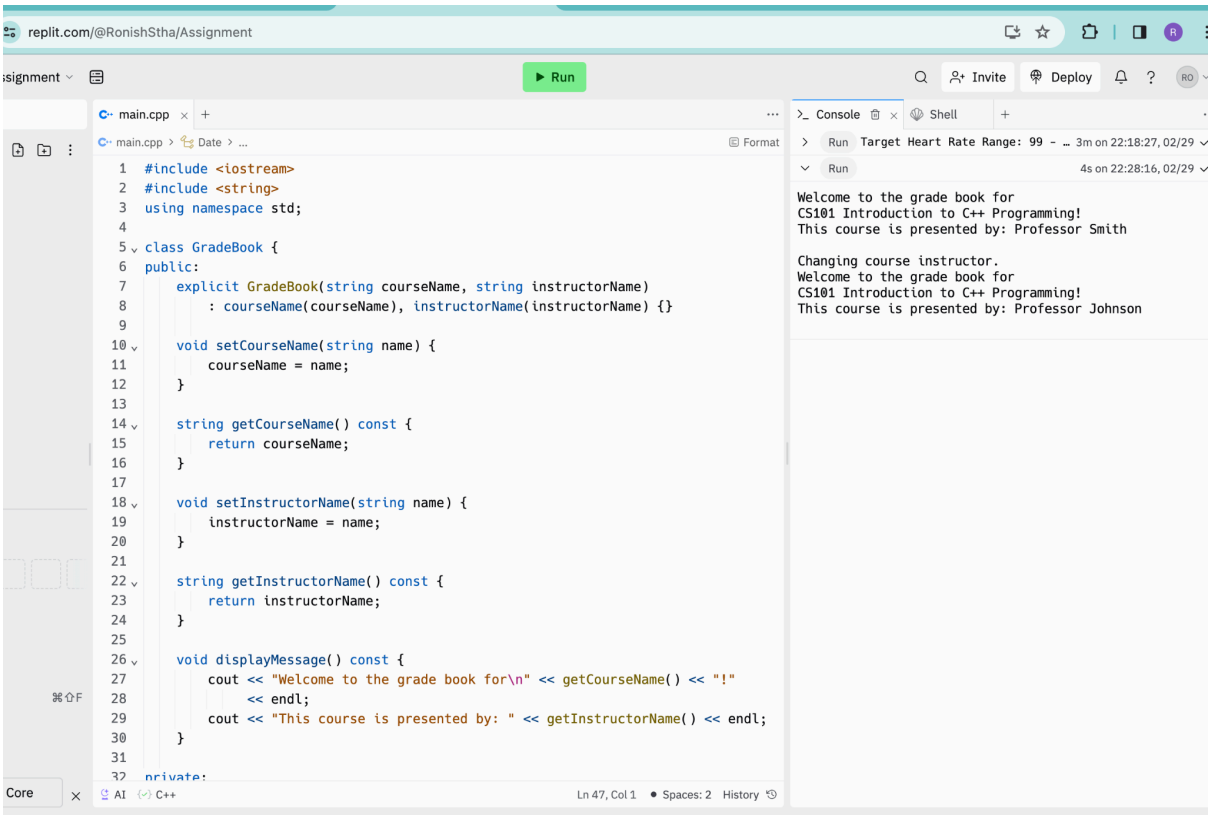
#Question-no(1):

```
#include <iostream>
#include <string>
using namespace std;

class GradeBook {
public:
   explicit GradeBook(string courseName, string instructorName)
      : courseName(courseName), instructorName(instructorName) {}

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

   string getCourseName() const {
      return courseName;
   }

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

   string getInstructorName() const {
      return instructorName;
   }

   void displayMessage() const {
      cout << "Welcome to the grade book for\n" << getCourseName() << "!"
