



San Francisco Bay University

CS360 - Programming in C and C++ Homework Assignment #2

Due day: 2/27/2024

Kritika Regmi
ID:19702

1. Modify class *GradeBook* as follows:
 - a. Include a second-string data member that represents the course instructor's name.
 - b. Provide a *set* function to change the instructor's name and a *get* function to retrieve it.
 - c. Modify the constructor to specify course name and instructor name parameters.
 - d. Modify function *displayMessage* to output the welcome message and course name, then the string *"This course is presented by: "* followed by the instructor's name.

Use your modified class in main function that demonstrates the class's new capabilities.

ANSWER:

OUTPUT:

```
>_ Console  Shell  x  +
~/CS3602$ ls
main Makefile num1 replit.nix
~/CS3602$ cd num1/
~/CS3602/num1$ ls
main.cpp
~/CS3602/num1$ g++ main.cpp -o output1
~/CS3602/num1$ ./output1
Welcome to the grade book for
CS101 Introduction to C++ Programming!
This course is presented by: Jesse Lopez
~/CS3602/num1$
```

CODE:

```
num1 > C++ main.cpp > ...
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  class GradeBook{
6  public:
7      explicit GradeBook(string, string ); // constructor initialize courseName
8      void setCourseName(string ); // sets the course name
9      string getCourseName() const; // gets the course name
10
11     void setCourseInstructor(string); //sets the insctructor name
12     string getCourseInstructor() const; //gets the instructor name
13
14     void displayMessage() const; // displays a welcome message
15
16 private:
17     std::string courseName; // course name for this GradeBook
18     std::string courseInstructor;
19 }; // end class GradeBook
```

```
21 GradeBook::GradeBook( string name, string Instructorname ):courseName( name
    ),courseInstructor( Instructorname) {}
22
23 void GradeBook::setCourseName( string name ){
24     courseName = name;
25 }
26
27 void GradeBook::setCourseInstructor(string Instructorname){
28     courseInstructor = Instructorname;
29 }
30
31 string GradeBook::getCourseName() const{return courseName;}
32 string GradeBook::getCourseInstructor() const{return courseInstructor;}
33
34 void GradeBook::displayMessage() const{
35     cout << "Welcome to the grade book for\n" << getCourseName() << "!" << endl;
36     cout<< "This course is presented by: " << getCourseInstructor() << endl;
37 }
38
39 int main(){
40     GradeBook gradeBook1( "CS101 Introduction to C++ Programming", "Jesse Lopez");
41     gradeBook1.displayMessage(); // Call to display welcome message
42     return 0;
43 }
```

2. Create a class called *Date* that includes three pieces of information as data members--a month (type *int*), a day (type *int*) and a year (type *int*). Your class should have a constructor with three parameters that uses the parameters to initialize the three data members. Assume that the values provided for the year and day are correct but ensure that the month value is in the range *1-12*; if it isn't, set the month to *1*. Provide a *set* and a *get* function for each data member. Provide a member function *displayDate* that displays the month, day and year separated by forward slashes (/). Write a test program that demonstrates class *Date*'s capabilities.

ANSWER:

OUTPUT:

```
>_ Console Shell x +
~/CS3602/num2$ g++ num2.cpp -o output2
~/CS3602/num2$ ./output2
Today's date is: 2/27/2024
Invalid date is: 1/45/2025
~/CS3602/num2$
```

CODE:

```
main.cpp num2.cpp × +
num2 > num2.cpp > f main

1  #include <iostream>
2  using namespace std;
3
4  class Date {
5  private:
6      int month;
7      int day;
8      int year;
9
10 public:
11     Date(int m, int d, int y) {
12         setMonth(m);
13         setDay(d);
14         year = y;
15     }
16
17     void setMonth(int m) {
18         if (m >= 1 && m <= 12)
19             month = m;
20         else
21             month = 1;
22     }
23
24     int getMonth() const {
25         return month;
26     }
```

```

28 v void setDay(int d) {
29     day = d;
30 }
31
32 v int getDay() const {
33     return day;
34 }
35
36 v void setYear(int y) {
37     year = y;
38 }
39
40 v int getYear() const {
41     return year;
42 }
43
44 v void displayDate() const {
45     cout << month << "/" << day << "/" << year << endl;
46 }
47 };

```

```

49 v int main() {
50     Date today(02, 27, 2024);
51     cout << "Today's date is: ";
52     today.displayDate();
53
54     Date invalidDate(35, 45, 2025);
55     cout << "Invalid date is: ";
56     invalidDate.displayDate();
57
58     return 0;
59 }

