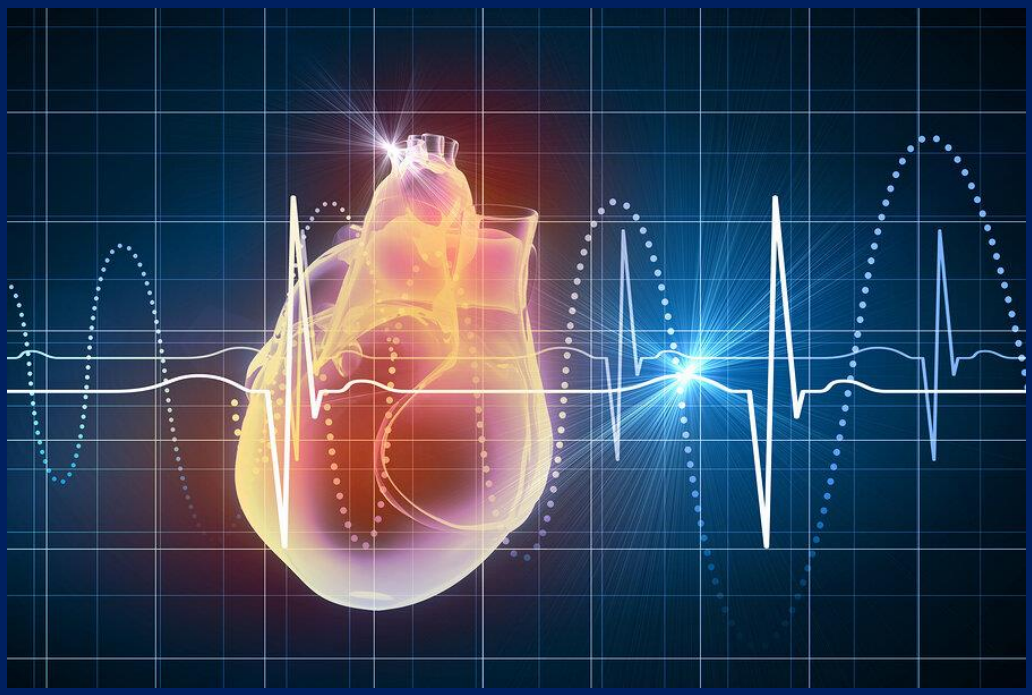
Homework #3



ECE 330 – Heart-Rate Calculator Report

Spring 2021

Author: Clarizza Morales
Date: February 26th , 2021
Instructor: Amir Raeisi
Course: Software Design

Table of Contents

..... 1

Table of Contents 2

List of Figures 3

Heart_RateCalculator.cpp 4

Heart_RateCalculator.hpp 7

Main.cpp 8

Output 9

Conclusion 10

List of Figures

Figure 1. Heart-Rate Calcultor .cpp file..... 4

Figure 2. Heart-Rate Calcultor .cpp file..... 5

Figure 3. Heart-Rate Calcultor .cpp file..... 6

Figure 4. Heart_RateCalculator.hpp Header File 7

Figure 5. Heart-RateCalculator Main.cpp Routine 8

Figure 6. Heart_RateCalculator Output 9

Heart_RateCalculator.cpp

```
#include "heart_calculator.hpp"
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <string>

std::string _Name;
std::string _LastName;
int _birthMonth;
int _birthYear;
int _birthDay;
        //int _todaysDay;
        //int _todaysMonth;
        //int _todaysYear;

HeartRates::HeartRates(std::string Name, std::string LastName, int birthMonth, int birthDay, int birthYear){
                //std::cout << "Parametrized constructor\n";
                _Name(Name);
                _LastName(LastName);
                _birthDay(birthDay);
                _birthMonth( birthMonth);
                _birthYear(birthYear);
                }
        //setters and getter methods

        //setter for Name
void HeartRates::_Name(std::string Name)
        {
                personName  = Name;
                //void method since we do not have to return anything
        }
        //Getter for Name
std::string HeartRates::getName()
        {
                return personName;
        }

       //setter for last name
void HeartRates::_LastName(std::string LastName)
        {
                personlastName = LastName;
        }
        //getter for last name
```

Figure 1. Heart-Rate Calcultor .cpp file

```
        }
        //getter for last name
std::string HeartRates::getLastName()
        {
                return personlastName;
        }

        //setter for birth month
void HeartRates::_birthMonth(int birthMonth)
        {
                BirthMonth = birthMonth;
        }
        //getter for birth month
int  HeartRates::getbirthMonth()
        {
                return BirthMonth;
                //int method since we have to return an int value
        }

        //setter for birth day
void HeartRates::_birthDay(int birthDay)
        {
                BirthDay = birthDay;
        }
        //getter for birth day
int HeartRates::getbirthDay()
        {
                return BirthDay;
        }

        //setter for birth year
void HeartRates::_birthYear(int birthYear)
        {
                BirthYear = birthYear;
        }
        //getter for birth year
int HeartRates::getbirthYear()
        {
                return BirthYear;
        }
```

