**Operating Systems - Lab 02 In-Lab & Post-Lab Tasks**

Ibrahim Johar Farooqi

23K-0074

BAI-4A

29 January 2025

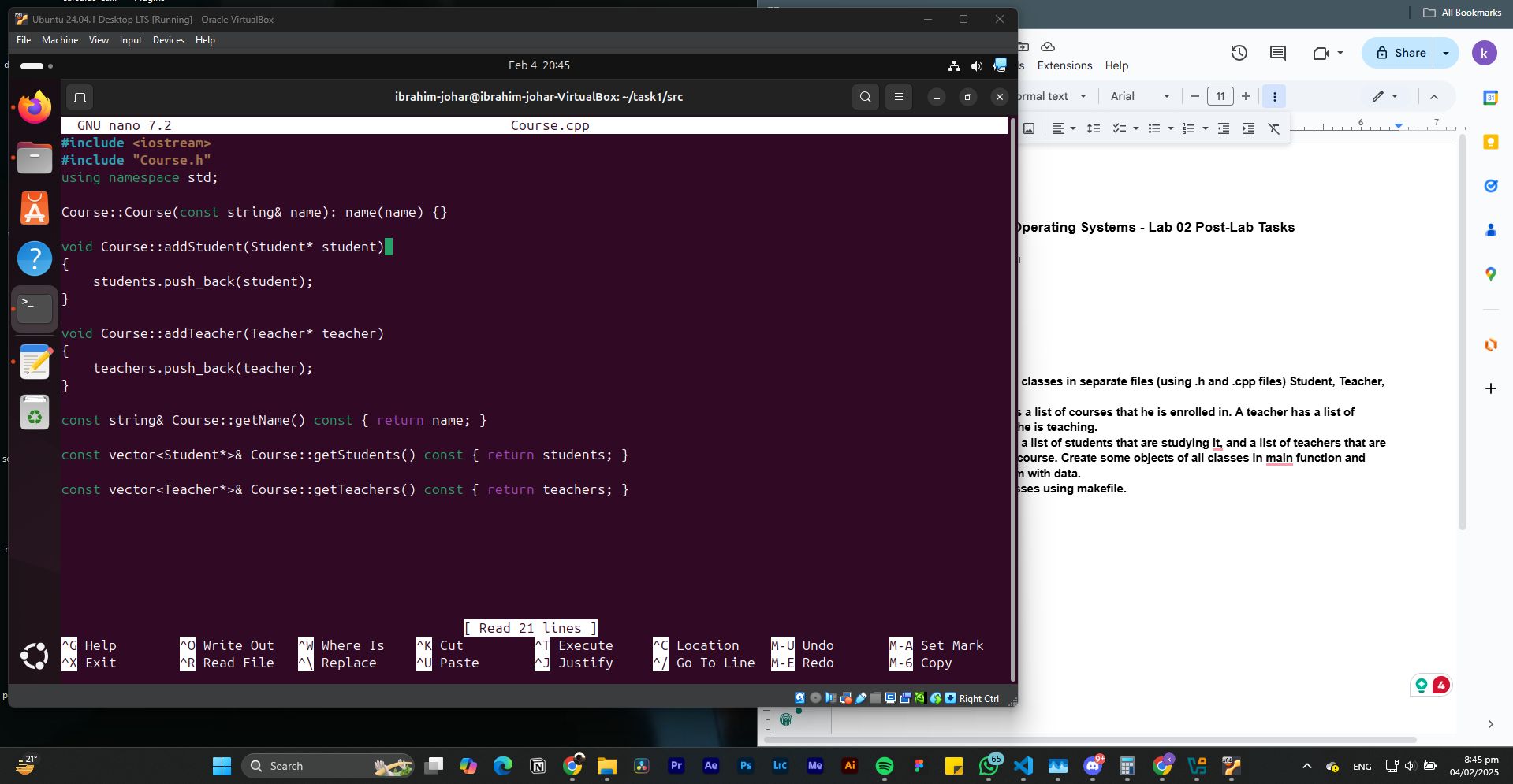
**In-Lab Questions:**

**Task 1:**

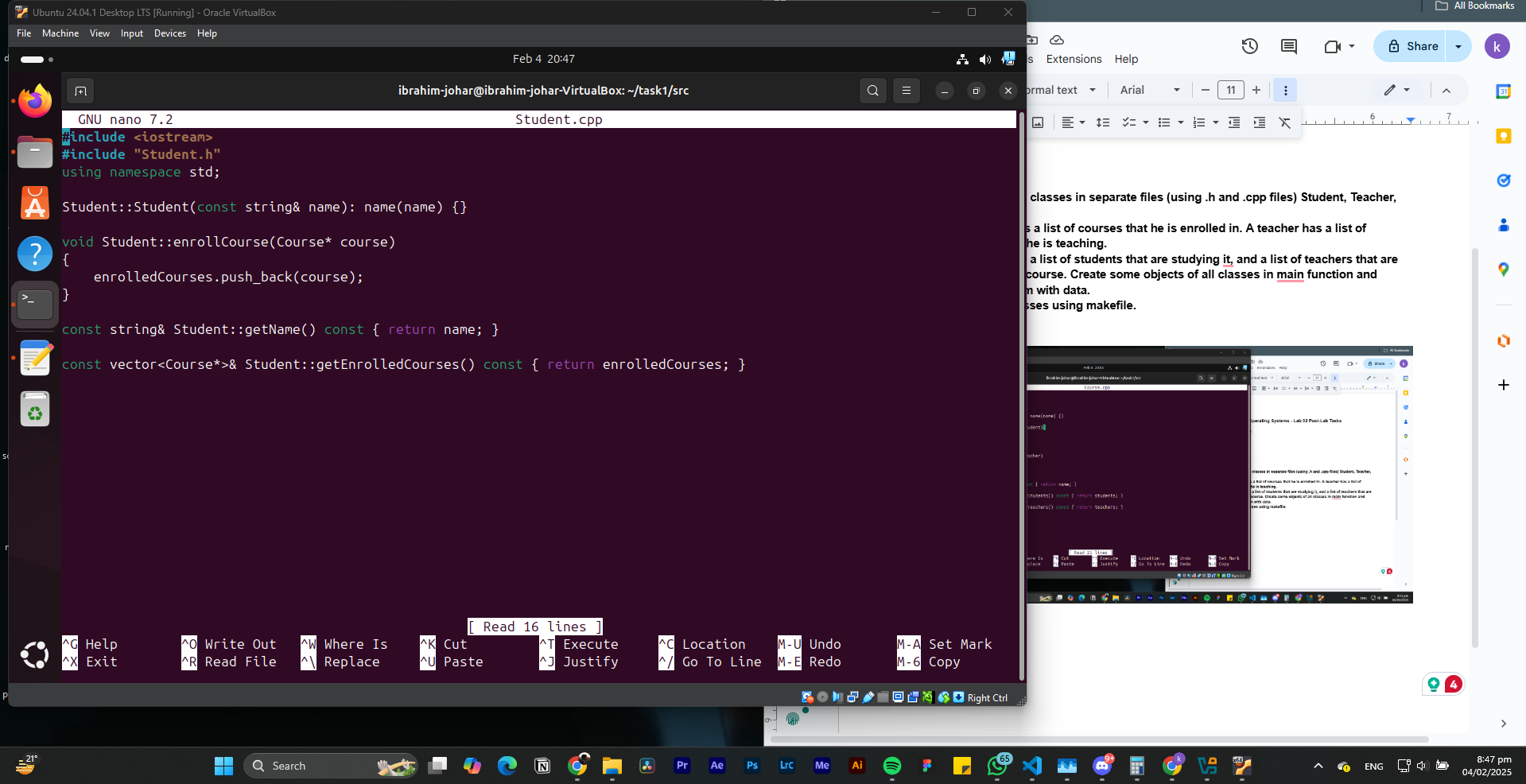
**Create the following classes in separate files (using .h and .cpp files) Student, Teacher, Course.**

