Question no. 1

Code:

#include <string>

#include <iostream>

class GradeBook {

public:

// constructor initializes courseName and instructorName

explicit GradeBook(std::string course, std::string instructor)

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

// sets the course name

void setCourseName(std::string name) {

courseName = name;

}

// gets the course name

std::string getCourseName() const {

return courseName;

}

// sets the instructor name

void setInstructorName(std::string name) {

instructorName = name;

}

// gets the instructor name

std::string getInstructorName() const {

return instructorName;

}

// displays a welcome message

void displayMessage() const {

std::cout << "Welcome to the grade book for\n" << getCourseName()

<< "!\nThis course is presented by: " << getInstructorName() << std::endl;

}

private:

std::string courseName; // course name for this GradeBook

std::string instructorName; // instructor name for this GradeBook

}; // end class GradeBook

int main() {

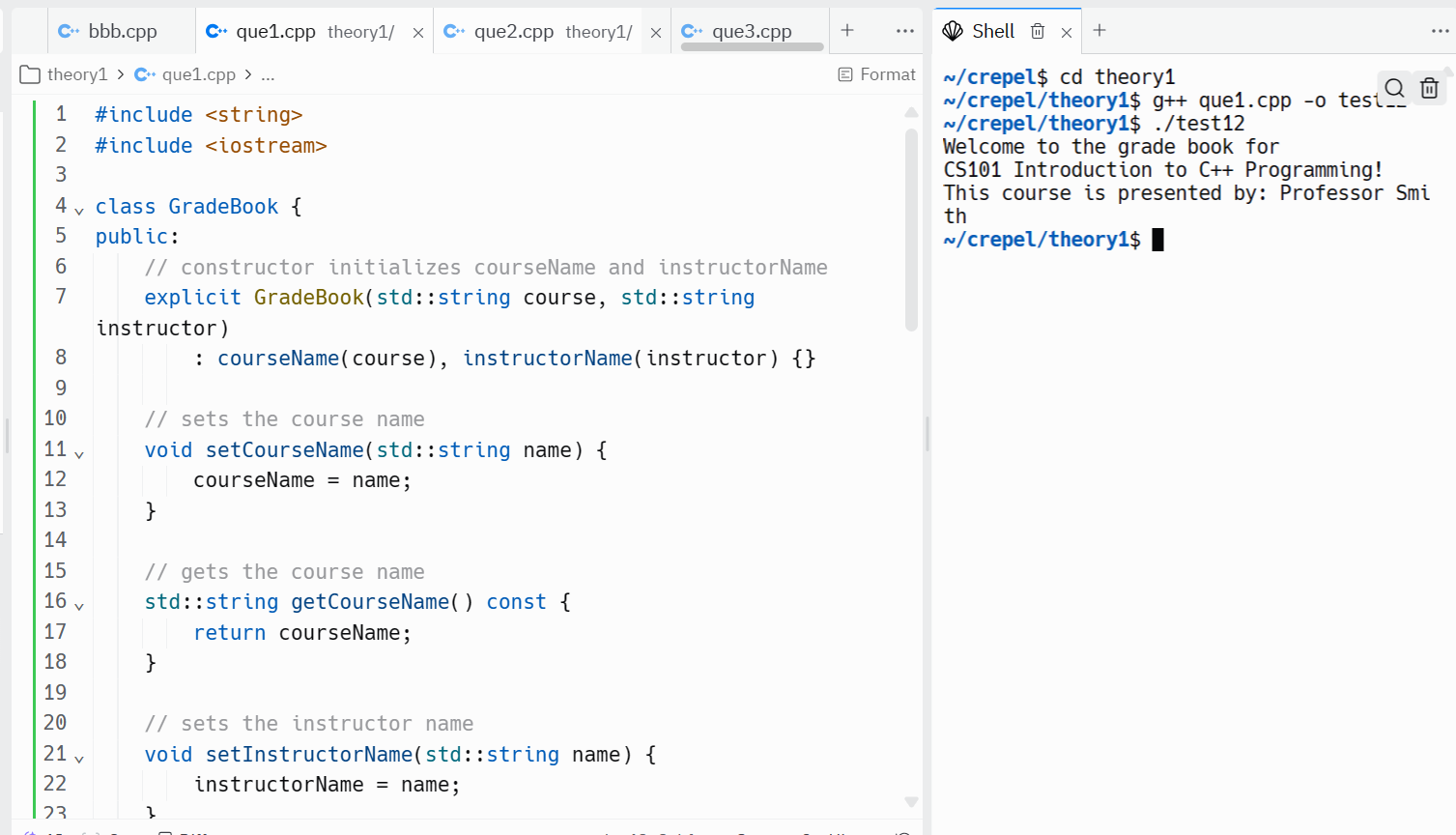
// Create a GradeBook object and demonstrate its new capabilities