          << endl;
      cout << "This course is presented by: " << getInstructorName() << endl;
   }

private:
   string courseName;
   string instructorName;
}; // end class GradeBook

int main() {
   GradeBook myGradeBook("CS101 Introduction to C++ Programming", "Professor Smith");
   myGradeBook.displayMessage();

   // Demonstrating changing the instructor's name
   myGradeBook.setInstructorName("Professor Johnson");
   cout << "\nChanging course instructor.\n";
   myGradeBook.displayMessage();
   return 0;
```

}



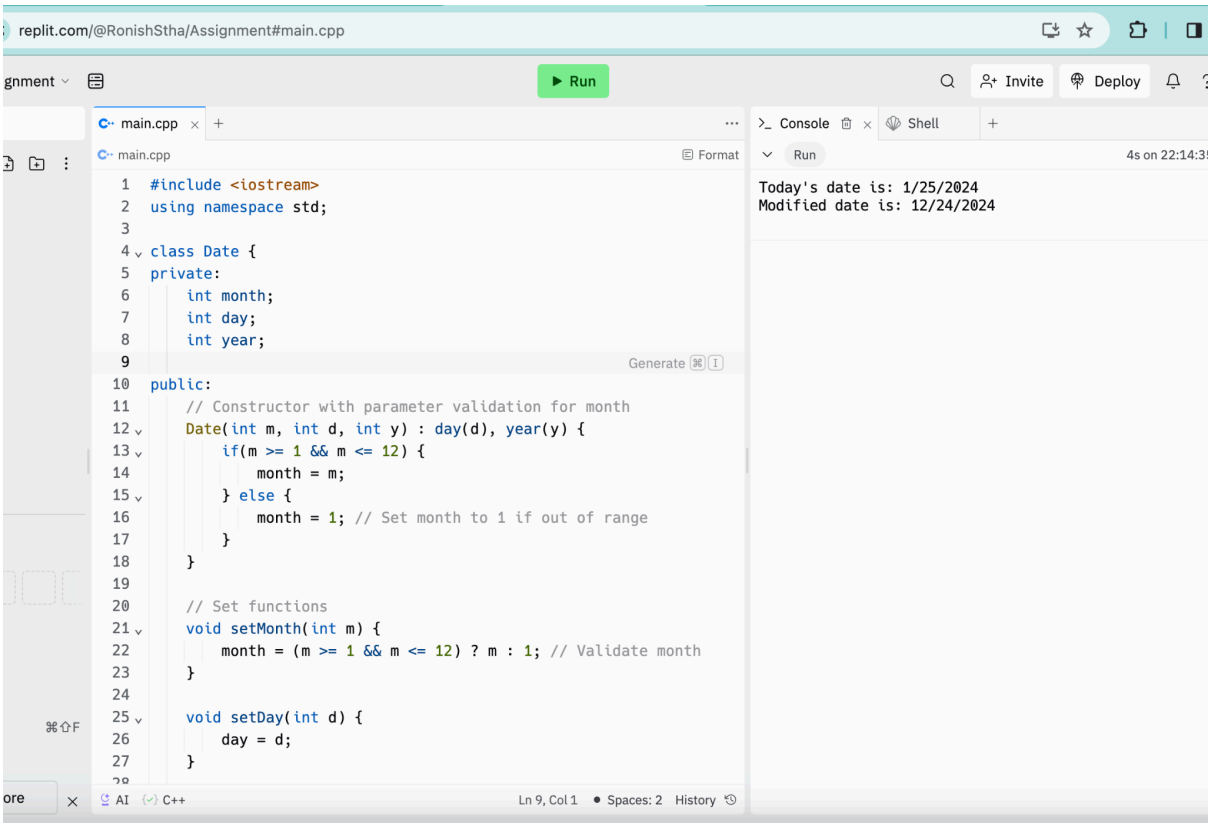

```cpp
    }

private:
    string courseName;
    string instructorName;
}; // end class GradeBook

int main() {
    GradeBook myGradeBook("CS101 Introduction to C++ Programming", "Professor Smith");
    myGradeBook.displayMessage();

    // Demonstrating changing the instructor's name
    myGradeBook.setInstructorName("Professor Johnson");
    cout << "\nChanging course instructor.\n";
    myGradeBook.displayMessage();
    return 0;
}
```

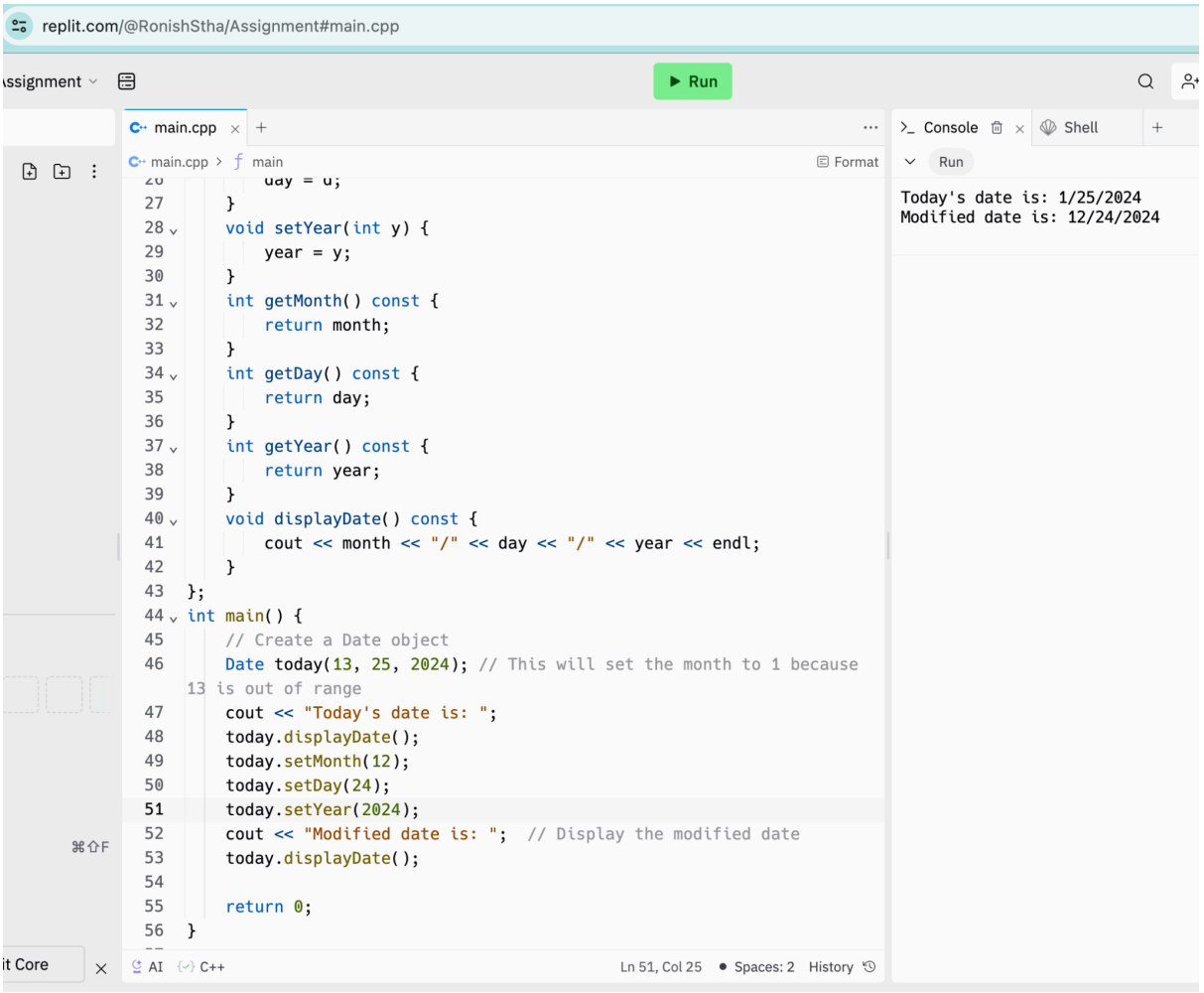
Generate ⌘ I

#Question-no(2):

```
#include <iostream>
using namespace std;

class Date {
private:
    int month;
    int day;
    int year;

public:
    // Constructor with parameter validation for month
    Date(int m, int d, int y) : day(d), year(y) {
        if(m >= 1 && m <= 12) {
            month = m;
        } else {
            month = 1; // Set month to 1 if out of range
        }
    }

    // Set functions