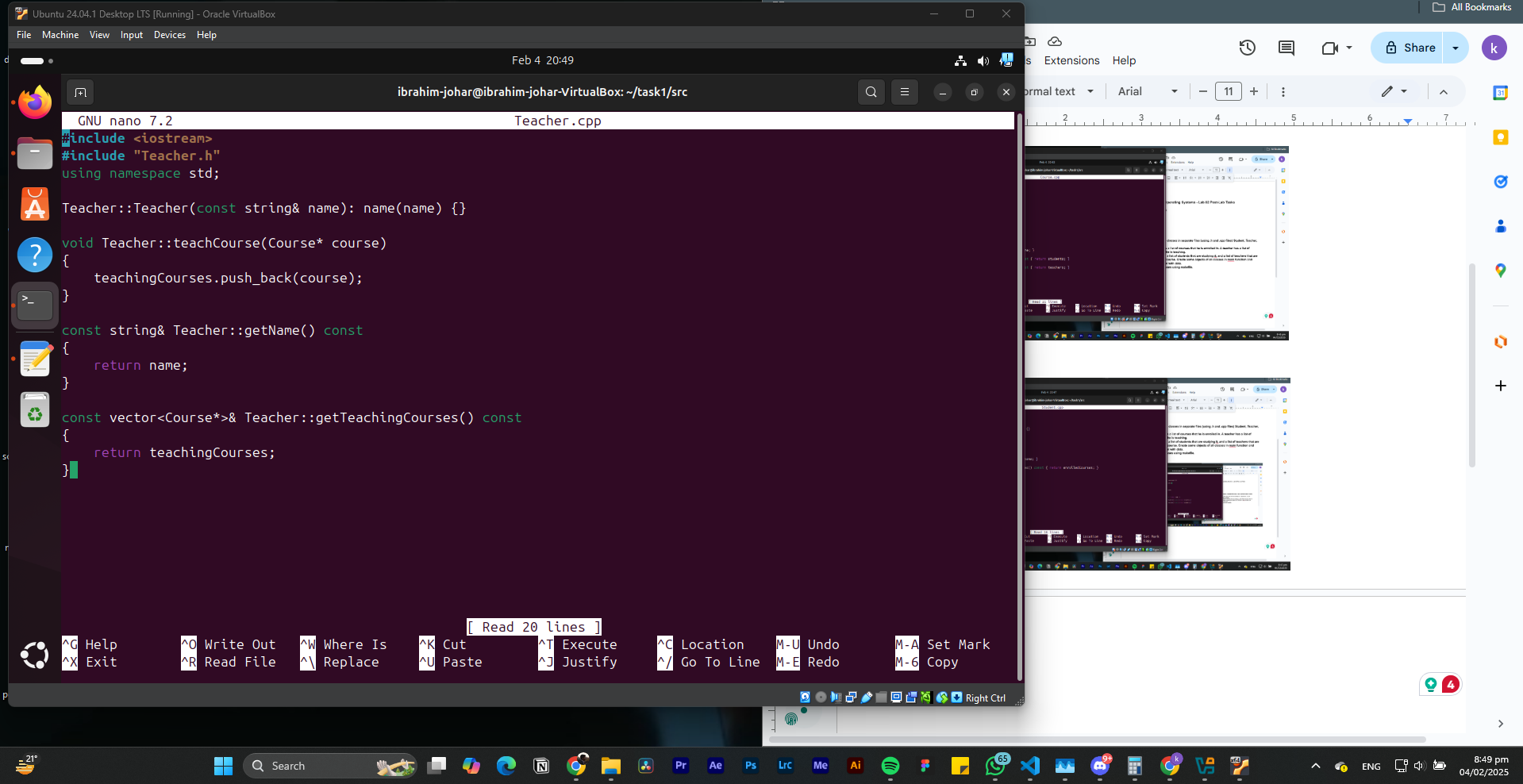
* **A student has a list of courses that he is enrolled in. A teacher has a list of courses that he is teaching.**
* **A course has a list of students that are studying it, and a list of teachers that are teaching the course. Create some objects of all classes in main function and populate them with data.**

