Question no.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 display Message 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 the main function that demonstrates the class's new capabilities.

```
#include <string>
class GradeBook{
public:
explicit GradeBook( std::string ); // constructor initialize
courseName
void setCourseName( std::string ); // sets the course name
std::string getCourseName() const; // gets the course name
void displayMessage() const; // displays a welcome message
private:
std::string courseName; // course name for this GradeBook
}; // end class GradeBook
#include <iostream>
using namespace std;
GradeBook::GradeBook( string name ):courseName( name ){}
void GradeBook::setCourseName( string name ){
courseName = name;
}
string GradeBook::getCourseName() const{return courseName;}
void GradeBook::displayMessage() const{
cout << "Welcome to the grade book for\n" << getCourseName()
```

<< "!" << endl;
}

⇨ Answer:
The code:

Q1.cpp › ... Format

```
23      courseName = name;
24  }
25
26  void GradeBook::setInstructorName(string name) {
27      instructorName = name;
28  }
29
30  string GradeBook::getCourseName() const {
31      return courseName;
32  }
33
34  string GradeBook::getInstructorName() const {
35      return instructorName;
36  }
37
38  void GradeBook::displayMessage() const {
39      cout << "Welcome to the grade book for\n" << getCourseName() <<
    "!" << endl;
40      cout << "This course is presented by: " << getInstructorName() <<
    endl;
41  }
42
43  int main() {
44      GradeBook myGradeBook("Mathematics", "Mr. Smith");
45      myGradeBook.displayMessage();
46      return 0;
47  }
48
```

Generate ⌘ I

```cpp
#include <iostream>
#include <string>

using namespace std;

class GradeBook {
public:
    explicit GradeBook(string, string); // constructor initialize courseName and instructorName
    void setCourseName(string); // sets the course name
    void setInstructorName(string); // sets the instructor's name
    string getCourseName() const; // gets the course name
    string getInstructorName() const; // gets the instructor's name
    void displayMessage() const; // displays a welcome message
private:
    string courseName; // course name for this GradeBook
    string instructorName; // instructor's name for this GradeBook
};

GradeBook::GradeBook(string course, string instructor)
    : courseName(course), instructorName(instructor) {}

void GradeBook::setCourseName(string name) {
    courseName = name;
}

void GradeBook::setInstructorName(string name) {
    instructorName = name;
}

```

The output:

```
~/lab2$ g++ Q1.cpp -o Q1_output
~/lab2$ ./Q1_output
Welcome to the grade book for
Mathematics!
This course is presented by: Mr. Smith
```

Question no.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.

⇨ The answer:

The code:

```cpp
#include <iostream>

class Date {
public:
    Date(int m, int d, int y); // constructor with three parameters
    void setMonth(int m); // set month
    int getMonth() const; // get month
    void setDay(int d); // set day
    int getDay() const; // get day
    void setYear(int y); // set year
    int getYear() const; // get year
    void displayDate() const; // display date in the format mm/dd/yyyy
private:
    int month; // month for the date
    int day; // day for the date
    int year; // year for the date
};

Date::Date(int m, int d, int y) {
    setMonth(m);
    setDay(d);
    setYear(y);
}

void Date::setMonth(int m) {
    if (m >= 1 && m <= 12)
        month = m;
    else
        month = 1;
}
```

```
}

int Date::getMonth() const {
    return month;
}

void Date::setDay(int d) {
    day = d;
}

int Date::getDay() const {
    return day;
}

void Date::setYear(int y) {
    year = y;
}

int Date::getYear() const {
    return year;
}

void Date::displayDate() const {
    std::cout << month << "/" << day << "/" << year << std::endl;
}

int main() {
    // Creating a Date object and testing its functionality
    Date myDate(2, 29, 2024); // February 29, 2024 (leap year)
    std::cout << "Date: ";
```

```
    std::cout << "Date: ";
    myDate.displayDate(); // Display the date

    // Testing setting and getting individual components of the date
    myDate.setMonth(12);
    myDate.setDay(25);
    myDate.setYear(2025);
    std::cout << "Updated Date: ";
    myDate.displayDate(); // Display the updated date

    return 0;
}
```

Generate ⌘ I

The output:

```
~/lab2$ g++ Q2.cpp -o Q2_output
~/lab2$ ./Q2_output
Date: 2/29/2024
Updated Date: 12/25/2025
```

Question no.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 getMaxiumumHeartRate 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