    void setMonth(int m) {
        month = (m >= 1 && m <= 12) ? m : 1; // Validate month
    }

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

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

    // Get functions
    int getMonth() const {
        return month;
    }

    int getDay() const {
        return day;
    }

    int getYear() const {
        return year;
    }
```

```
    // displayDate function
    void displayDate() const {
        cout << month << "/" << day << "/" << year << endl;
    }
};

int main() {
    // Create a Date object
    Date today(13, 25, 2024); // This will set the month to 1 because 13 is out of range

    // Display the date
    cout << "Today's date is: ";
    today.displayDate();

    // Modify the date using set functions
    today.setMonth(12);
    today.setDay(24);
    today.setYear(2024);

    // Display the modified date
    cout << "Modified date is: ";
    today.displayDate();

    return 0;
}
```





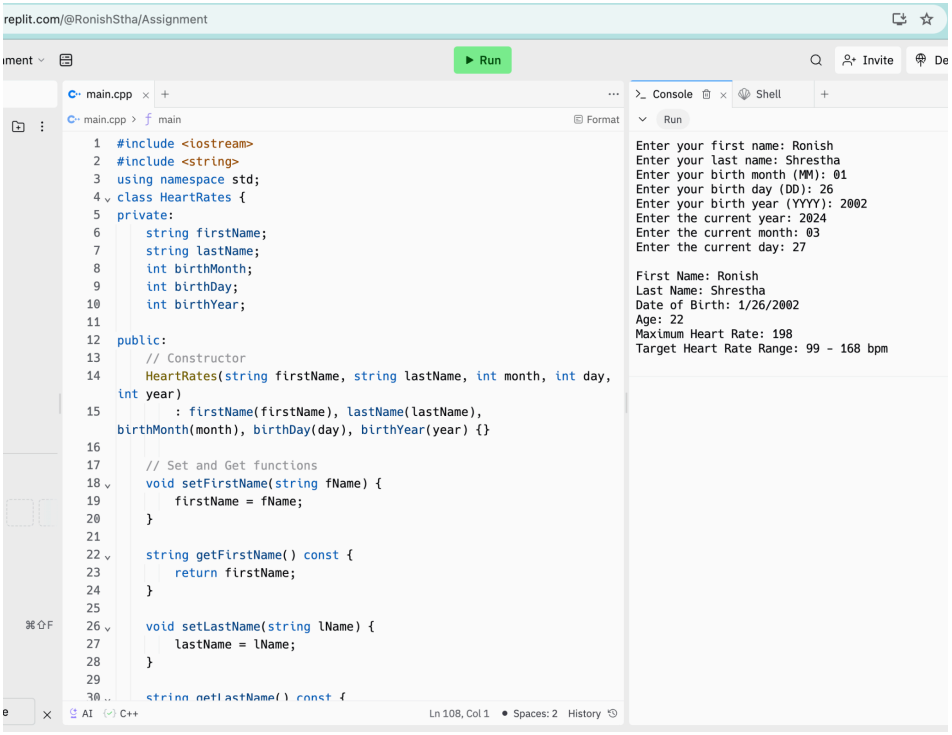
#Solution-3:

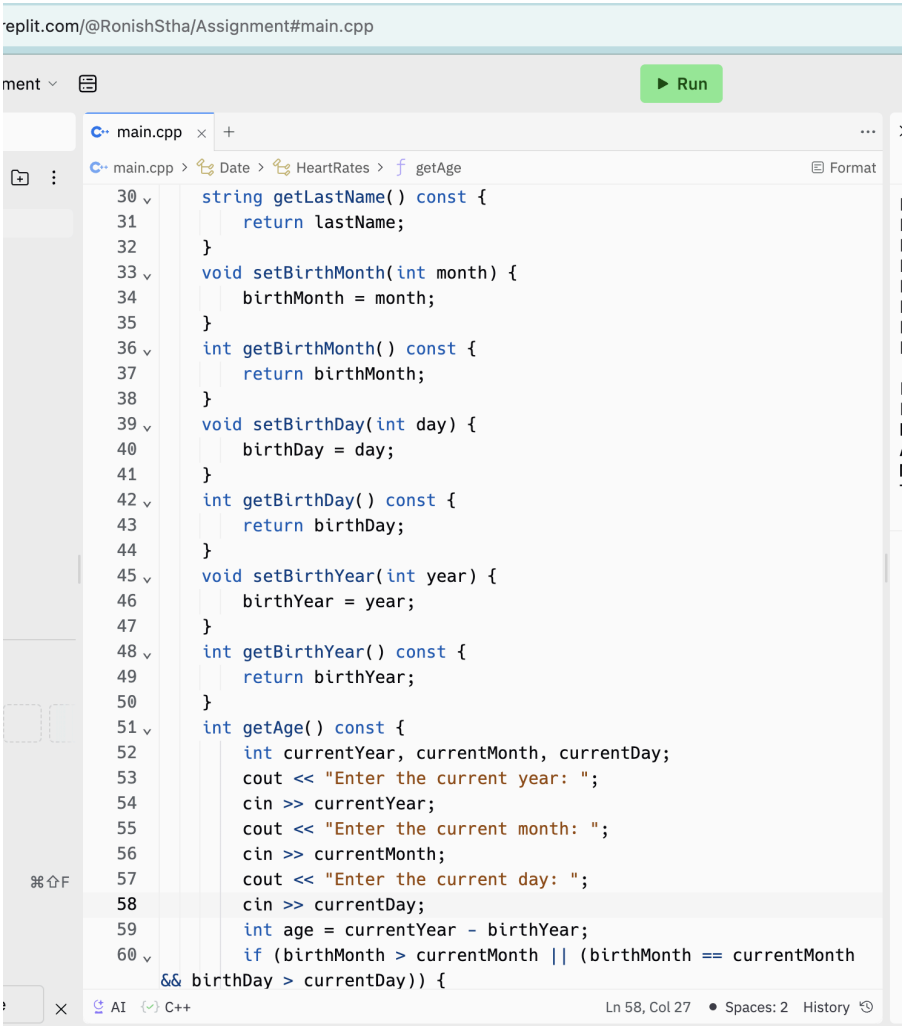
```
#include <iostream>
#include <string>
using namespace std;
class HeartRates {
private:
    string firstName;
    string lastName;
    int birthMonth;
    int birthDay;
    int birthYear;

public:
    // Constructor
    HeartRates(string firstName, string lastName, int month, int day, int year)
        : firstName(firstName), lastName(lastName), birthMonth(month), birthDay(day),
birthYear(year) {}
```