**Now compile all classes using makefile.**

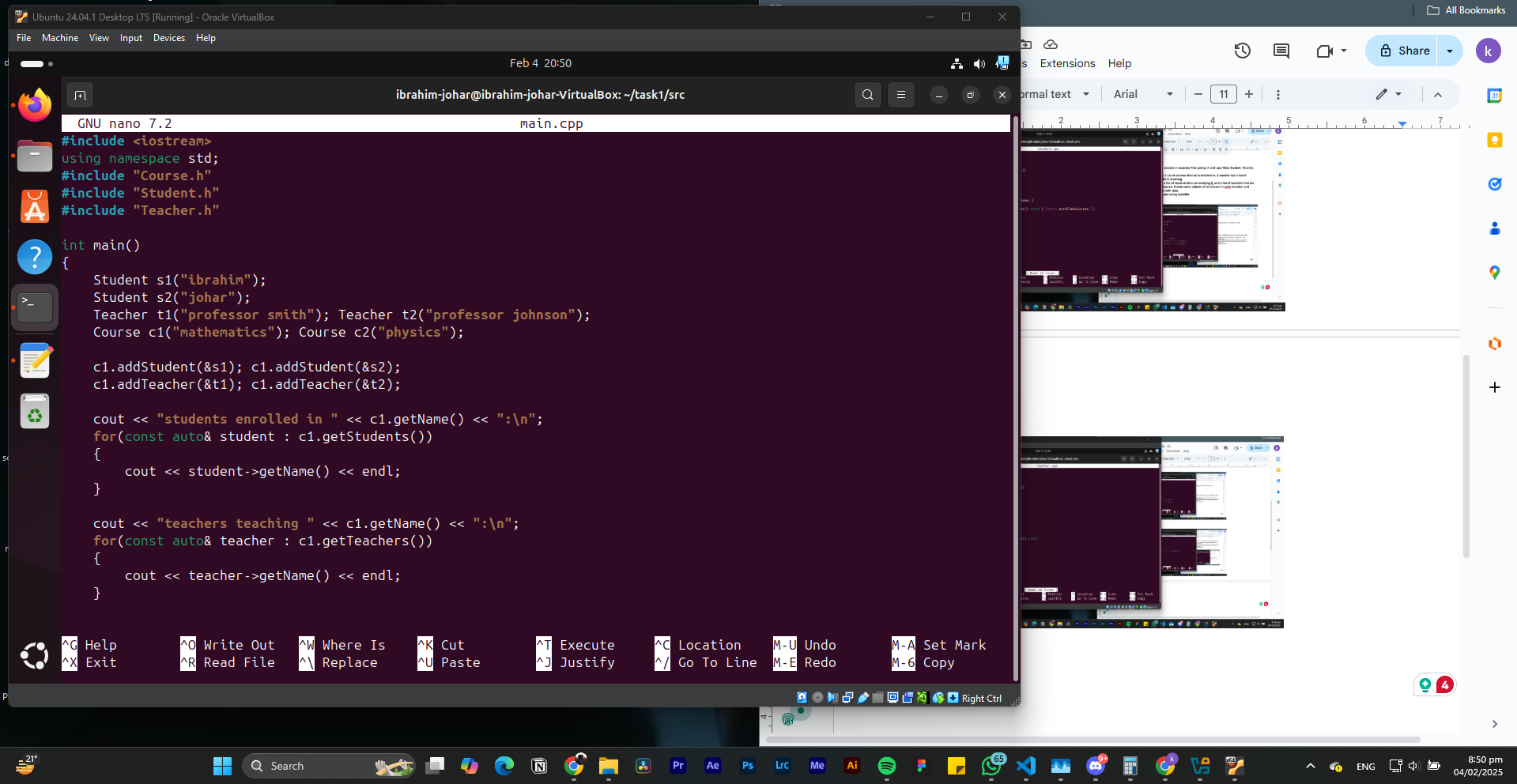
Course.cpp

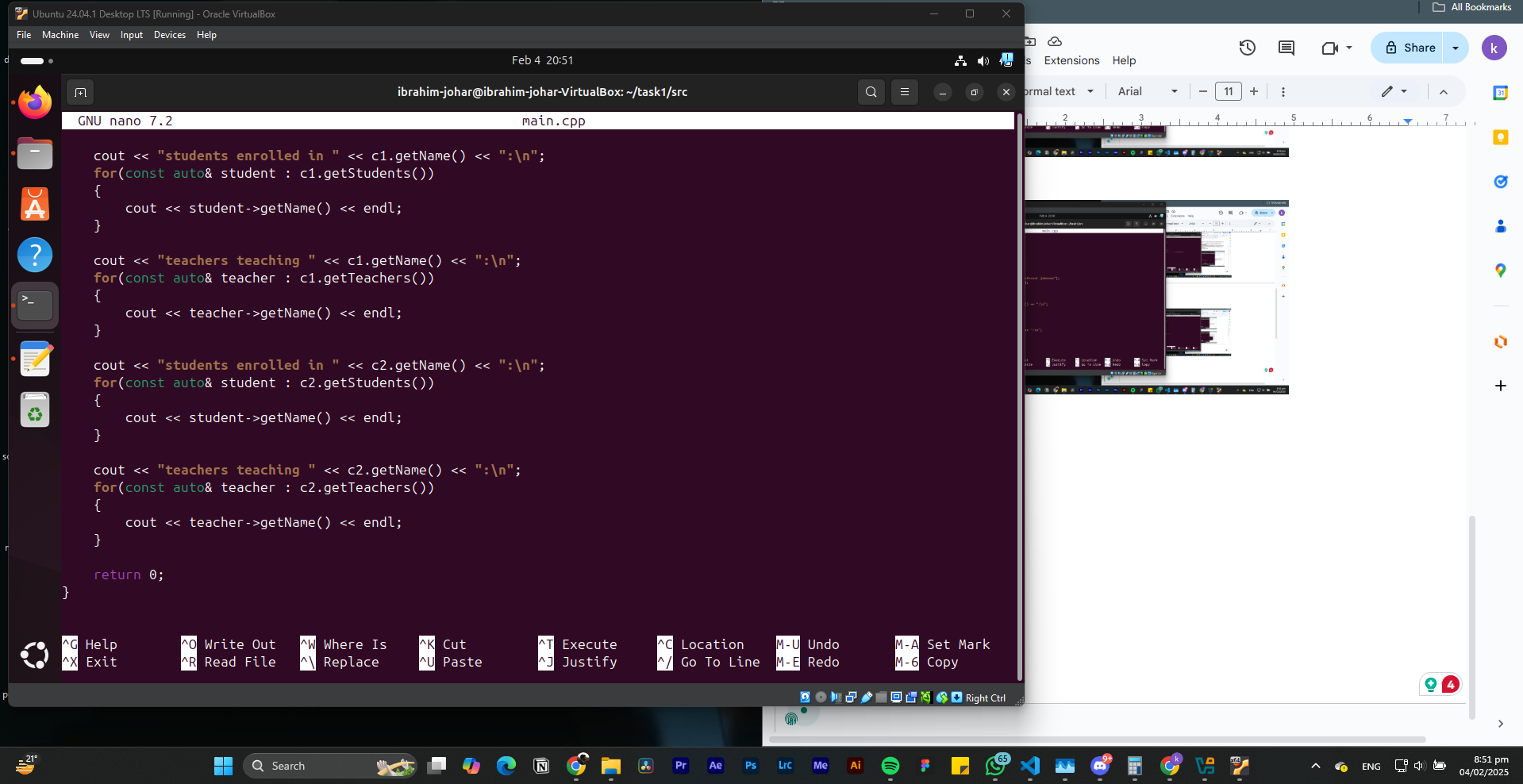
Student.cpp

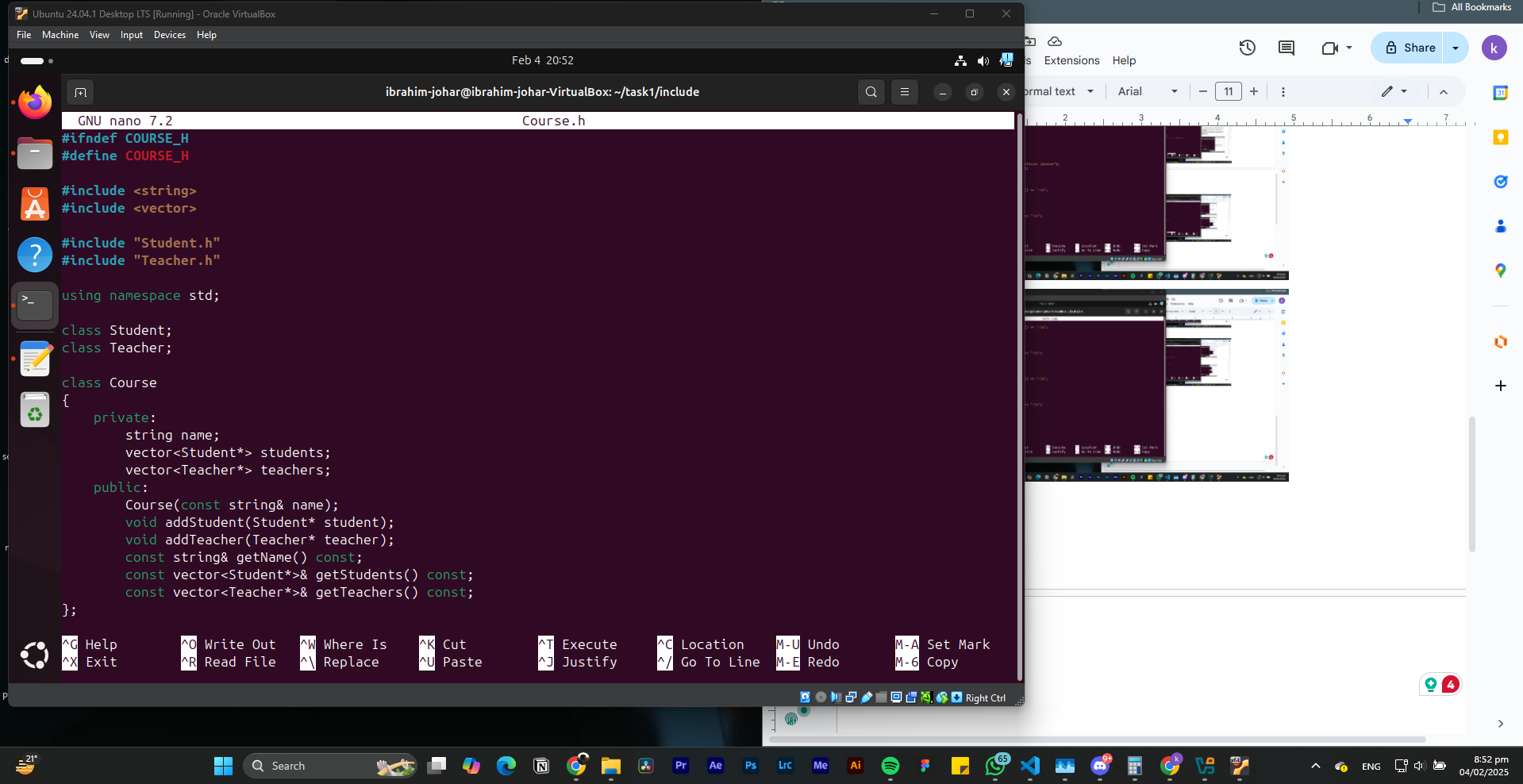


Teacher.cpp

main.cpp