```

3. While exercising, you can use a heart rate monitor to see that your heart rate stays within a safe range suggested by your trainers and doctors. According to the American Heart Association (AHA) (www.americanheart.org/presenter.jhtml?identifier=4736), the formula for calculating your *maximum heart rate* in beats per minute is 220 minus your age in years. Your *target heart rate* is a range that is 50-85% of your maximum heart rate. [Note: *These formulas are estimates provided by the AHA. Maximum and target heart rates may vary based on the health, fitness and gender of the individual. Always consult a physician or qualified health care professional before beginning or modifying an exercise program.*]. Create a class called *HeartRates*. The class attributes should include the person's first name, last name and date of birth (consisting of separate attributes for the month, day and year of birth). Your class should have a constructor that receives this data as parameters. For each attribute provide *set* and *get* functions. The class also should include a function *getAge* that calculates and returns the person's age (in years), a function *getMaximumHeartRate* that calculates and returns the person's maximum heart rate and a function *getTargetHeartRate* that calculates and returns the person's target heart rate. Since you do not yet know how to obtain the current date from the computer, function *getAge* should prompt the user to enter the current month, day and year before calculating the person's age. Write an application that prompts for the person's information, instantiates an object of class *HeartRates* and prints the information from that object—including the person's first name, last name and date of birth—then calculates and prints the person's age in (years), maximum heart rate and target-heart-rate range.

ANSWER:

OUTPUT:

```
>_ Console  Shell  x  +
~/CS3602/num3$ ls
num3.cpp
~/CS3602/num3$ g++ num3.cpp -o output3
~/CS3602/num3$ ./output3
Enter your first name: Kritika
Enter your last name: Regmi
Enter your birth month (1-12): 12
Enter your birth day (1-31): 30
Enter your birth year: 2003

Name: Kritika Regmi
Date of Birth: 12/30/2003
Age: 20 years
Maximum Heart Rate: 200 beats per minute
Target Heart Rate Range: 100 - 170 beats per minute
~/CS3602/num3$
```

CODE:

```
main.cpp  num2.cpp  num3.cpp  ×  +
num3 > num3.cpp
1  #include <iostream>
2  #include <string>
3  #include <ctime> // for current date
4  using namespace std;
5
6  class HeartRates {
7  private:
8      string firstName;
9      string lastName;
10     int birthMonth;
11     int birthDay;
12     int birthYear;
13
14 public:
15     HeartRates(string first, string last, int month, int day, int year)
16         : firstName(first), lastName(last), birthMonth(month), birthDay(day), birthYear(year) {}
17
18     // Setters
19     void setFirstName(string first) { firstName = first; }
20     void setLastName(string last) { lastName = last; }
21     void setBirthMonth(int month) { birthMonth = month; }
22     void setBirthDay(int day) { birthDay = day; }
23     void setBirthYear(int year) { birthYear = year; }
24
25     // Getters
26     string getFirstName() const { return firstName; }
27     string getLastName() const { return lastName; }
28     int getBirthMonth() const { return birthMonth; }
29     int getBirthDay() const { return birthDay; }
30     int getBirthYear() const { return birthYear; }
31
32     // Calculate age
33     int getAge() const {
34         time_t now = time(0);
35         tm* ltm = localtime(&now);
36         int currentYear = 1900 + ltm->tm_year;
37         int age = currentYear - birthYear;
38         if (birthMonth > ltm->tm_mon + 1 || (birthMonth == ltm->tm_mon + 1 && birthDay > ltm->tm_mday)) {
39             age--;
40         }
41         return age;
42     }
43
44     // Calculate maximum heart rate
45     int getMaximumHeartRate() const {
46         return 220 - getAge();
47     }
48 }
```

```

49 // Calculate target heart rate range
50 pair<int, int> getTargetHeartRate() const {
51     int maxHeartRate = getMaximumHeartRate();
52     return make_pair(0.5 * maxHeartRate, 0.85 * maxHeartRate);
53 }
54 };
55
56 int main() {
57     string firstName, lastName;
58     int birthMonth, birthDay, birthYear;
59
60     cout << "Enter your first name: ";
61     cin >> firstName;
62     cout << "Enter your last name: ";
63     cin >> lastName;
64     cout << "Enter your birth month (1-12): ";
65     cin >> birthMonth;
66     cout << "Enter your birth day (1-31): ";
67     cin >> birthDay;
68     cout << "Enter your birth year: ";
69     cin >> birthYear;
70

```

```

71     HeartRates person(firstName, lastName, birthMonth, birthDay, birthYear);
72
73     cout << "\nName: " << person.getFirstName() << " " << person.getLastName() << endl;
74     cout << "Date of Birth: " << person.getBirthMonth() << "/" << person.getBirthDay() << "/" <<
    person.getBirthYear() << endl;
75     cout << "Age: " << person.getAge() << " years" << endl;
76     cout << "Maximum Heart Rate: " << person.getMaximumHeartRate() << " beats per minute" << endl;
77
78     pair<int, int> targetHeartRate = person.getTargetHeartRate();
79     cout << "Target Heart Rate Range: " << targetHeartRate.first << " - " << targetHeartRate.second << " beats per
    minute" << endl;
80
81     return 0;
82 }
83

```