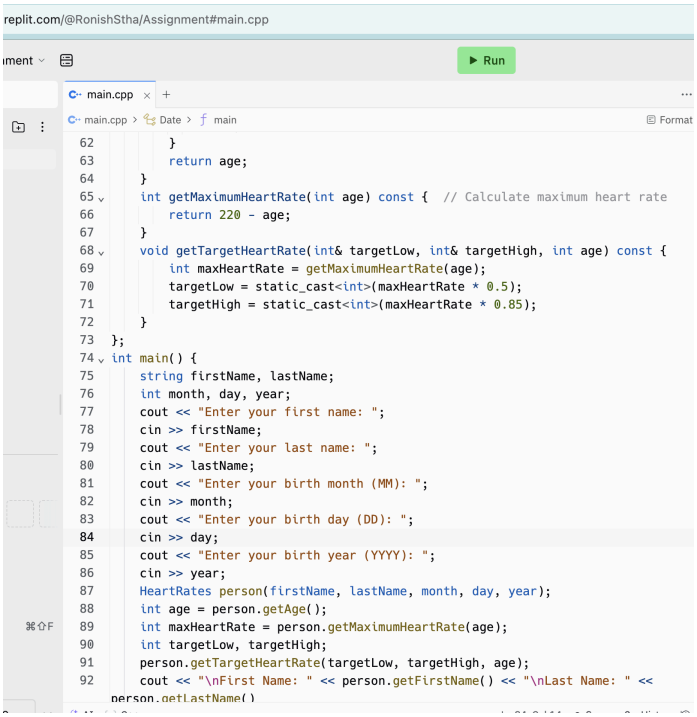
```
// Set and Get functions
void setFirstName(string fName) {
    firstName = fName;
}

string getFirstName() const {
    return firstName;
}

void setLastName(string lName) {
    lastName = lName;
}

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;
}

// Calculate age
int getAge() const {
    int currentYear, currentMonth, currentDay;
    cout << "Enter the current year: ";
    cin >> currentYear;
    cout << "Enter the current month: ";
```

```
        cin >> currentMonth;
        cout << "Enter the current day: ";
        cin >> currentDay;

        int age = currentYear - birthYear;
        if (birthMonth > currentMonth || (birthMonth == currentMonth && birthDay >
currentDay)) {
            age--;
        }
        return age;
    }

    // Calculate maximum heart rate
    int getMaximumHeartRate(int age) const {
        return 220 - age;
    }

    // Calculate target heart rate
    void getTargetHeartRate(int& targetLow, int& targetHigh, int age) const {
        int maxHeartRate = getMaximumHeartRate(age);
        targetLow = static_cast<int>(maxHeartRate * 0.5);
        targetHigh = static_cast<int>(maxHeartRate * 0.85);
    }
};

int main() {
    string firstName, lastName;
    int month, day, year;

    cout << "Enter your first name: ";
    cin >> firstName;
    cout << "Enter your last name: ";
    cin >> lastName;
    cout << "Enter your birth month (MM): ";
    cin >> month;
    cout << "Enter your birth day (DD): ";
