Bellow is my Header File that contains the class HeartRates

swyman2501@Knightys-Castle: ~/ECE330/homework3/homework3

```
#ifndef HEARTRATE_HPP
#define HEARTRATE_HPP
#include <iostream>
#include <string>

class HeartRates {
    private:
    std::string _fname;
    std::string _lname;
    int _day, _month, _year;
    int _age;
    public:

    //Constructors
    HeartRates();
    HeartRates(std::string fname, std::string lname, int day, int month, int year, int age);
    HeartRates(const HeartRates &h);

    //setters
    void set_fname(std::string fname);
    void set_lname(std::string lname);
    void set_day(int day);
    void set_month(int month);
    void set_year(int year);
    void set_age(int year);
    //getters
    std::string fname();
    std::string lname();
    int day() const;
    int month() const;
    int year() const;

    int age() const;
    int getMaxHeartRate(int age);
    int getTargetHeartRateupper(int age);
    int getTargetHeartRateLower(int age);
};
#endif // HEARTRATE_HPP
```

Below is my cpp file for HeartRates pretty much the entire program

swyman2501@Knightys-Castle: ~/ECE330/homework3/homework3

```cpp
#include "heartrate.hpp"
//OKAY THIS WORKS! Comment timeeeee!
//below are my constructors, including an unperamertized, peramertized, & a copy constructor
HeartRates::HeartRates() : _fname{"Steven"}, _lname{"Wyman"}, _day{29}, _month{12}, _year{1989}, _age{32} {}
HeartRates::HeartRates(std::string fname, std::string lname, int day, int month, int year, int age) :
     _fname{"Steven"}, _lname{"Wyman"}, _day{29}, _month{12}, _year{1989}, _age{32} {
         set_fname(fname);
         set_lname(lname);
         set_day(day);
         set_month(month);
         set_year(year);
         set_age(year);
     }
HeartRates::HeartRates(const HeartRates &h) : _fname{h._fname}, _lname{h._lname}, _day{h._day}, _month{h._month},
    _year{h._year}, _age{h._age} {}
//bellow is going to be my function calls for my setters and getters
void HeartRates::set_fname(std::string fname) { //this sets first name
    std::cout<<"Please enter your first name: ";
    std::cin >> fname;
    while(fname.size() > 30) {
        std::cout<< "Can not have a first name of over 30 characters please re-enter your name\n";
        std::cin >> fname;
    }
    _fname = fname;
}

void HeartRates::set_lname(std::string lname) { //this sets last name
    std::cout<<"Please enter your last name: ";
    std::cin >> lname;
    while(lname.size() > 30) {
        std::cout<< "Can not have a last name of over 30 characters please re-enter your name\n";
        std::cin >> lname;
    }
    _lname = lname;
}
void HeartRates::set_day(int day) { //sets day
    std::cout<<"Please enter the day you were born: ";
    std::cin >> day;
    //below is how I'm making sure that the input is a int and not a char or anything that can break the prog.
    while(std::cin.fail()) {
        std::cout << "Error please re-enter!: ";
        std::cin.clear();
        std::cin.ignore(256, '\n');
        std::cin >> day;
    }
    while(day > 31 || day <= 0) { //not accounting for february or leap year
        std::cout<<"Error please re-enter the day you were born: ";
        std::cin >> day;
    }
    _day = day;
}

void HeartRates::set_month(int month) { //sets month
    std::cout<<"Please enter the month # you were born: ";
    std::cin >> month;
    while(std::cin.fail()) {
        std::cout << "Error please re-enter!: ";
        std::cin.clear();
        std::cin.ignore(256, '\n');
        std::cin >> month;
    }
    while(month > 12 || month <= 0) {
        std::cout<<"Error please re-enter the month # you were born: ";
        std::cin >> month;
    }
    _month = month;
}

void HeartRates::set_year(int year) { //sets year
    std::cout<<"Please enter the year you were born: ";
    std::cin >> year;
    while(std::cin.fail()) {
        std::cout << "Error please re-enter!: ";
        std::cin.clear();
        std::cin.ignore(256, '\n');
        std::cin >> year;
    }
    while(year < 1900 || year > 2018) {
        std::cout<<"Error please re-enter the year you were born: ";
```

