface ve Implementation

- Prototip sınıf tanımlaması fonksiyonların yaptığı işlerin farklı kullanıcılar tarafından değiştirilmesine olanak sağlar.
- Prototip içinde fonksiyon tanımlamaları sonuna ; konulur.
- Prototip sınıf tanımlamalarında fonksiyonların değişken adlarının yazılması opsiyoneldir.

30

## Interface ve Implementation

### Prototip sınıf tanımlaması

```
1 // Fig. 3.11: GradeBook.h
2 // GradeBook class definition. This file presents GradeBook's public
3 // interface without revealing the implementations of GradeBook's member
4 // functions, which are defined in GradeBook.cpp.
5 #include <string> // class GradeBook uses C++ standard string class
6 using std::string;
7
8 // GradeBook class definition
9 class GradeBook
10 {
11 public:
12     GradeBook( string ); // constructor that initializes courseName
13     void setCourseName( string ); // function that sets the course name
14     string getCourseName(); // function that gets the course name
15     void displayMessage(); // function that displays a welcome message
16 private:
17     string courseName; // course name for this GradeBook
18 }; // end class GradeBook
```

31

## Interface ve Implementation

### Fonksiyon üyelerin tanımlaması - 1

```
1 // Fig. 3.12: GradeBook.cpp
2 // GradeBook member-function definitions. This file contains
3 // implementations of the member functions prototyped in GradeBook.h.
4 #include <iostream>
5 using std::cout;
6 using std::endl;
7
8 #include "GradeBook.h" // include definition of class GradeBook
9
10 // constructor initializes courseName with string supplied as argument
11 GradeBook::GradeBook( string name )
12 {
13     setCourseName( name ); // call set function to initialize courseName
14 } // end GradeBook constructor
15
16 // function to set the course name
17 void GradeBook::setCourseName( string name )
18 {
19     courseName = name; // store the course name in the object
20 } // end function setCourseName
```

32



## Interface ve Implementation

### Fonksiyon üyelerin tanımlaması - 2

```
21
22 // function to get the course name
23 string GradeBook::getCourseName()
24 {
25     return courseName; // return object's courseName
26 } // end function getCourseName
27
28 // display a welcome message to the GradeBook user
29 void GradeBook::displayMessage()
30 {
31     // call getCourseName to get the courseName
32     cout << "Welcome to the grade book for\n" << getCourseName()
33         << "\n" << endl;
34 } // end function displayMessage
```

33

## Interface ve Implementation

### Tanımlanan sınıfın kullanılması

```
1 // Fig. 3.13: fig03_13.cpp
2 // GradeBook class demonstration after separating
3 // its interface from its implementation.
4 #include <iostream>
5 using std::cout;
6 using std::endl;
7
8 #include "GradeBook.h" // include definition of class GradeBook
9
10 // function main begins program execution
11 int main()
12 {
13     // create two GradeBook objects
14     GradeBook gradeBook1( "CS101 Introduction to C++ Programming" );
15     GradeBook gradeBook2( "CS102 Data Structures in C++" );
16
17     // display initial value of courseName for each GradeBook
18     cout << "gradeBook1 created for course: " << gradeBook1.getCourseName()
19         << "\ngradeBook2 created for course: " << gradeBook2.getCourseName()
20         << endl;
21     return 0; // indicate successful termination
22 } // end main
```