    cin >> day;
    cout << "Enter your birth year (YYYY): ";
    cin >> year;

    HeartRates person(firstName, lastName, month, day, year);
    int age = person.getAge();
    int maxHeartRate = person.getMaximumHeartRate(age);
    int targetLow, targetHigh;
    person.getTargetHeartRate(targetLow, targetHigh, age);

    cout << "\nFirst Name: " << person.getFirstName() << "\nLast Name: " <<
person.getLastName()
```

```
        << "\nDate of Birth: " << person.getBirthMonth() << "/" << person.getBirthDay() << "/"
<< person.getBirthYear()
        << "\nAge: " << age << "\nMaximum Heart Rate: " << maxHeartRate
        << "\nTarget Heart Rate Range: " << targetLow << " - " << targetHigh << " bpm" <<
endl;

    return 0;
}
```




replit.com/@RonishStha/Assignment#main.cpp
Run
main.cpp
main.cpp > Date > HeartRates > getAge
Format
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;
}
int getAge() const {
int currentYear, currentMonth, currentDay;
cout << "Enter the current year: ";
cin >> currentYear;
cout << "Enter the current month: ";
cin >> currentMonth;
cout << "Enter the current day: ";
cin >> currentDay;
int age = currentYear - birthYear;
if (birthMonth > currentMonth || (birthMonth == currentMonth
&& birthDay > currentDay)) {
AI
C++
Ln 58, Col 27
Spaces: 2
History


replit.com/@RonishStha/Assignment#main.cpp
Run
main.cpp
main.cpp > Date > main
Format
}
return age;
}
int getMaximumHeartRate(int age) const { // Calculate maximum heart rate