1,1 Top

```
78      while(year < 1900 || year > 2018) {
79          std::cout<<"Error please re-enter the year you were born: ";
80          std::cin >> year;
81      }
82      _year = year;
83  }
84  std::string HeartRates::fname(){ //returns the class attribute of _fname
85      return _fname;
86  }
87
88  std::string HeartRates::lname(){ //returns the class attribute of _lname
89      return _lname;
90  }
91  int HeartRates::day() const { //returns the class attribute of _day
92      return _day;
93  }
94
95  int HeartRates::month() const { //returns the class attribute of _month
96      return _month;
97  }
98
99  int HeartRates::year() const { //returns the class attribute of _year
100     return _year;
101 }
102 void HeartRates::set_age(int year) { //this sets the age attribute
103     _age = 2022 - year;
104 }
105 int HeartRates::age() const{ //this returns the class attribute of_age
106     return _age;
107 }
108
109 int HeartRates::getMaxHeartRate(int age) { //this will give the max heart rate
110   int MHR = 220 - age;
111   return MHR; //MHR = Max Heart Rate
112 }
113
114 int HeartRates::getTargetHeartRatelower(int age) { //this will give the Lower target heart rate
115     int mhr = getMaxHeartRate(age);
116     int LTHR = mhr * .5;
117     return LTHR; //LTHR = Lower Target Heart Rate
118 }
119
120 int HeartRates::getTargetHeartRateupper(int age) { //this will give the upper target heart rate
121     int mhr = getMaxHeartRate(age);
122     int UTHR = mhr * .85;
123     return UTHR; //UTHR = Upper Target Heart Rate
124 }
125     //end of HeartRate.cpp
                                                                    125,1-4       Bot
```

Bellow is my main.cpp file (I'm using slate color scheme with monofur text.. yes I installed custom text on my IDE and yes it was a pain to do)

swyman2501@Knightys-Castle: ~/ECE330/homework3/homework3

```cpp
#include "heartrate.hpp"


int main() {
    char yorn; //yes or no
    int yn = 0;
//  std::cout<< "test!\n"; print statments... the BEST debugging tool
    //I needed the above because I tottaly space out, and for got I need yn to have == and not just =
    while(yn == 0) { // this while loop is my entire main pretty much.. yn = YesNo which gets adjusted at the end
        HeartRates h1;
        std::cout<<"Hello!\n" << "Please enter the requested information for this program to calculate your"
            << " target and max heart rate.\n";
    h1.set_fname(h1.fname());
    h1.set_lname(h1.lname());
    h1.set_month(h1.month());
    h1.set_day(h1.day());
    h1.set_year(h1.year());
    h1.set_age(h1.year());
    std::cout << "\nHere is the info you provided\n" << "Name: " << h1.fname() << ' ' << h1.lname()
        << "\nDate of Birth in Day/Month/Year format: " << h1.day() << "/" << h1.month() << "/" << h1.year()
        << " making you " << h1.age() << " years old this year." << "\nYour Max heart rate is: "
        << h1.getMaxHeartRate(h1.age()) << " bpm.\nYour Target heart rate is: "
        << h1.getTargetHeartRatelower(h1.age()) << "-" << h1.getTargetHeartRateupper(h1.age()) << " bpm.\n\n";
    //below is my way of asking the user if they want to add new info or restart the program
    ask: //this ask: is part of a goto in the switch statement below
    std::cout << "Would you like to restart with a new entry? [Y/N]: ";
    std::cin >> yorn; //yes or no
        switch(yorn) {
            case 'Y': //I account for lower and upper in this statement
            case 'y':
                std::cout<< "\nrestarting!\n\n"; //restarts the loop by not adjusting yn thus sending it back up
                break;
            case 'N':
            case 'n':
                std::cout<<"Thank you for using this program, we hope that it helped you!" << std::endl;
                yn++; //adds 1 to yn thus breaking the loop
                break;
            default:
                std::cout<< "Error in proccessing request" << std::endl;
                goto ask; //here I didn't want to restart the whole while loop but also didn't want to exit
                break; // I've used goto statements in 131 much to my instructor's surpise..but they work okay!? XD
        }
    }
    return 0;
}
//end of main
```

Bellow is my Makefile using the c++ 2017 library

swyman2501@Knightys-Castle: ~/ECE330/homework3/homework3

```
GPP= g++ -std=c++17

all: HeartRates

HeartRates: HeartRate.o main.o
    $(GPP) $^ -o $@

HeartRate.o: HeartRate.cpp heartrate.hpp
    $(GPP) -c $<

main.o: main.cpp heartrate.hpp
    $(GPP) -c $<

clean:
    -rm *.o HeartRates
```