



PROGRAMI I ORIENTUAR NE OBJEKTE

Introduction to Classes, Objects and
Strings

DEFINIMI I KLASËS ME ANËTARIN FUNKSIONIT



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

DEFINIMI I KLASËS ME ANËTARIN FUNKSIONIT



- ◉ Definimi i klasës **GradeBook** përmban anëtarin e funksionit të quajtur ***displayMessage*** që shfaq një mesazh në ekran.
- ◉ Duhet të krijojmë një objekt të klasës **GradeBook** dhe të thirrim anëtarin e funksionit ***displayMessage*** që të egzekutojë dhe shfaq mesazhin “***Welcome***”.

DEFINIMI I KLASËS ME ANËTARIN FUNKSIONIT



- ◉ Fjala **public**, është një specifikues i qasjes (access specifier). Rreshtat 10-14 definojnë anëtarin e funksionit *displayMessage*.
- ◉ Ky anëtar i funksionit shfaqet pas specifikuesit të qasjes **public**: për të treguar që funksioni është “në dispozicion për publikun” – që do të thotë, ai mund të thirret nga funksionet tjerë në program (siq është *main*), dhe nga anëtarët e funksioneve nga klasat tjera (nëse ka).
- ◉ Kur definojmë funksion, duhet të specifikojmë tipin e vlerës kthyese që të shfaq tipin e vlerës kthyese nga funksioni kur të kompletojë detyrat e tij.
- ◉ Në rreshtin 10, fjala **void** në të majtë të emrit të *displayMessage* është tipi kthyes i funksionit.
- ◉ Tipi kthyes void tregon se *displayMessage* nuk do të kthej të dhëna nga funksioni i thirrur kur ai të përfundojë me detyrat e tij.

DEFINIMI I KLASËS ME ANËTARIN FUNKSIONIT



- ◉ Emri i anëtarit të funksionit, ***displayMessage***, ndjek tipin kthyes.
- ◉ Nuk mund të thirret një anëtar i funksionit derisa të krijohet një objekt i asaj klase. Kompajleri nuk është në dijeni se çfarë tipi është **GradeBook** – është tip i përcaktuar nga useri apo shfrytëzuesi (GradeBook myGradeBook;)
- ◉ Ne i tregojmë kompajlerit çfarë është **GradeBook** duke përfshirë edhe përkufizimin e klasës (6-14).
- ◉ myGradeBook.displayMessage(); // call object's displayMessage function

DEFINIMI I ANËTARIT TË FUNKSIONIT ME PARAMETËR



```
6 | // GradeBook class definition
7 | class GradeBook
8 | {
9 |     public:
10 |         // function that displays a welcome message to the GradeBook user
11 |         void displayMessage( string courseName )
12 |         {
13 |             cout << "Welcome to the grade book for\n" << courseName
14 |                 << "!"
15 |                 << endl;
16 |         } // end function displayMessage
17 | }; // endclass GradeBook
18 | // function main begins program execution
19 | int main()
20 | {
21 |     string nameOfCourse; // string of characters to store the course name
22 |     GradeBook myGradeBook; // create a GradeBook object named myGradeBook
23 |     // prompt for and input course name
24 |     cout << "Please enter the course name:" << endl;
25 |     getline(cin, nameOfCourse); // read a course name with blanks
26 |     cout << endl; // output a blank line
27 |
28 |     // call myGradeBook's displayMessage function
29 |     // and pass nameOfCourse as an argument
30 |     myGradeBook.displayMessage(nameOfCourse);
31 |     cin.get();
32 |     cin.get();
33 |     return 0;
34 | } // end main
```

DEFINIMI I ANËTARIT TË FUNKSIONIT ME PARAMETËR



- ◉ Shembulli i ardhshëm ridefinon klasën **GradeBook** (rreshtat 7-16) me anëtarin e funksionit *displayMessage* (rreshtat 11-17) që tregon emrin e kursit si pjesë e mesazhit “welcome”. Verzioni i ri i *displayMessage* kërkon parametër (*courseName* në rreshtin 11) përfaqëson emrin e kursit në dalje apo si rezultat.
- ◉ Rreshti 25 lexon emrin nga shfrytëzuesi apo user-i dhe cakton atë në variablën *nameOfCourse*, duke shfrytëzuar librarinë e funksionit *getline* për të kryer apo egzekutuar daljen apo rezultatin.

DEFINIMI I ANËTARIT TË FUNKSIONIT ME PARAMETËR



- ◉ Rreshti 30 myGradeBook, thirr anëtarin e funksionit *displayMessage*.
- ◉ Variabla *nameOfCourse* në kllapa është argument që kalon në anëtarin e funksionit *displayMessage* kështu që ai mund të performojë apo kryejë detyrën e tij.
- ◉ Vlera e variablës *nameOfCourse* në *main* është kopjuar në anëtarin e funksionit *displayMessage* parametrin *courseName* në rreshtin 11.
- ◉ Kur ekzekutohet ky program, anëtari i funksionit *displayMessage* rezulton si pjesë e mesazhit “welcome” të emrit të kursit që e insertojmë (në rastin tonë, CS101 Introduction to C++ Programming).