GradeBook myGradeBook("CS101 Introduction to C++ Programming", "Professor Smith");

myGradeBook.displayMessage();

return 0;

}



Que 2

Code:

#include <iostream>

class Date {

public:

// Constructor that uses parameters to initialize the data members

Date(int month, int day, int year)

: day(day), year(year) {

setMonth(month);

}

// Set and get functions for each data member

void setMonth(int month) {

if(month >= 1 && month <= 12) {

this->month = month;

} else {

this->month = 1;

}

}

int getMonth() const {

return month;

}

void setDay(int day) {

this->day = day;

}

int getDay() const {

return day;

}

void setYear(int year) {

this->year = year;

}

int getYear() const {

return year;

}

// Member function that displays the date

void displayDate() const {

std::cout << month << "/" << day << "/" << year << std::endl;

}

private:

int month; // month for this Date

int day; // day for this Date

int year; // year for this Date

}; // end class Date

int main() {

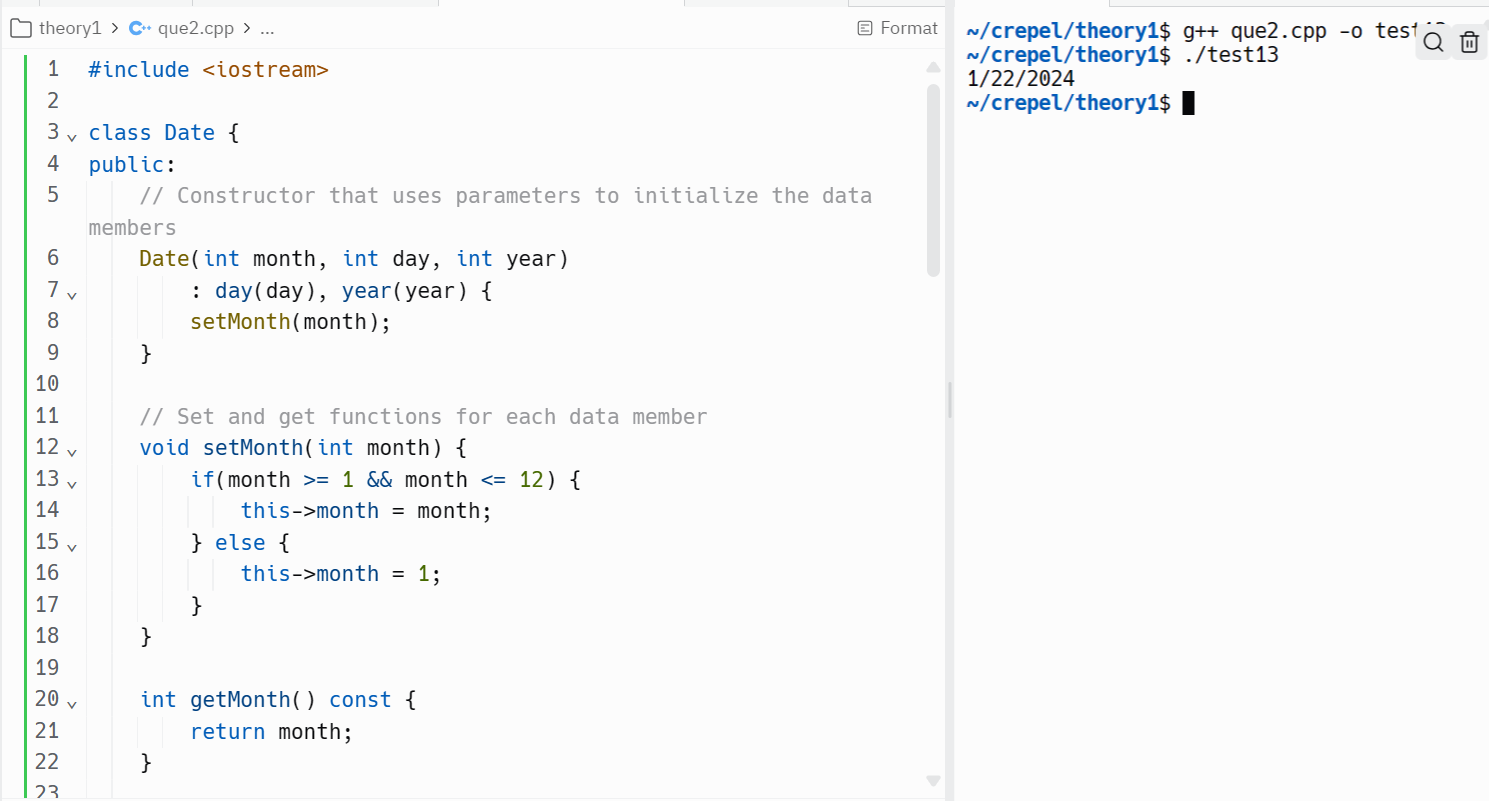
// Create a Date object and demonstrate its capabilities