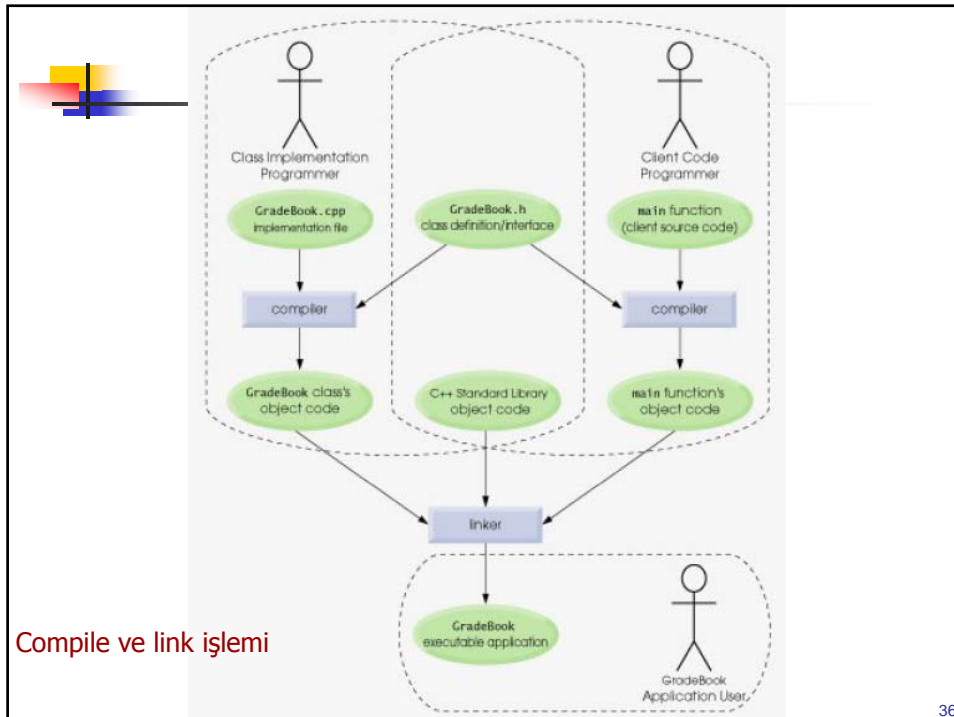
34

## Interface ve Implementation

Ekran çıktısı

```
gradeBook1 created for course: CS101 Introduction to C++ Programming  
gradeBook2 created for course: CS102 Data Structures in C++
```

35



36

## Konular

- Giriş
- Sınıflar, Nesneler, Fonksiyon ve Veri Üyeleri
- Sınıf ve Fonksiyon Üyeleri
- Sınıf ve Veri Üyeleri, Get ve Set Fonksiyonları
- Nesneleri Başlatma ve Constructor Fonksiyonu
- Sınıflar ve Ayrı Dosyalarda Kullanılması
- Interface ve Implementation
- **Set Fonksiyonuyla Veri Geerleme**

## Set Fonksiyonuyla Veri Geerleme

- Bir sınıfın data üyelerinin deęerlerinin geerlilięi set fonksiyonları tarafından yapılabilir (validation).
- Atama işlemi yapılırken istenen şartlar kontrol edilir.
- İstenen şartlar sağlanmıyorsa kullanıcıya mesajla uyarı verilebilir.

## Set Fonksiyonuyla Veri Geçerleme

### Sınıf prototip tanımı

```
1 // Fig. 3.15: GradeBook.h
2 // GradeBook class definition presents the public interface of
3 // the class. Member-function definitions appear 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 that initializes a GradeBook object
12     void setCourseName( string ); // function that sets the course name
13     string getCourseName(); // function that gets the course name
14     void displayMessage(); // function that displays a welcome message
15 private:
16     string courseName; // course name for this GradeBook
17 }; // end class GradeBook
```

39

## Set Fonksiyonuyla Veri Geçerleme

### Sınıf fonksiyonlarının tanımı - 1

```
1 // Fig. 3.16: GradeBook.cpp
2 // Implementations of the GradeBook member-function definitions.
3 // The setCourseName function performs validation.
4 #include <iostream>
5 using std::cout;
6 using std::endl;
7
8 #include "GradeBook.h" // include definition of class GradeBook
9
10 // constructor initializes courseName with string supplied as argument
11 GradeBook::GradeBook( string name )
12 {
13     setCourseName( name ); // validate and store courseName
14 } // end GradeBook constructor
15
```

40

## Set Fonksiyonuyla Veri Geçerleme

### Sınıf fonksiyonlarının tanımı - 2

```
16 // function that sets the course name;
17 // ensures that the course name has at most 25 characters
18 void GradeBook::setCourseName( string name )
19 {
20     if ( name.length() <= 25 ) // if name has 25 or fewer characters
21         courseName = name; // store the course name in the object
22
23     if ( name.length() > 25 ) // if name has more than 25 characters
24     {
25         // set courseName to first 25 characters of parameter name
26         courseName = name.substr( 0, 25 ); // start at 0, length of 25
27
28         cout << "Name \"" << name << "\" exceeds maximum length (25).\n"
29              << "Limiting courseName to first 25 characters.\n" << endl;
30     } // end if
31 } // end function setCourseName
32
```