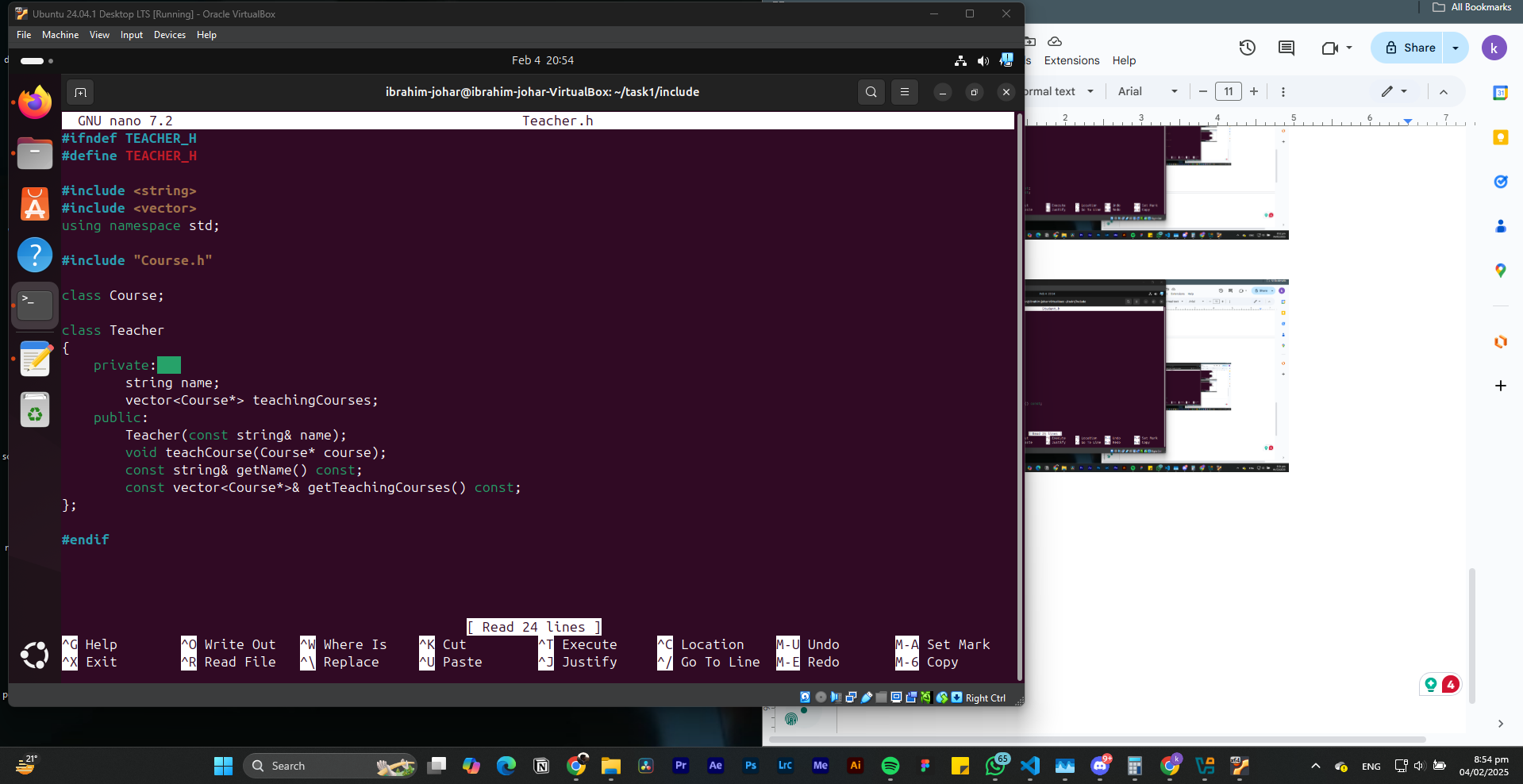


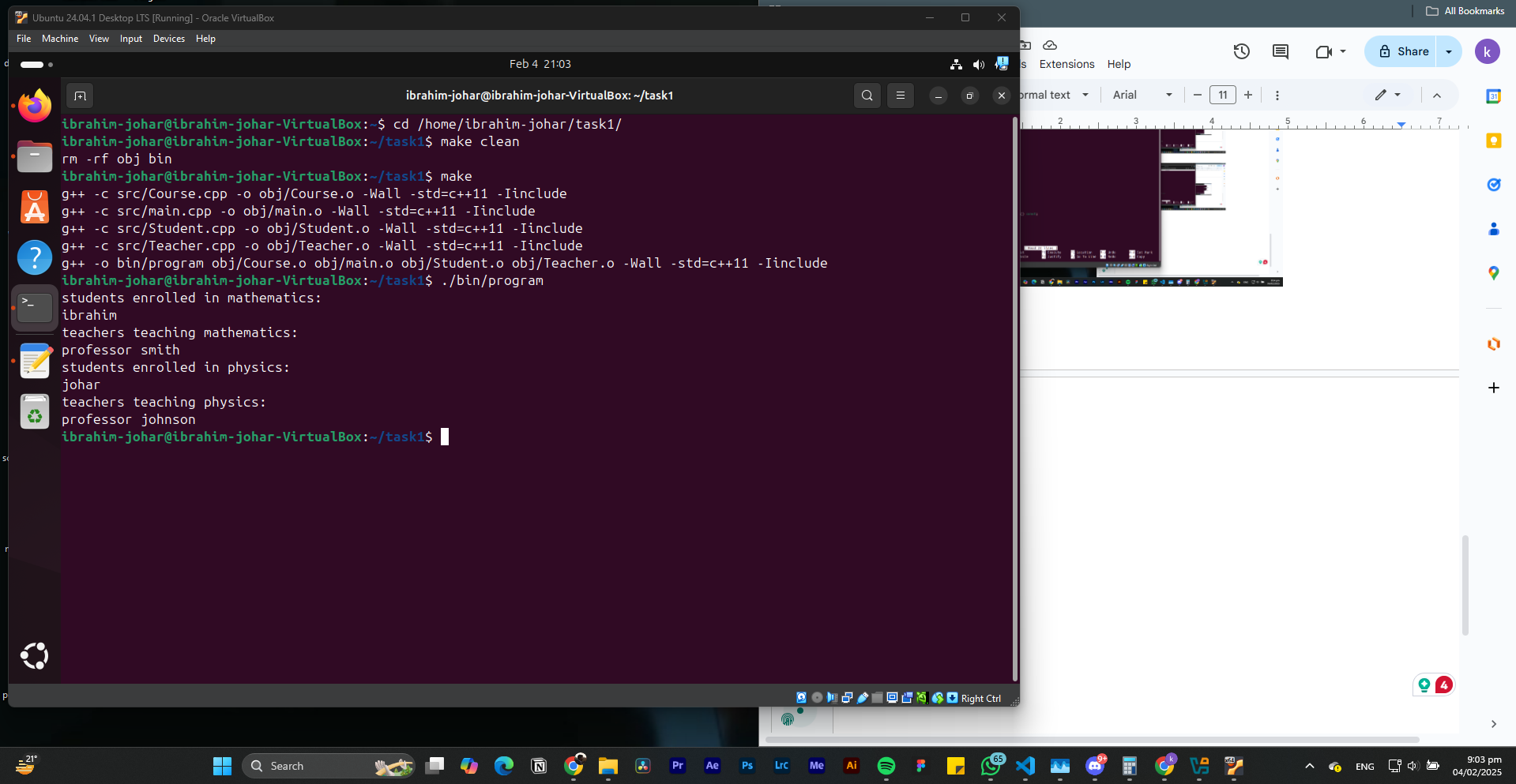


Course.h

Student.h

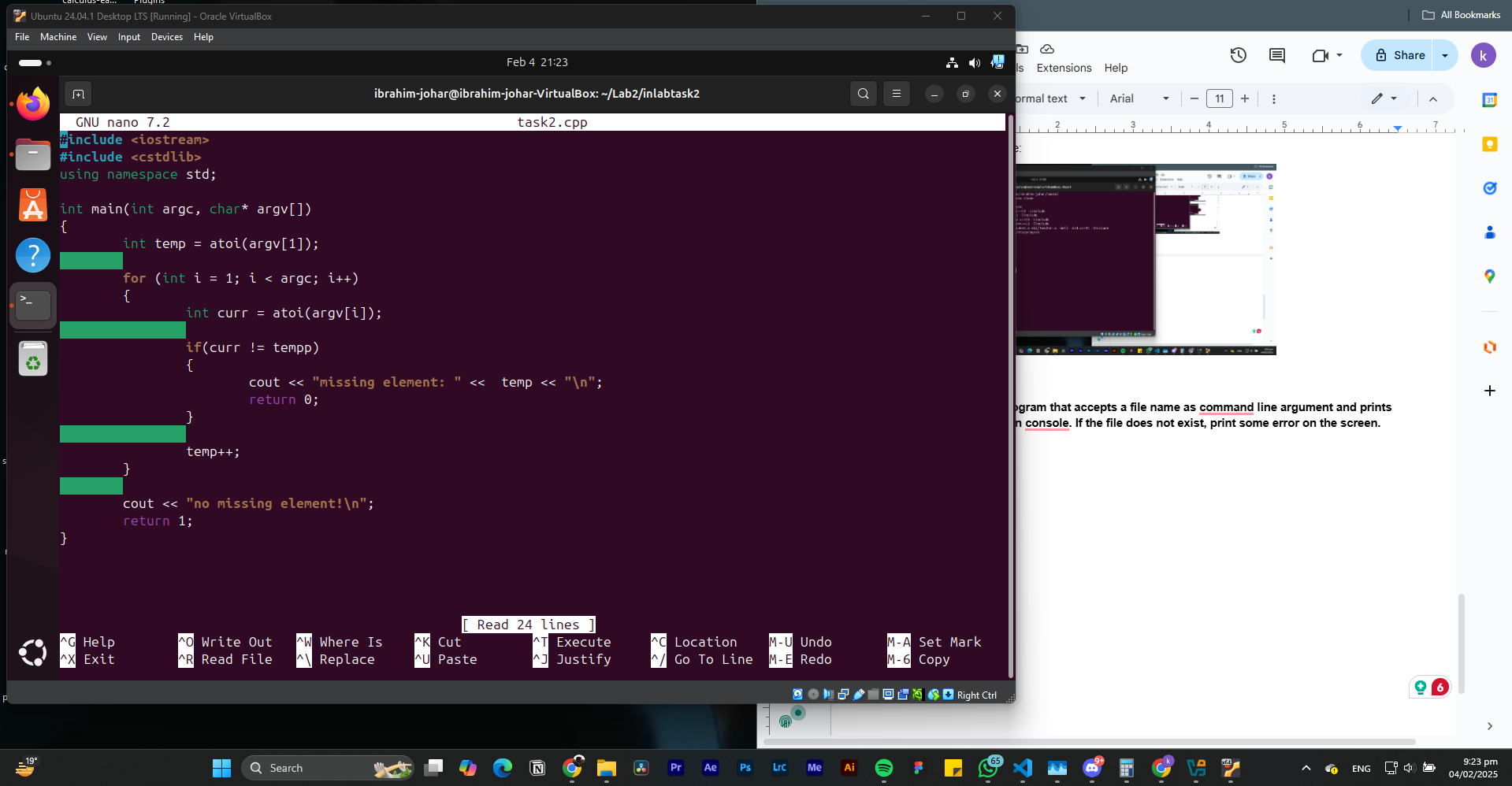


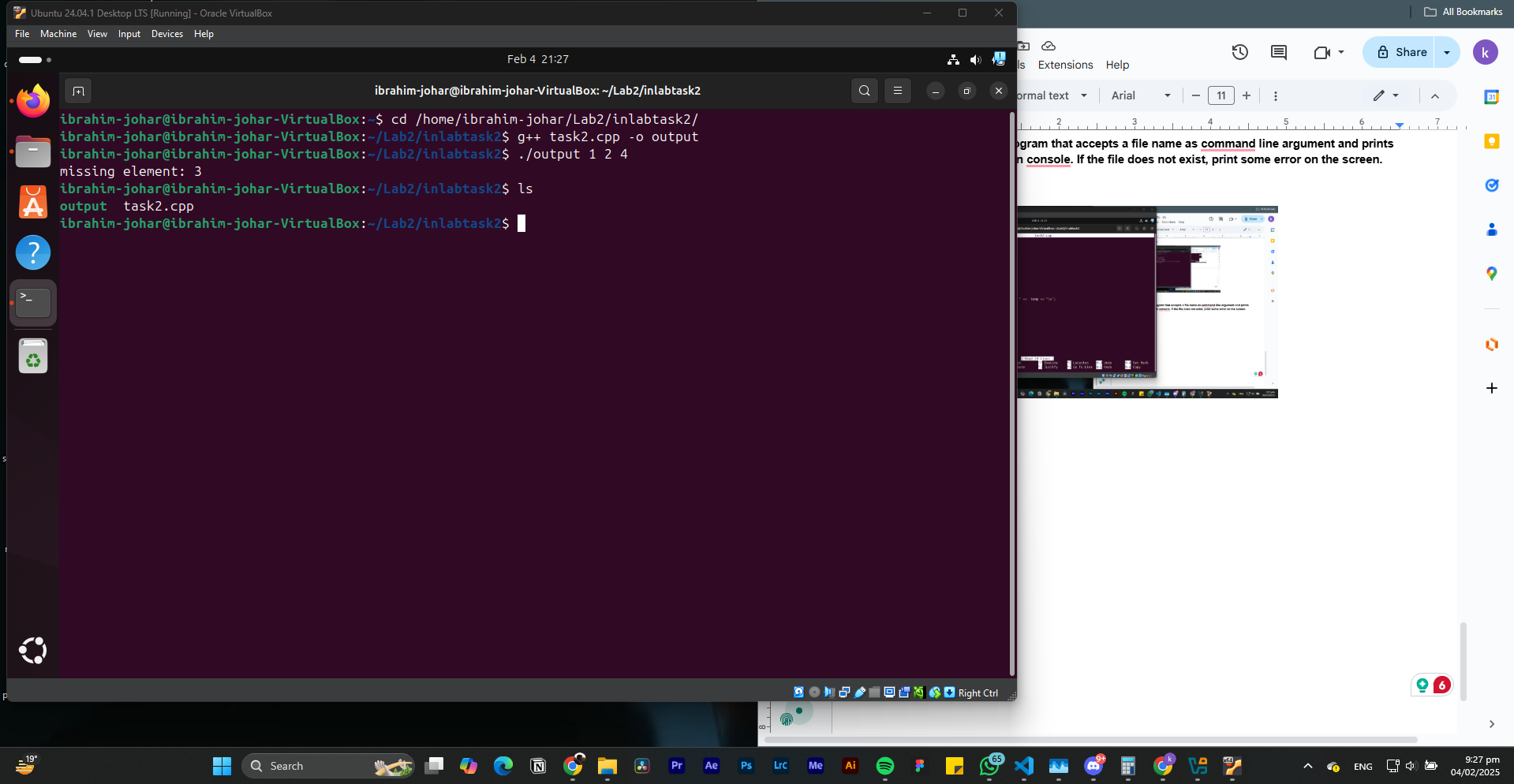
Teacher.h

Output using makefile:

**Task 2:**

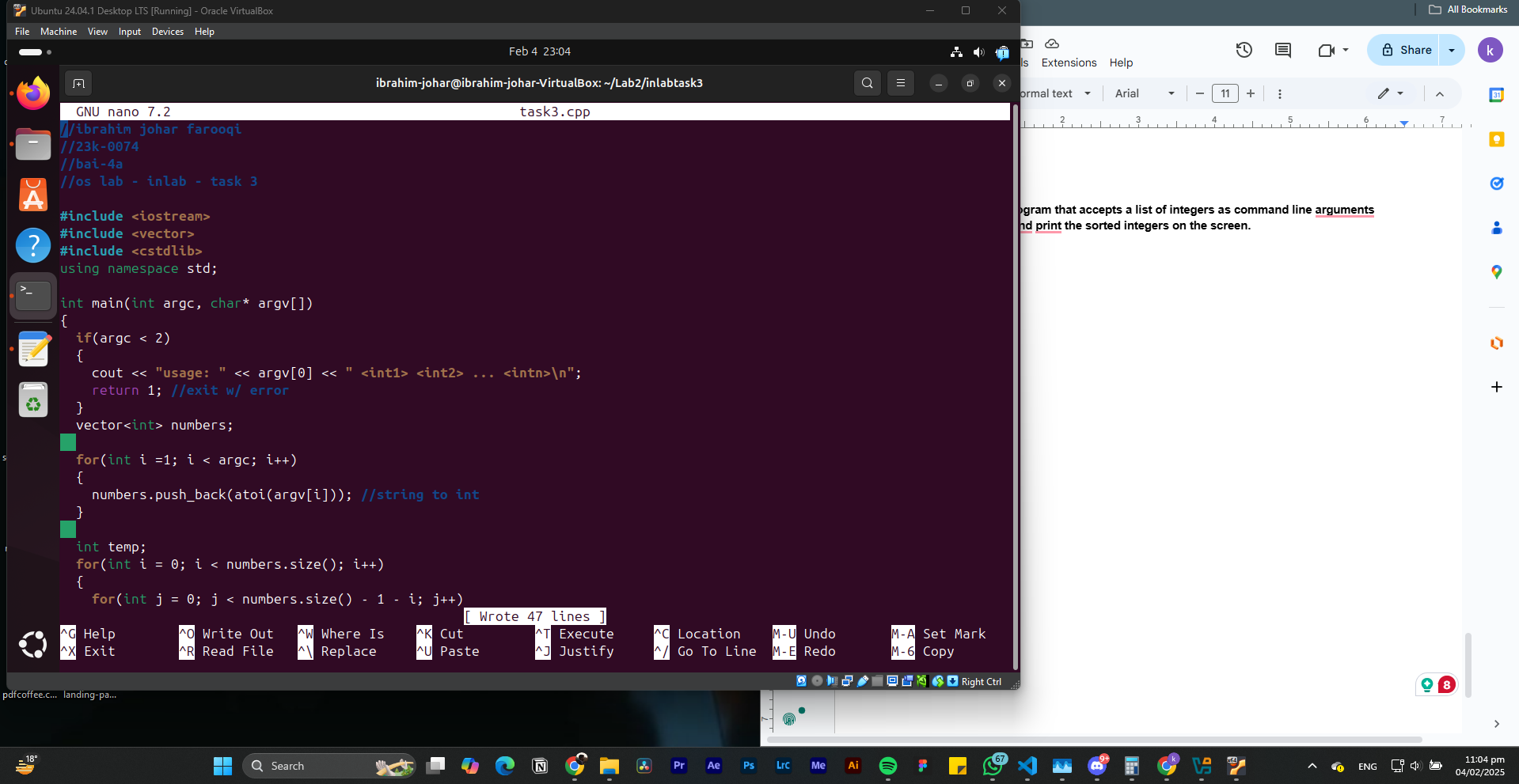
**Write a C or C++ program that accepts a file name as command line argument and prints the file’s contents on console. If the file does not exist, print some error on the screen.**

