Øving 6

Main.cpp

```
#include <iostream>
#include <filesystem>
#include <fstream>
#include <map>
#include "opg3.h"
#include "temp.h"
#include <vector>


//1-----------------------------------
void writeFile() {
    std::string input;
    std::cout << "Skriv inn i fil: ";
    std::getline(std::cin, input);

    std::ofstream fileName{"myFile.txt"};

    fileName << input << std::endl;

    fileName.close();
    std::cout << input << " skrevet inn i fil";
}


void readFile() {
    std::ifstream fil("myFile.txt");
    std::ofstream fileName{"newFile.txt"};

    std::string line;

    while(std::getline(fil, line)) {
        fileName << line << std::endl;
        std::cout << line << " skrevet inn" << std::endl;
    }
}
//-----------------------------------

//2-----------------------------------

void countLetters() {
    std::ifstream fil("grunnlov.txt");
    std::map<char, int> letterCount;
    char c;
    std::vector<char> tegn;
    int tegnCount = 0;

    while(fil.get(c)) {
        tegn.push_back(c);
        tegnCount++;
    }

    for(char bokstav = 'a'; bokstav <= 'z'; bokstav++) {
        int k = 0;
        for(int i = 0 ; i < tegnCount; i++) {
```

```
            if(bokstav == tegn.at(i)) {
                k++;
            }
        }
        letterCount[bokstav] = k;

        std::cout << "Antall ganger " << bokstav << " er i teksten: " << letterCount[bokstav] << std::endl;
    }

    std::cout << "Antall tegn i teksten: " << tegnCount << std::endl;

}


//-------------------------------------

int main() {
    //1-----------------------------------

    //writeFile();
    //readFile();

    //-------------------------------------
    //-------------------------------------
    //-------------------------------------


    //2-----------------------------------

    //countLetters();

    //2b: problemet er funksjonen getCapital der capitalsMap er ikke gjenkjent fra før av,
    //    derfor må du sette en map i getCapital, og sette inn capitalsMap når du printer ut.
    //    (rart å bruke iostream og std_lib_facilities.h samtidig)

    //-------------------------------------
    //-------------------------------------
    //-------------------------------------


    //3-----------------------------------
    CourseCatalog cc;
    //cc.addCourse();

    //cc.removeCourse();
    //std::cout << cc << std::endl;
    //std::cout << cc.getCourse() << std::endl;
    //cc.wFile();
    //cc.rFile();


    //-------------------------------------
    //-------------------------------------
    //-------------------------------------


    //4-----------------------------------
    Temps temp;

    //std::cout << temp << std::endl;

    //temp.readTemps("temperatures.txt");
```

```
    //temp.readTemps("temperatures.txt");

    //temp.readTemps("temperatures.txt");
    temp.tempStats(temp.readTemps("temperatures.txt"));

    //std::cout << temp.readTemps("temperatures.txt") << std::endl;
    //-------------------------------------
    //-------------------------------------
    //-------------------------------------



    return 0;
}
```

opg3.h

```
#pragma once
#include <iostream>
#include <map>
#include <fstream>
//#include <ostream>

class CourseCatalog {
private:
    std::map<std::string, std::string> course;
public:
    CourseCatalog(std::map<std::string, std::string> course) : course{course} {};
    friend std::ostream& operator<<(std::ostream& os, const CourseCatalog& c);
    CourseCatalog() {}
    void addCourse();
    void removeCourse();
    std::string getCourse();
    void wFile();
    void rFile();
};
```

opg3.cpp

```
#include "opg3.h"


void CourseCatalog::addCourse() {
    course.insert({"TDT4110","Informasjonsteknologi grunnkurs"});
    course.insert({"TDT4102","Prosedyre- og objektorientert programmering"});
    course.insert({"TMA4100","Matematikk 1"});
    //course.insert({"TDT4102","Prosedyre- og objektorientert programmering"}); oppdaterer ikke
    course["TDT4102"] = "C++"; //oppdaterer
}

void CourseCatalog::removeCourse() {
```

```
    course.erase("TDT4102");
}


std::string CourseCatalog::getCourse() {
    std::string kurs = course["TDT4110"];
    //std::string kurs = course.at("TDT4110"); samme faen
    return kurs;
}

std::ostream& operator<<(std::ostream& os, const CourseCatalog& c) {
    for(auto const& element : c.course) {
        std::cout << "Fag: " << element.first << ", emnekode: " << element.second << std::endl;
    }
    return os;
}


void CourseCatalog::wFile() {
    std::ofstream fileName{"fag.txt"};

    for(auto const& element : course) {
        fileName << "Fag: " << element.first << ", emnekode: " << element.second << std::endl;
    }
}

void CourseCatalog::rFile() {
    std::ifstream fil("fag.txt");
    std::string line;

    while(std::getline(fil, line)) {
        std::cout << line << std::endl;
    }

}
```

temp.h

```
#pragma once
#include <iostream>
#include <map>
#include <fstream>
#include <vector>

class Temps {
private:
    std::map<double, double> temp;
public:
    //Temps(std::map<double, double> tempMap) : temp(tempMap) {};
    Temps() {};
    //friend std::istream& operator>>(std::istream& is, const Temps& t);
    std::vector<double> readTemps(std::string fil);
    void readTemps1();
    void tempStats(std::vector<double> vec);
};
```

temp.cpp

```
#include "temp.h"

std::istream& operator>>(std::istream& is, const Temps& t) {
    std::filesystem::path temperatureFile{"temperatures.txt"};
   std::ifstream temp_file{temperatureFile};
    temp_file >> t;
    return is;
}

std::vector<double> Temps::readTemps(std::string fil) {
    std::ifstream file(fil);
    std::vector<double> vektor;
    std::string line;

    while(std::getline(file, line)) {
        //std::cout << line << std::endl;
        //vektor.push_back(double(line.substr(0,pos)));
        std::string st2 = line.substr(0, 4);
        std::string st3 = line.substr(4);

        double a = std::stod(st2);
        double b = std::stod(st3);

        vektor.push_back(a);
        vektor.push_back(b);


        //std::string st1 = line[0,pos];
        //std::string st2 = line.substr(pos+1);
        //std::cout << st1 << " " << st2 << std::endl;
        //std::cout << st2 << "                " << st3 << std::endl;

    }
    return vektor;
}


void Temps::readTemps1() {
    std::ifstream file("temperatures.txt");
    std::vector<double> vektor;
    std::string line;

    while(std::getline(file, line)) {
        std::cout << line << std::endl;
    }
}

void Temps::tempStats(std::vector<double> vec) {
    double a = *std::max_element(vec.begin(),vec.end());
    double b = *std::min_element(vec.begin(),vec.end());
    int i = 0;
    int j = 0;

    while(vec.at(i) != a) {
        i++;
    }
```

```
    while(vec.at(j) != b) {
        j++;
    }

    if(i % 2 != 0) {
        i--;
    }

    if(j % 2 != 0) {
        j--;
    }

    double dagmax = i / 2;
    double dagmin = j / 2;

    std::cout << "Maksimal temperatur: " << a << " på dag: " << dagmax << std::endl;
    std::cout << "Minimal temperatur: " << b << " på dag: " << dagmin << std::endl;

}
```