The code:

```cpp
#include <iostream>
#include <string>
#include <cmath>

using namespace std;

class HeartRates {
public:
    HeartRates(string firstName, string lastName, int birthMonth, int birthDay, int birthYear);
    void setFirstName(string firstName);
    string getFirstName() const;
    void setLastName(string lastName);
    string getLastName() const;
    void setBirthMonth(int birthMonth);
    int getBirthMonth() const;
    void setBirthDay(int birthDay);
    int getBirthDay() const;
    void setBirthYear(int birthYear);
    int getBirthYear() const;
    int getAge() const;
    int getMaximumHeartRate() const;
    pair<int, int> getTargetHeartRate() const;
private:
    string firstName;
    string lastName;
    int birthMonth;
    int birthDay;
    int birthYear;
```

```
25      string lastName;
26      int birthMonth;
27      int birthDay;
28      int birthYear;
29  };
30
31  HeartRates::HeartRates(string firstName, string lastName, int
    birthMonth, int birthDay, int birthYear)
32      : firstName(firstName), lastName(lastName),
    birthMonth(birthMonth), birthDay(birthDay), birthYear(birthYear) {}
33
34  void HeartRates::setFirstName(string firstName) {
35      this->firstName = firstName;
36  }
37
38  string HeartRates::getFirstName() const {
39      return firstName;
40  }
41
42  void HeartRates::setLastName(string lastName) {
43      this->lastName = lastName;
44  }
45
46  string HeartRates::getLastName() const {
47      return lastName;
48  }
49
50  void HeartRates::setBirthMonth(int birthMonth) {
51      this->birthMonth = birthMonth;
52  }
```

```
52  }
53
54  int HeartRates::getBirthMonth() const {
55      return birthMonth;
56  }
57
58  void HeartRates::setBirthDay(int birthDay) {
59      this->birthDay = birthDay;
60  }
61
62  int HeartRates::getBirthDay() const {
63      return birthDay;
64  }
65
66  void HeartRates::setBirthYear(int birthYear) {
67      this->birthYear = birthYear;
68  }
69
70  int HeartRates::getBirthYear() const {
71      return birthYear;
72  }
73
74  int HeartRates::getAge() const {
75      int currentMonth, currentDay, currentYear;
76      cout << "Enter the current date (mm dd yyyy): ";
77      cin >> currentMonth >> currentDay >> currentYear;
78
79      int age = currentYear - birthYear;
80      if (currentMonth < birthMonth || (currentMonth == birthMonth &&
    currentDay < birthDay)) {
```

```
80      if (currentMonth < birthMonth || (currentMonth == birthMonth &&
    currentDay < birthDay)) {
81          age--;
82      }
83      return age;
84  }
85
86  int HeartRates::getMaximumHeartRate() const {
87      int age = getAge();
88      return 220 - age;
89  }
90
91  pair<int, int> HeartRates::getTargetHeartRate() const {
92      int maxHeartRate = getMaximumHeartRate();
93      int lowerRange = round(0.5 * maxHeartRate);
94      int upperRange = round(0.85 * maxHeartRate);
95      return make_pair(lowerRange, upperRange);
96  }
97
98  int main() {
99      string firstName, lastName;
100     int birthMonth, birthDay, birthYear;
101
102     cout << "Enter person's information:" << endl;
103     cout << "First Name: ";
104     cin >> firstName;
105     cout << "Last Name: ";
106     cin >> lastName;
107     cout << "Date of Birth (mm dd yyyy): ";
108     cin >> birthMonth >> birthDay >> birthYear;
```

Q3.cpp

```
    cout << "Date of Birth (mm dd yyyy): ";
    cin >> birthMonth >> birthDay >> birthYear;

    HeartRates person(firstName, lastName, birthMonth, birthDay, birthYear);

    cout << "\nPerson's Information:" << endl;
    cout << "First Name: " << person.getFirstName() << endl;
    cout << "Last Name: " << person.getLastName() << endl;
    cout << "Date of Birth: " << person.getBirthMonth() << "/" << person.getBirthDay() << "/" << person.getBirthYear() << endl;

    int age = person.getAge();
    cout << "Age: " << age << " years" << endl;

    int maxHeartRate = person.getMaximumHeartRate();
    cout << "Maximum Heart Rate: " << maxHeartRate << " beats per minute" << endl;

    pair<int, int> targetHeartRate = person.getTargetHeartRate();
    cout << "Target Heart Rate Range: " << targetHeartRate.first << " - " << targetHeartRate.second << " beats per minute" << endl;

    return 0;
}
```

The output:

```
~/lab2$ g++ Q3.cpp -o Q3_output
~/lab2$ ./Q3_output
Enter person's information:
First Name: Yunisha
Last Name: Basnet
Date of Birth (mm dd yyyy): 04/26/2003

Person's Information:
First Name: Yunisha
Last Name: Basnet
Date of Birth: 4/0/32543
Enter the current date (mm dd yyyy): Age: 0 years
Enter the current date (mm dd yyyy): Maximum Heart Rate: 220 bea
ts per minute
Enter the current date (mm dd yyyy): Target Heart Rate Range: 11
0 - 187 beats per minute
~/lab2$
```