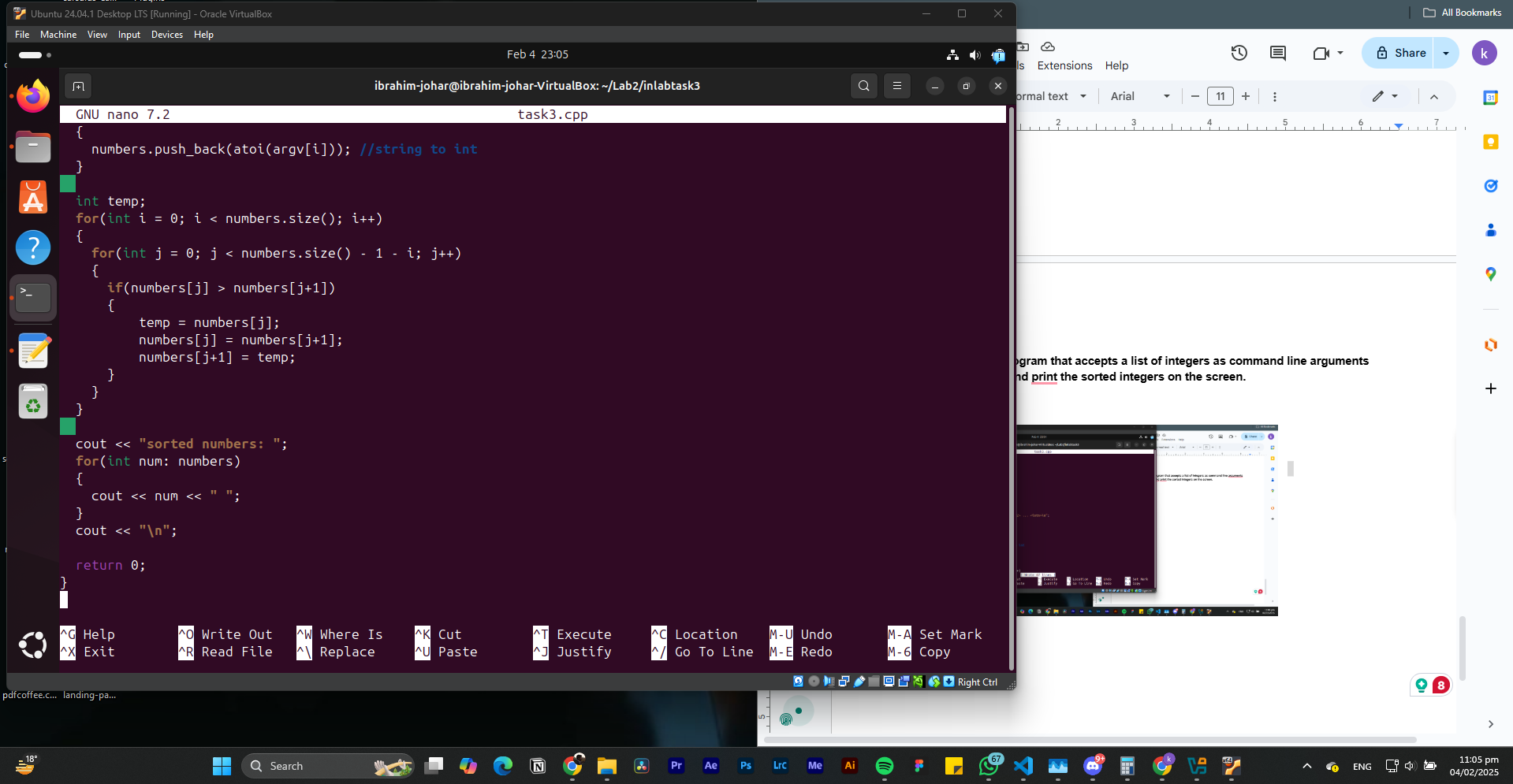
task2.cpp

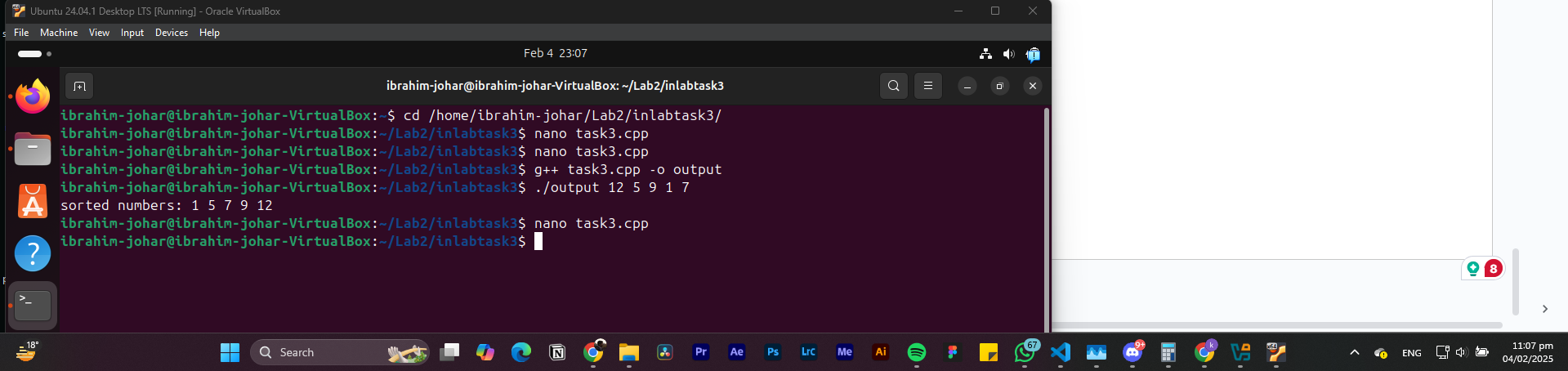
task2 console output:

**Task 3:**

**Write a C or C++ program that accepts a list of integers as command line arguments sorts the integers and print the sorted integers on the screen.**

task3.cpp



task3 output:

**Post-Lab Questions:**

**Task 1: Write a C/C++ program that takes some integers as command line parameters, store them in an array and prints the sum and average of that array. Also note that you have to run the program for all possible error checks.**

task1.cpp

//ibrahim johar farooqi

//23k-0074

//bai-4a

//os lab - post lab - task 1

#include <iostream>

#include <vector>

#include <cstdlib>

using namespace std;

int main(int argc, char\* argv[])

{

if(argc < 2)

{

cout << "error: no numbers provided\n";

cout << "usage: " << argv[0] << "<int1> <int2> ... <intN>\n";

return 1;

}

vector<int> numbers;

int sum = 0;

for(int i = 1; i < argc; i++)

{

bool isnum = true;

for(int j = 0; argv[i][j] != '\0'; j++)

{

if(!(argv[i][j] >= '0' && argv[i][j] <= '9') && !(argv[i][j] == '-' && j == 0))

{

isnum = false;

break;

}

}

if(!isnum)

{

cout << "error: '" << argv[i] << "' is not valid integer\n";

return 1;

}

int num = atoi(argv[i]); //string to int

numbers.push\_back(num);

sum += num;

}

double average = static\_cast<double>(sum) / numbers.size();

cout << "numbers: ";

for (int num : numbers) { cout << num << " "; }

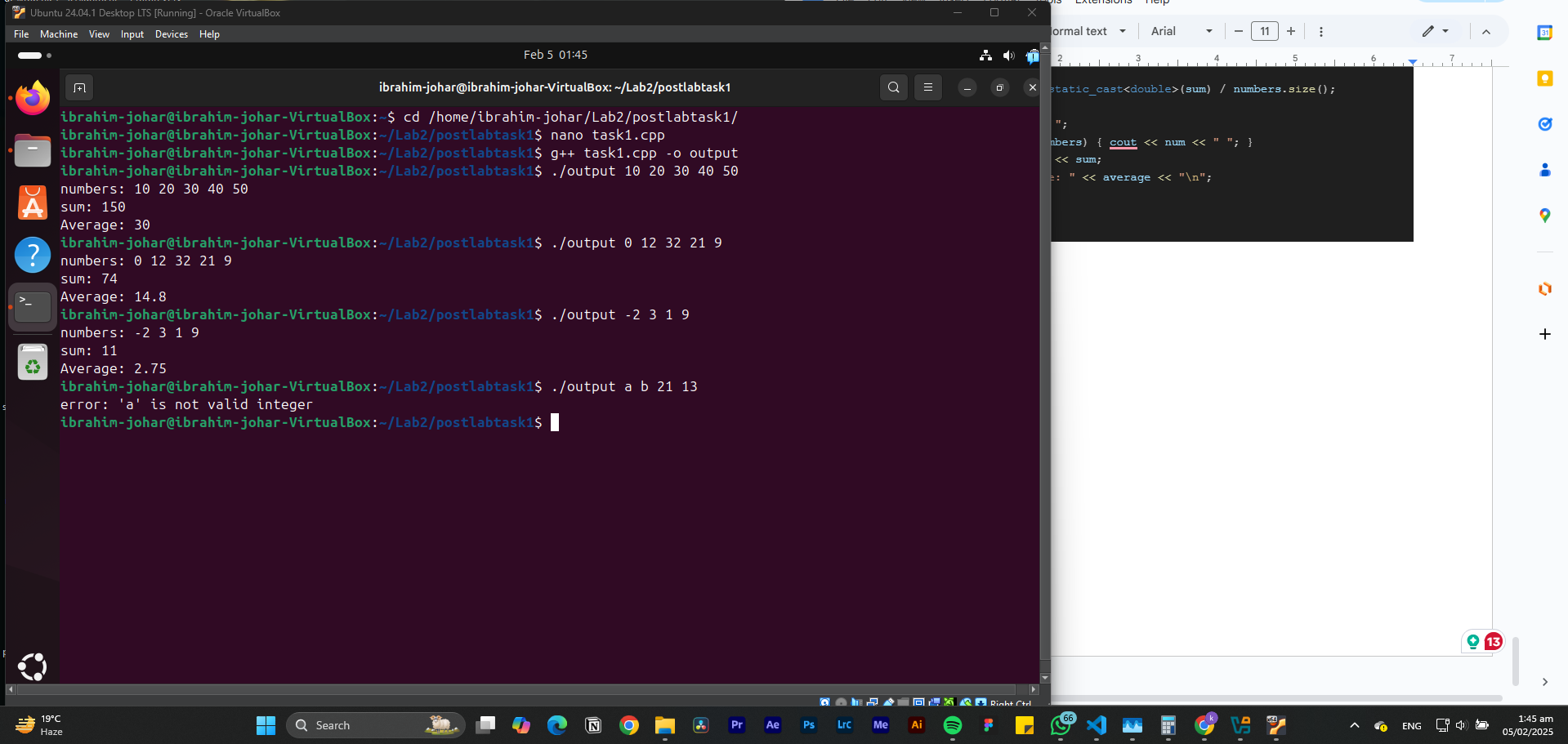
cout << "\nsum: " << sum;