DEFINIMI I ANËTARIT TË FUNKSIONIT ME PARAMETËR



- Rreshtat 13-14 paraqesin vlerën e parametrimit *courseName* si pjesë e mesazhit “welcome”. Emri i variablës së parametrimit (*courseName* në rreshtin 11) mund të jetë i njëjtë apo i ndryshëm nga argumenti i emrit të variablës (*nameOfCourse* në rreshtin 30).

ANËTARËT E TË DHËNAVE, FUNKSIONET SET DHE GET



```
8 // GradeBook classdefinition
9 class GradeBook
10 {
11     public:
12         // function that sets the course name
13         void setCourseName( string name )
14         {
15             courseName = name; // store the course name in the object
16
17         } // end function setCourseName
18         // function that gets the course name
19         string getCourseName()
20         {
21             return courseName; // return the object's courseName
22         } // end function getCourseName
23         // function that displays awelcome message
24         void displayMessage()
25         {
26             // thisstatement calls getCourseName to getthe
27             // nameofthe coursethisGradeBook represents
28             cout << "Welcome to the grade book for\n" << getCourseName() << "!"
29                 << endl;
30         } // end function displayMessage
31     private:
32         string courseName; // courseNamefor thisGradeBook
33 }; // end class GradeBook
```

ANËTARËT E TË DHËNAVE, FUNKSIONET SET DHE GET



```
35 // function main begins program execution
36 int main()
37 {
38     string nameOfCourse; // string of characters to store the course name
39     GradeBook myGradeBook; // create a GradeBook object named myGradeBook
40
41     // display initial value of courseName
42     cout << "Initial course name is: " << myGradeBook.getCourseName()
43         << endl;
44
45     // prompt for, input and set course name
46     cout << "\nPlease enter the course name:" << endl;
47     getline( cin, nameOfCourse ); // read a course name with blanks
48     myGradeBook.setCourseName(nameOfCourse); // set the course name
49
50     cout << endl;
51     myGradeBook.displayMessage();
52
53     cin.get();
54     cin.get();
55
56 }
```

ANËTARËT E TË DHËNAVE, FUNKSIONET SET DHE GET



- ◉ Klasa përmban anëtarë të funksionit *setCourseName*, *getCourseName* and *displayMessage*.
- ◉ Anëtari i funksionit *setCourseName* ruan emrin e kursit në **GradeBook** anëtarin e të dhënave (data member).
- ◉ Anëtari i funksionit *getCourseName* merr emrin e kursit nga anëtari i të dhënave.
- ◉ Anëtari i funksionit *displayMessage* –i cili tani nuk specifikon asnjë parametër – ende shfaq mesazhin “welcome” që përfshin emrin e kursit.
- ◉ Megjithatë, funksioni tani merr emrin e kursit duke thirr funksion tjetër në klasën e njejtë—*getCourseName*.

ANËTARËT E TË DHËNAVE, FUNKSIONET SET DHE GET



- ◉ Dobitë nga anëtari me të dhëna *courseName* janë se gjithë anëtarët e funksioneve të klasës mund të manipulojnë cilëndo anëtar me të dhëna që shfaqen në definimin e klasës (në këtë rast, *courseName*).
- ◉ Në përgjithësi, anëtarët e të dhënave duhet të deklarohen private dhe anëtarët e funksioneve duhet te deklarohen publike.

ANËTARËT E TË DHËNAVE, FUNKSIONET SET DHE GET



- ◉ Anëtari i funksionit **setCourseName** (rreshti 13-17) nuk kthen asnjë të dhënë kur përfundon me punë, kështu që tipi i vlerës kthyesë është void. Anëtari i funksionit merr një parametër – name – që paraqet emrin e kursit që do të kalohet si argument. (në rreshtin 48 të *main*).
- ◉ Anëtari i funksionit **getCourseName** (rreshtat 19-22) kthen objekt të veçantë të GradeBook, *courseName*. Anëtari i funksionit ka listën e parametrave bosh, kështuqë nuk kërkohen të dhëna shtesë që të kryhen detyrat. Funkzioni specifikon se ai kthen vlera string(karakterë). Kur thirret funksioni i cili specifikon tipin e vlerës kthyesë që është i ndryshëm nga void dhe përfundohen detyrat, funksioni përdor ngjarjen (rreshti 21) kthyesë për të kthyer rezultat të funksionit të tij thirrës.