Figure 2. Heart-Rate Calcultor .cpp file

```
int HeartRates::getAge(int currentMonth, int currentDay, int currentYear){
		int x;
		x = 0;
		int age;

		if (getbirthMonth() > currentMonth)
		{
			x = -1;
		}
		if ((getbirthMonth() == currentMonth) && (getbirthDay() > currentDay))
		{
			x = -2;
		}

		age = (currentYear - getbirthYear() + x);
		return age;
	}


	//function getMaximumHeartRate() to find the persons heartrate

int HeartRates::getMaximumHeartRate(int currentMonth, int currentDay, int currentYear)
	{
		//220 - age in years
		int age;
		age = (220 - getAge(currentMonth, currentDay, currentYear));
		return age;
	}

	//function getTargetHeartRate to calculate and return the person's target heart rate
int HeartRates::getTargetHeartRate(int &minimumTargetRate, int &maximumTargetRate, int currentMonth, int currentDay, int currentYear)
	{
		//target heart rate is 50-85% of max heart rate
		//e.g max hr = 180
		//min target is (180*50)/100

		int i;
		i = getMaximumHeartRate(currentMonth, currentDay, currentYear);
		minimumTargetRate = ((i*50) / 100);
		maximumTargetRate = ((i*85) / 100);

		return 0;
```

Figure 3. Heart-Rate Calcultor .cpp file

Heart_RateCalculator.hpp

```
/* File: heart_calculator.h */

#include <stdio.h>
#include <stdlib.h>
#include <string>

class HeartRates{
        private:
                std::string personName;
                std::string personlastName;
                int BirthMonth;
                int BirthDay;
                int BirthYear;

        public:

                HeartRates(std::string Name, std::string  LastName, int birthMonth, int  birthDay, int  birthYear);
                //setters and getter functions
                void _Name(std::string Name);
                std::string getName();
                void _LastName(std::string LastName);
                std::string getLastName();
                void _birthMonth(int birthMonth);
                int getbirthMonth();
                void _birthDay(int birthDay);
                int getbirthDay();
                void _birthYear(int birthYear);
                int getbirthYear();
                int getAge(int currentMonth, int currrentDay, int currentYear);
                int getMaximumHeartRate(int currentMonth, int currentDay, int currentYear);
                int getTargetHeartRate(int &, int &, int currentMonth, int currentDay, int currentYear);
};
```

Figure 4. Heart_RateCalculator.hpp Header File

Main.cpp

```
//#include "heart_calculator.cpp"
#include "heart_calculator.hpp"
#include <string>
#include <stdlib.h>
#include <iostream>

int main()
{
        int day, month, year;
        std::string name = " ";
        std::string lastName = " ";
        int minimumTarget, maximumTarget;
        int age;
        int currentMonth;
        int currentDay;
        int currentYear;

        std::cout << "Please enter your name and date of birth as follows(name lastname mm dd yyyy): " << "\n";
        std::cin >> name >>  lastName >>  month >> day >> year;

        HeartRates user(name, lastName, month, day, year);

        std::cout << "Your first name is: " << user.getName() << "\n";
        std::cout << "Your last name is: " << user.getLastName() << "\n";
        std::cout << "Your birthdate is: " << user.getbirthMonth() << "/" << user.getbirthDay() << "/" << user.getbirthYear() << "\n";

        std::cout << "Enter present date as (mm dd yyyy) : " << "\n";
        std::cin >> currentMonth >> currentDay >> currentYear;

        std::cout << "Your age in years is: " << user.getAge(currentMonth, currentDay, currentYear) << " years"<< "\n";


        std::cout << "Your Maximum Heart Rate according to the American Heart Association is: " << user.getMaximumHeartRate(currentMonth, currentDay, currentYear) << "\n";

        std::cout << "You Target Heart Rate is: " << user.getTargetHeartRate(minimumTarget, maximumTarget, currentMonth, currentDay, currentYear);
        std::cout << minimumTarget << " -- " << maximumTarget << "\n";

        return 0;
}
```

Figure 5. Heart-RateCalculator Main.cpp Routine

Output

```
(base) clarizza@MacBook-Pro ECE-330-SoftwareDesing-HW3 % ./run_heartCalc
Please enter your name and date of birth as follows(name lastname mm dd yyyy):
Clarizza Morales 11 12 1999
Your first name is: Clarizza
Your last name is: Morales
Your birthdate is: 11/12/1999
Enter present date as (mm dd yyyy) :
03 16 2021
Your age in years is: 21 years
Your Maximum Heart Rate according to the American Heart Association is: 199
You Target Heart Rate is: 099 -- 169
```

Figure 6. Heart_RateCalculator Output

Conclusion

I struggled a good time to create the class and all its methods. Especially with calculating the age and then finding the maximum and target heart rates. This was because the assignment asks to have the function getAge() ask the user for the present date, but whenever I would do everything inside of this function I was not able to get the return value and use it in the other methods. Whenever I needed the age and called this function then used it to get the heart rates, of course the function will ask the user again and again the present date. So, I decided to give it the parameters and prompt the user for the present date in main, then use those values of currentDay, currentMonth, and currentYear as the parameters to the function. In this way whenever I called the function getAge() inside of getMaximumHeartRate() and getTargetHeartRate(), it used those values to calculate the age but avoided asking the user again for the present date. I forgot at the beginning how to set some of the setters and getters, especially for the string type values, but I referenced the c++ documentation and was able to fix it.

I have uploaded the Makefile, main.cpp, heart_calculator.cpp where the methods are declared, and heart_calculator.hpp with the functions' prototypes and their parameters as well as where I defined the class, and finally the HeartCalculator_Output.txt file with the output results.