return 220 - age;
}
void getTargetHeartRate(int& targetLow, int& targetHigh, int age) const {
int maxHeartRate = getMaximumHeartRate(age);
targetLow = static_cast<int>(maxHeartRate * 0.5);
targetHigh = static_cast<int>(maxHeartRate * 0.85);
}
};
int main() {
string firstName, lastName;
int month, day, year;
cout << "Enter your first name: ";
cin >> firstName;
cout << "Enter your last name: ";
cin >> lastName;
cout << "Enter your birth month (MM): ";
cin >> month;
cout << "Enter your birth day (DD): ";
cin >> day;
cout << "Enter your birth year (YYYY): ";
cin >> year;
HeartRates person(firstName, lastName, month, day, year);
int age = person.getAge();
int maxHeartRate = person.getMaximumHeartRate(age);
int targetLow, targetHigh;
person.getTargetHeartRate(targetLow, targetHigh, age);
cout << "\nFirst Name: " << person.getFirstName() << "\nLast Name: " <<
person.getLastName()

```
    int age = person.getAge();
    int maxHeartRate = person.getMaximumHeartRate(age);
    int targetLow, targetHigh;
    person.getTargetHeartRate(targetLow, targetHigh, age);
    cout << "\nFirst Name: " << person.getFirstName() << "\nLast Name: " <<
person.getLastName()
         << "\nDate of Birth: " << person.getBirthMonth() << "/" <<
person.getBirthDay() << "/" << person.getBirthYear()
         << "\nAge: " << age << "\nMaximum Heart Rate: " << maxHeartRate
         << "\nTarget Heart Rate Range: " << targetLow << " - " << targetHigh
<< " bpm" << endl;
    return 0;
}
```