Date myDate(13, 22, 2024); // month is out of range, so it will be set to 1

myDate.displayDate();

return 0;

}



Que3

Code:

#include <iostream>

#include <string>

class HeartRates {

public:

// Constructor that uses parameters to initialize the data members

HeartRates(std::string firstName, std::string lastName, int birthMonth,

int birthDay, int birthYear)

: firstName(firstName), lastName(lastName), birthMonth(birthMonth),

birthDay(birthDay), birthYear(birthYear) {}

// Set and get functions for each data member

void setFirstName(std::string name) { firstName = name; }

std::string getFirstName() const { return firstName; }

void setLastName(std::string name) { lastName = name; }

std::string getLastName() const { return lastName; }

void setBirthMonth(int month) { birthMonth = month; }

int getBirthMonth() const { return birthMonth; }

void setBirthDay(int day) { birthDay = day; }

int getBirthDay() const { return birthDay; }

void setBirthYear(int year) { birthYear = year; }

int getBirthYear() const { return birthYear; }

// Function to calculate and return the person's age

int getAge(int currentYear) const { return currentYear - birthYear; }

// Function to calculate and return the person's maximum heart rate

int getMaximumHeartRate(int currentYear) const {

return 220 - getAge(currentYear);

}

// Function to calculate and return the person's target heart rate

std::pair<int, int> getTargetHeartRate(int currentYear) const {

int maxRate = getMaximumHeartRate(currentYear);

return std::make\_pair(maxRate \* 0.5, maxRate \* 0.85);

}

private:

std::string firstName; // first name for this HeartRates

std::string lastName; // last name for this HeartRates

int birthMonth; // birth month for this HeartRates

int birthDay; // birth day for this HeartRates

int birthYear; // birth year for this HeartRates

}; // end class HeartRates

int main() {

// Prompt for the person's information

std::string firstName, lastName;

int birthMonth, birthDay, birthYear, currentYear;

std::cout << "Enter your first name: ";

std::cin >> firstName;

std::cout << "Enter your last name: ";

std::cin >> lastName;

std::cout << "Enter your birth month (1-12): ";

std::cin >> birthMonth;

std::cout << "Enter your birth day (1-31): ";

std::cin >> birthDay;

std::cout << "Enter your birth year: ";

std::cin >> birthYear;

std::cout << "Enter the current year: ";

std::cin >> currentYear;

// Instantiate an object of class HeartRates

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

// Print the information from the object

std::cout << "First Name: " << myHeartRates.getFirstName() << std::endl;

std::cout << "Last Name: " << myHeartRates.getLastName() << std::endl;

std::cout << "Date of Birth: " << myHeartRates.getBirthMonth() << "/"

<< myHeartRates.getBirthDay() << "/" << myHeartRates.getBirthYear()

<< std::endl;

// Calculate and print the person's age, maximum heart rate, and target heart

// rate range

std::cout << "Age: " << myHeartRates.getAge(currentYear) << " years"

<< std::endl;

std::cout << "Maximum Heart Rate: "

<< myHeartRates.getMaximumHeartRate(currentYear) << " bpm"

<< std::endl;

std::pair<int, int> targetRange =

myHeartRates.getTargetHeartRate(currentYear);

std::cout << "Target Heart Rate Range: " << targetRange.first << " - "

<< targetRange.second << " bpm" << std::endl;

return 0;

}