cout << "\nAverage: " << average << "\n";

return 0;

}

task1 output:



**Task 2:**

**Write a C/C++ program that takes some integers in the form of series as command line parameters; store them in array than compute the missing element from that series and output that missing element to file.**

**//ibrahim johar farooqi**

**//23k-0074**

**//bai-4a**

**//os lab - post lab - task 2**

**#include <iostream>**

**#include <vector>**

**#include <fstream>**

**#include <algorithm>**

**#include <cstdlib>**

**using namespace std;**

**int missingnum(vector<int> &numbers)**

**{**

**sort(numbers.begin(), numbers.end()); //sort**

**for (int i = 0; i < numbers.size() - 1; i++)**

**{**

**if (numbers[i] + 1 != numbers[i + 1])**

**{**

**return numbers[i] + 1; //missing num found**

**}**

**}**

**return -1; //no missing num found**

**}**

**int main(int argc, char \*argv[])**

**{**

**if (argc < 3)**

**{**

**cout << "error: please provide a series that has atleast 2 numbers.\n";**

**cout << "usage: " << argv[0] << " <int1> <int2> ... <intN>\n";**

**return 1;**

**}**

**vector<int> numbers;**

**for (int i = 1; i < argc; i++)**

**{**

**numbers.push\_back(atoi(argv[i])); //string to integer**

**}**

**int missing = missingnum(numbers);**

**//write missing num to file**

**ofstream outputfile("output.txt");**

**if (!outputfile)**

**{**

**cout << "couldnt open file for writing.\n";**

**return 1;**

**}**

**if (missing == -1)**

**{**

**outputfile << "no missing number found in sequence.\n";**

**cout << "no missing number found.\n";**

**}**

**else**

**{**

**outputfile << "missing number: " << missing << "\n";**

**cout << "missing number found & saved in output.txt: " << missing << "\n";**

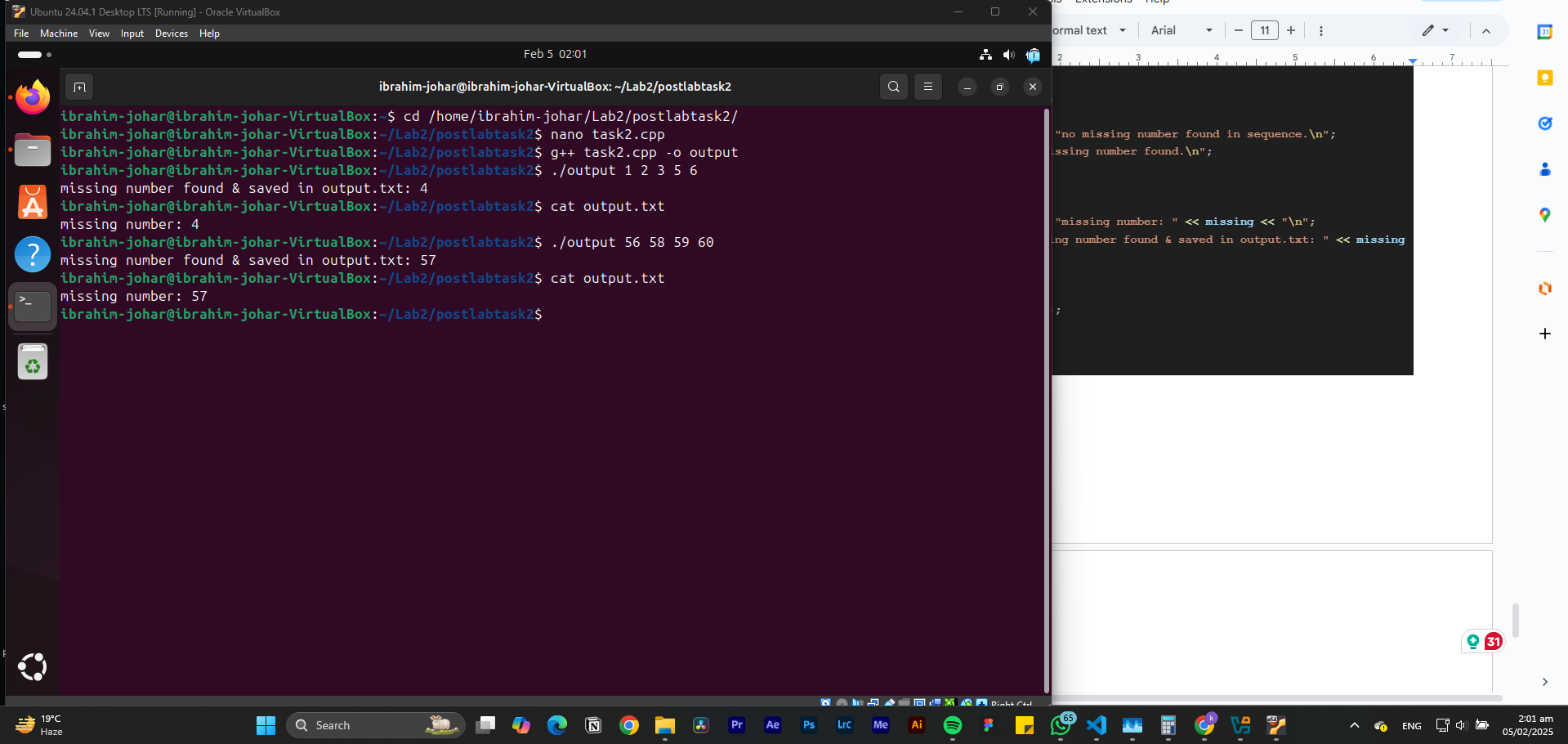
**}**

**outputfile.close();**

**return 0;**

**}**

task2 output:



**Task 3:**

**Write a C/C++ program that reads file in which there are integers related to series and store them in array than compute the missing element from that series and output that missing element to file.**