41

## Set Fonksiyonuyla Veri Geçerleme

### Sınıf fonksiyonlarının tanımı - 3

```
33 // function to get the course name
34 string GradeBook::getCourseName()
35 {
36     return courseName; // return object's courseName
37 } // end function getCourseName
38
39 // display a welcome message to the GradeBook user
40 void GradeBook::displayMessage()
41 {
42     // call getCourseName to get the courseName
43     cout << "Welcome to the grade book for\n" << getCourseName()
44          << "\n" << endl;
45 } // end function displayMessage
```

42

## Set Fonksiyonuyla Veri Geçerleme

- `length()` fonksiyonu bir string içindeki karakter sayısını verir.
- `substr()` fonksiyonu bir string içinden bir kısmı almak için kullanılır.
- `setCourseName()` fonksiyonu `name` değişkeninin karakter sayısını kontrol eder.
- `name` değişkeninin karakter sayısı 25'ten büyükse ilke 25 karakteri alır.
- Aşağıdaki satırla karakter sayısı 25'ten büyükse kullanıcıya mesaj verilir.

```
cout << "Name \"" << name << "\" exceeds maximum length (25).\n"
      << "Limiting courseName to first 25 characters.\n" << endl;
```

43

## Set Fonksiyonuyla Veri Geçerleme

### Tanımlanan sınıfın kullanılması - 1

```
1 // Fig. 3.17: fig03_17.cpp
2 // Create and manipulate a GradeBook object; illustrate validation.
3 #include <iostream>
4 using std::cout;
5 using std::endl;
6
7 #include "GradeBook.h" // include definition of class GradeBook
8
9 // function main begins program execution
10 int main()
11 {
12     // create two GradeBook objects;
13     // initial course name of gradeBook1 is too long
14     GradeBook gradeBook1( "CS101 Introduction to Programming in C++" );
15     GradeBook gradeBook2( "CS102 C++ Data Structures" );
16
17     // display each GradeBook's courseName
18     cout << "gradeBook1's initial course name is: "
19           << gradeBook1.getCourseName()
20           << "\ngradeBook2's initial course name is: "
21           << gradeBook2.getCourseName() << endl;
```

44

## Set Fonksiyonuyla Veri Geçerleme

### Tanımlanan sınıfın kullanılması - 2

```
22
23 // modify myGradeBook's courseName (with a valid-length string)
24 gradeBook1.setCourseName( "CS101 C++ Programming" );
25
26 // display each GradeBook's courseName
27 cout << "\ngradeBook1's course name is: "
28     << gradeBook1.getCourseName()
29     << "\ngradeBook2's course name is: "
30     << gradeBook2.getCourseName() << endl;
31 return 0; // indicate successful termination
32 } // end main
```

45

## Set Fonksiyonuyla Veri Geçerleme

### Ekran çıktısı

```
Name "CS101 Introduction to Programming in C++" exceeds maximum length (25).
Limiting courseName to first 25 characters.
```

```
gradeBook1's initial course name is: CS101 Introduction to Pro
gradeBook2's initial course name is: CS102 C++ Data Structures
```

```
gradeBook1's course name is: CS101 C++ Programming
gradeBook2's course name is: CS102 C++ Data Structures
```

46

## Ödev

- Bir hesap makinesi sınıfı oluşturunuz.
- Hesap makinesi sınıfını prototip ve kaynak kodla geliştiriniz.
- Bir program ile include yaparak çalıştırınız.
- Hesap makinesi toplama, çıkarma, çarpma ve bölme işlemleri yapacaktır.
- İşlemler aşağıdaki gibi ardarda girilen iki sayı kullanılarak yapılacaktır.  
1. sayıyı giriniz:20  
2. sayıyı giriniz:10
- İki sayı girildikten sonra ekrana aşağıdaki işlem menüsü gelecek ve yapılan girişe göre işlem yapılarak sonuç ekrana yazılacaktır.  
1- Toplama  
2- Çıkarma  
3- Çarpma  
4- Bölme  
İşlem türünü seçiniz (1,2,3,4) : 2  
  
İşlem sonucu = 10