ANËTARËT E TË DHËNAVE, FUNKSIONET SET DHE GET



- Anëtari i funksionit **displayMessage** (rreshtat 24-30) gjithashtu është me vlerë kthyese void. Rreshtat 28-29 si shfaq si rezultat mesazhin “welcome” që përfshin vlerën e anëtarit me të dhëna **courseName**.
- Klasat shpesh sigurojnë funksionet me të dhëna publike të lejojnë klientët e klasës të vendosin (caktojnë vlerat në) apo marrin (të marrë vlerat e) anëtarë me të dhëna private.

INICIALIZIMI I OBJEKTEVE ME KONSTRUKTORËT



```
7 | // GradeBook classdefinition
8 | class GradeBook
9 | {
10 | public:
11 |     GradeBook ( string name )
12 |     {
13 |         setCourseName( name );
14 |     } // end GradeBook constructor
15 |     // function to set the course name
16 |     void setCourseName(string name )
17 |     {
18 |         courseName =name; // store the course name in the object
19 |     } // end function setCourseName
20 |     // function to get the course name
21 |     string getCourseName()
22 |     {
23 |         return courseName; // return object's courseName
24 |     } // end function getCourseName
25 |     // displayawelcome message to the GradeBook user
26 |     void displayMessage()
27 |     {
28 |         // call getCourseName to get the courseName
29 |         cout << "Welcome to the grade book for\n" << getCourseName()
30 |             << "!" << endl;
31 |     } // end function displayMessage
32 | private:
33 |     string courseName; // course name for this GradeBook
34 | }; // end class GradeBook
```


INICIALIZIMI I OBJEKTEVE ME KONSTRUKTORËT



```
37 // function main begins program execution
38 int main()
39 {
40     // create two GradeBook objects
41
42     GradeBook gradeBook1( "CS101 Introduction to C++ Programming" );
43     GradeBook gradeBook2( "CS102 Data Structures in C++" );
44
45     // display initial value of courseName for each GradeBook
46     cout << "gradeBook1 created for course: " << gradeBook1.getCourseName()
47         << "\ngradeBook2 created for course: " << gradeBook2.getCourseName()
48         << endl;
49     cin.get();
50     cin.get();
51
52     return 0;
53 } // end main
54
```

INICIALIZIMI I OBJEKTEVE ME KONSTRUKTORËT



- ◉ Secila klasë që deklarohet mund të sigurojë konstruktorë që mund të përdoret për të inicializuar një objekt të klasës kur objekti është krijuar.
- ◉ Dallimi mes konstruktorit dhe funksioneve tjera është se konstruktorët nuk mund të kthejnë vlera.
- ◉ Zakonisht, constructorët janë të deklaruar si publik.

INICIALIZIMI I OBJEKTEVE ME KONSTRUKTORËT



- ◉ Rreshtat 11-14 definojnë një konstruktor për klasën GradeBook. Konstruktori specifikon në listën e parametrave të dhëna që kërkohen për të performuar detyrat e tij.
- ◉ Kur krijohet një objekt, këto të dhëna vendosen në kllapa që pason emri i objektit. (është realizuar në rreshtat 42-43).
- ◉ Rreshti 13 tregon se konstruktori i klasës GradeBook ka si parametër string të quajtur name. Rreshti 13 nuk specifikon vlerën kthyesë, sepse konstruktorët nuk mund të kthejnë vlera.



INICIALIZIMI I OBJEKTEVE ME KONSTRUKTORËT

- Rreshti 13 në trupin e konstruktorit pason emrin e parametrin të konstruktorit në anëtarin e funksionit ***setCourseName*** (rreshtat 16-19), i cili thjesht cakton vlerën e parametrin në anëtarin me të dhëna member ***courseName***.
- Konstruktori (rreshti 11) dhe funksioni ***setCourseName*** (rreshti 16) përdor parametrin të quajtur ***name***. Mund të përdoren emra të njëjtë të parametrave në funksione të ndryshëm, sepse parametrat janë lokal në seclin nga funksionet, ato nuk ndërhyjnë me njëra tjetrën.

VENDOSJA E NJË KLASE NË NJË FILE TË VEÇANTË PËR PËRDORIM



```
6 // GradeBook class definition
7 class GradeBook
8 {
9 public:
10 // constructor initializes courseName with string supplied as argument
11 GradeBook(string name )
12 {
13     setCourseName( name ); // call set function to initialize courseName
14
15 } // end GradeBook constructor
16
17 // function to set the course name
18 void setCourseName(string name )
19 {
20     courseName =name; // store the course name in the object
21 } // end function setCourseName
22 // function to get the course name
23 string getCourseName()
24 {
25     return courseName; // return object's courseName
26 } // end function getCourseName
27 // display a welcome message to the GradeBook user
28 void displayMessage()
29 { // call getCourseName to get the courseName
30     cout << "Welcome to the grade book for\n" << getCourseName() << "!" << endl;
31 } // end function displayMessage
32 private:
33     string courseName; // course name for this GradeBook
34 }; // end class GradeBook
```

VENDOSJA E NJË KLASE NË NJË FILE TË VEÇANTË PËR PËRDORIM

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