**//ibrahim johar farooqi**

**//23k-0074**

**//bai-4a**

**//os lab - post lab - task 3**

**#include <iostream>**

**#include <vector>**

**#include <fstream>**

**#include <algorithm>**

**using namespace std;**

**int missingnum(vector<int> &numbers)**

**{**

**sort(numbers.begin(), numbers.end()); //sort**

**for (int i = 0; i < numbers.size() - 1; i++)**

**{**

**if (numbers[i] + 1 != numbers[i + 1])**

**{**

**return numbers[i] + 1; //missing num found**

**}**

**}**

**return -1; //no missing num found**

**}**

**int main()**

**{**

**string filename;**

**cout << "enter filename: ";**

**cin >> filename;**

**ifstream file1(filename);**

**if (!file1)**

**{**

**cout << "file not found or cant be opened.\n";**

**return 1;**

**}**

**vector<int> numbers;**

**int num;**

**//read file**

**while (file1 >> num)**

**{**

**numbers.push\_back(num);**

**}**

**file1.close();**

**if (numbers.size() < 2)**

**{**

**cout << "insufficient numbers in the file.\n";**

**return 1;**

**}**

**int missing = missingnum(numbers);**

**ofstream file2("output.txt");**

**if (!file2)**

**{**

**cout << "couldnt open file for writing.\n";**

**return 1;**

**}**

**if (missing == -1)**

**{**

**file2 << "no missing number found in sequence.\n";**

**cout << "no missing number found.\n";**

**}**

**else**

**{**

**file2 << "missing number: " << missing << "\n";**

**cout << "missing number found & saved in output.txt: " << missing << "\n";**

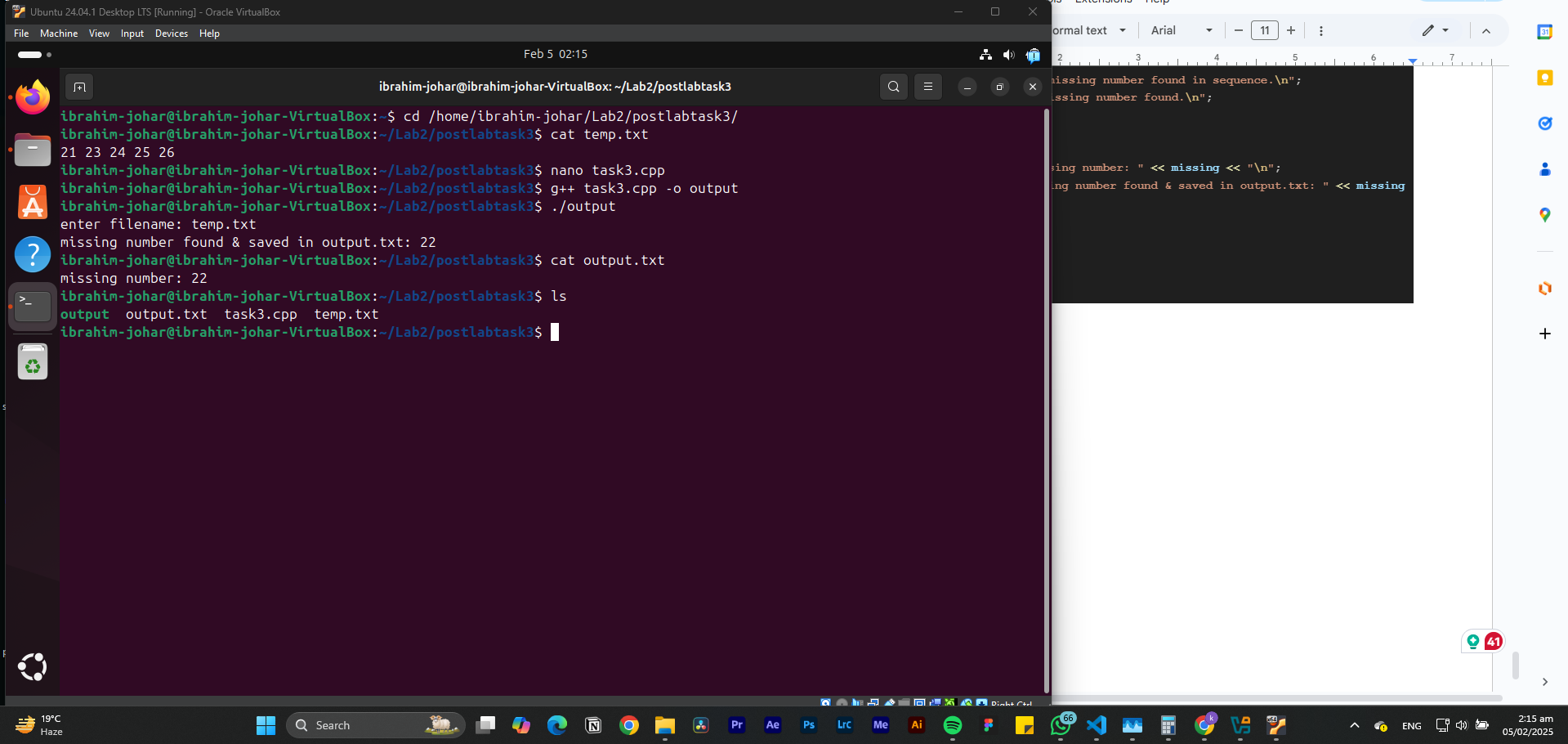
**}**

**file2.close();**

**return 0;**

**}**

task3 output:



**Task 4:**

**Create the following classes in separate files (using .h and .cpp files) LetterCount, WordCount, LineCount.**

**a. LetterCount counts number of letters in a text file.**

**b. WordCount counts number of words in a text file.**

**c. LineCount counts number of lines in a text file.**

**Create some objects of all classes in main function and populate them with data. Now compile all classes using makefile.**

task4 .h files:

#ifndef LETTER\_COUNT\_H

#define LETTER\_COUNT\_H

#include <string>

using namespace std;

class LetterCount

{

public:

int count\_letters(const string &filename);

};

#endif

#ifndef WORD\_COUNT\_H

#define WORD\_COUNT\_H

#include <string>

using namespace std;

class WordCount

{

public:

int count\_words(const string &filename);

};

#endif

#ifndef LINE\_COUNT\_H

#define LINE\_COUNT\_H

#include <string>

using namespace std;

class LineCount

{

public:

int count\_lines(const string &filename);

};

#endif

task4 .cpp files:

#include "LetterCount.h"

#include <iostream>

#include <fstream>

#include <cctype>

using namespace std;

int LetterCount::count\_letters(const string &filename)

{

ifstream file1(filename);

if (!file1)

{

cout << "cant open file.\n";

return -1;

}

char ch;

int letterCount = 0;

while (file1.get(ch))

{

if (isalpha(ch)) letterCount++; //only count letters

}

file1.close();

return letterCount;

}

#include "WordCount.h"

#include <iostream>

#include <fstream>

using namespace std;

int WordCount::count\_words(const string &filename)

{

ifstream file1(filename);

if (!file1)

{

cout << "cant open file.\n";

return -1;

}

string word;

int wordCount = 0;

while (file1 >> word)

{

wordCount++; //count words only (identified by whitespaces)

}

file1.close();

return wordCount;

}

#include "LineCount.h"

#include <iostream>

#include <fstream>

using namespace std;

int LineCount::count\_lines(const string &filename)

{

ifstream file1(filename);

if (!file1)

{

cout << "cant open file.\n";

return -1;

}

string line;

int lineCount = 0;

while (getline(file1, line))

{

lineCount++; //counting every line

}

file1.close();

return lineCount;

}

//ibrahim johar farooqi

//23k-0074

//bai-4a

//os lab - post lab - task 4

#include <iostream>

#include "LetterCount.h"

#include "WordCount.h"

#include "LineCount.h"

using namespace std;

int main()

{

string filename = "task4output.txt";

LetterCount letter\_counter;

WordCount word\_counter;

LineCount line\_counter;

int letters = letter\_counter.count\_letters(filename);

int words = word\_counter.count\_words(filename);

int lines = line\_counter.count\_lines(filename);

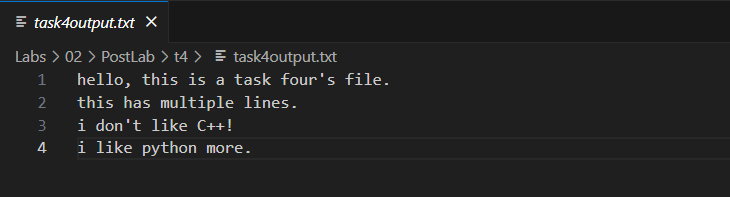
if (letters != -1) cout << "letters: " << letters << "\n";

if (words != -1) cout << "words: " << words << "\n";

if (lines != -1) cout << "lines: " << lines << "\n";

return 0;

}

task4output.txt

makefile:

#compiler

CC = g++

CFLAGS = -Wall -std=c++11 -Iinclude

#directories

SRC\_DIR = src

OBJ\_DIR = obj

BIN\_DIR = bin

#source & object files

SRCS = $(SRC\_DIR)/main.cpp $(SRC\_DIR)/LetterCount.cpp $(SRC\_DIR)/WordCount.cpp $(SRC\_DIR)/LineCount.cpp

OBJS = $(OBJ\_DIR)/main.o $(OBJ\_DIR)/LetterCount.o $(OBJ\_DIR)/WordCount.o $(OBJ\_DIR)/LineCount.o

#output executable

TARGET = $(BIN\_DIR)/counter

#rule to create the executable

all: $(TARGET)

$(TARGET): $(OBJS) | $(BIN\_DIR)

$(CC) -o $(TARGET) $(OBJS) $(CFLAGS)

#rule for compiling .cpp files into .o files

$(OBJ\_DIR)/%.o: $(SRC\_DIR)/%.cpp | $(OBJ\_DIR)

$(CC) -c $< -o $@ $(CFLAGS)

#ensure that required directories exist

$(OBJ\_DIR):

mkdir -p $(OBJ\_DIR)

$(BIN\_DIR):

mkdir -p $(BIN\_DIR)

#the clean rule

clean:

rm -rf $(OBJ\_DIR) $(BIN\_DIR)

task4 output:

