MUHAMMAD MUDASSAR ANWAR

ME-15

SEC-A

456925

LAB MANUAL 10

25 DEC 2023

TASK 1:

#include <iostream>

#include <vector>

using namespace std;

int main() {

vector<int> myVector;

for (int i = 1; i <= 10; ++i) {

myVector.push\_back(i);

}

cout << "Original Vector: ";

for (vector<int>::iterator it = myVector.begin(); it != myVector.end(); ++it) {

cout << \*it << " ";

}

cout << endl;

myVector.push\_back(11);

if (!myVector.empty()) {

vector<int>::iterator removeIterator = myVector.begin() + 1;

myVector.erase(removeIterator);

}

cout << "Modified Vector: ";

for (int element : myVector) {

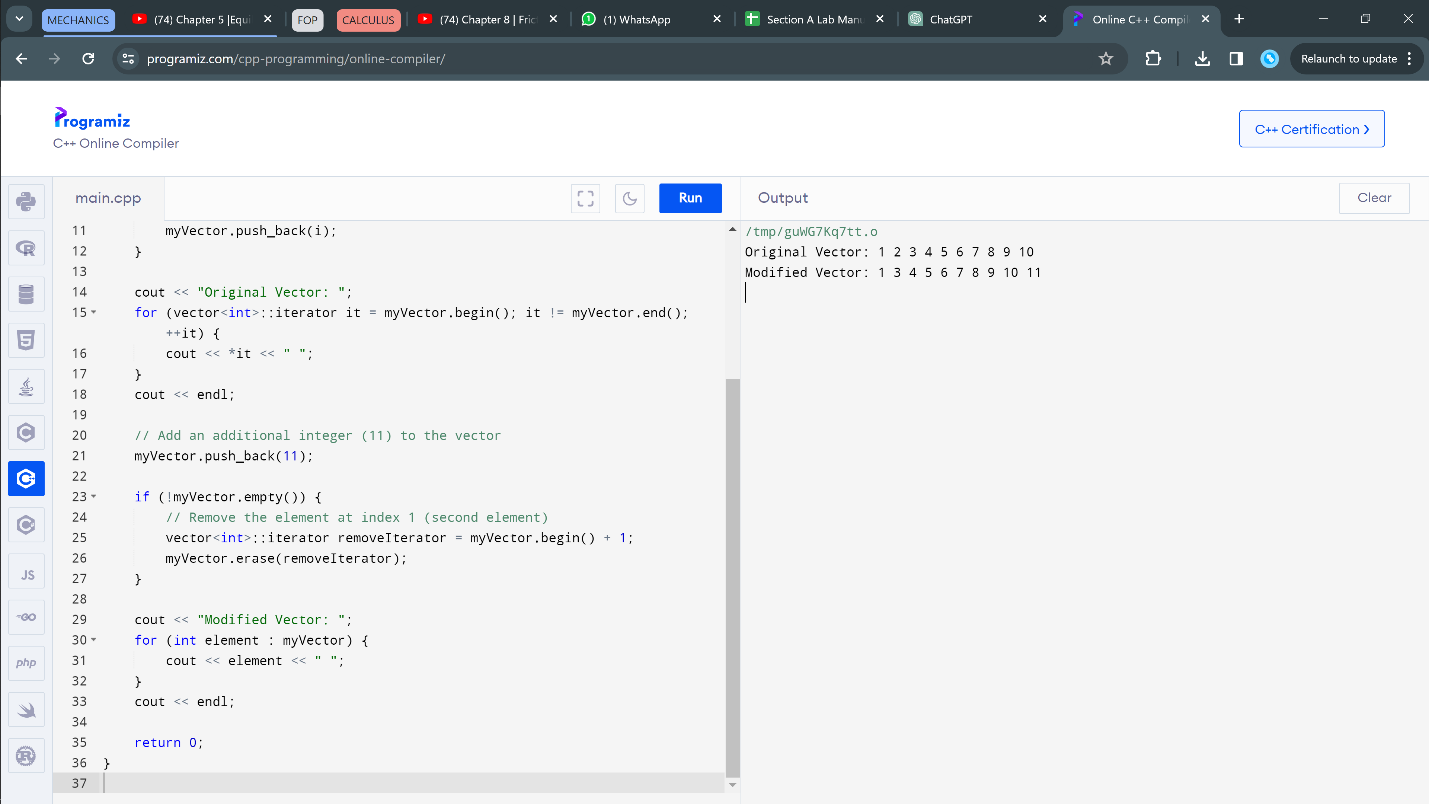
cout << element << " ";

}

cout << endl;

return 0;

}



TASK 2:

#include <iostream>

#include <vector>

#include <algorithm>

#include <map>

using namespace std;

double calculateMean(const vector<int>& grades) {

if (grades.empty()) {

return 0.0;

}

int sum = 0;

for (int grade : grades) {

sum += grade;

}

return static\_cast<double>(sum) / grades.size();

}

double calculateMedian(vector<int>& grades) {

if (grades.empty()) {

return 0.0;

}

sort(grades.begin(), grades.end());

size\_t size = grades.size();

if (size % 2 == 0) {

return (grades[size / 2 - 1] + grades[size / 2]) / 2.0;

} else {

return grades[size / 2];

}

}

vector<int> calculateMode(const vector<int>& grades) {

map<int, int> countMap;

for (int grade : grades) {

countMap[grade]++;

}

int maxCount = 0;

for (const auto& pair : countMap) {

maxCount = max(maxCount, pair.second);

}

vector<int> modeGrades;

for (const auto& pair : countMap) {

if (pair.second == maxCount) {

modeGrades.push\_back(pair.first);

}

}

return modeGrades;

}

int main() {

int numPairs;

cout << "Enter the number of name/grade pairs: ";

cin >> numPairs;

vector<string> names;

vector<int> grades;

// Input names and grades

for (int i = 0; i < numPairs; ++i) {

string name;

int grade;

cout << "Enter name #" << i + 1 << ": ";

cin >> name;

cout << "Enter grade #" << i + 1 << ": ";

cin >> grade;

names.push\_back(name);

grades.push\_back(grade);

}

cout << "Mean of the grades: " << calculateMean(grades) << endl;

cout << "Median of the grades: " << calculateMedian(grades) << endl;

vector<int> modeGrades = calculateMode(grades);

cout << "Mode of the grades: ";

for (int grade : modeGrades) {

cout << grade << " ";

}

cout << endl;

cout << "Names of students with the mode grade: ";

for (size\_t i = 0; i < grades.size(); ++i) {

if (find(modeGrades.begin(), modeGrades.end(), grades[i]) != modeGrades.end()) {

cout << names[i] << " ";

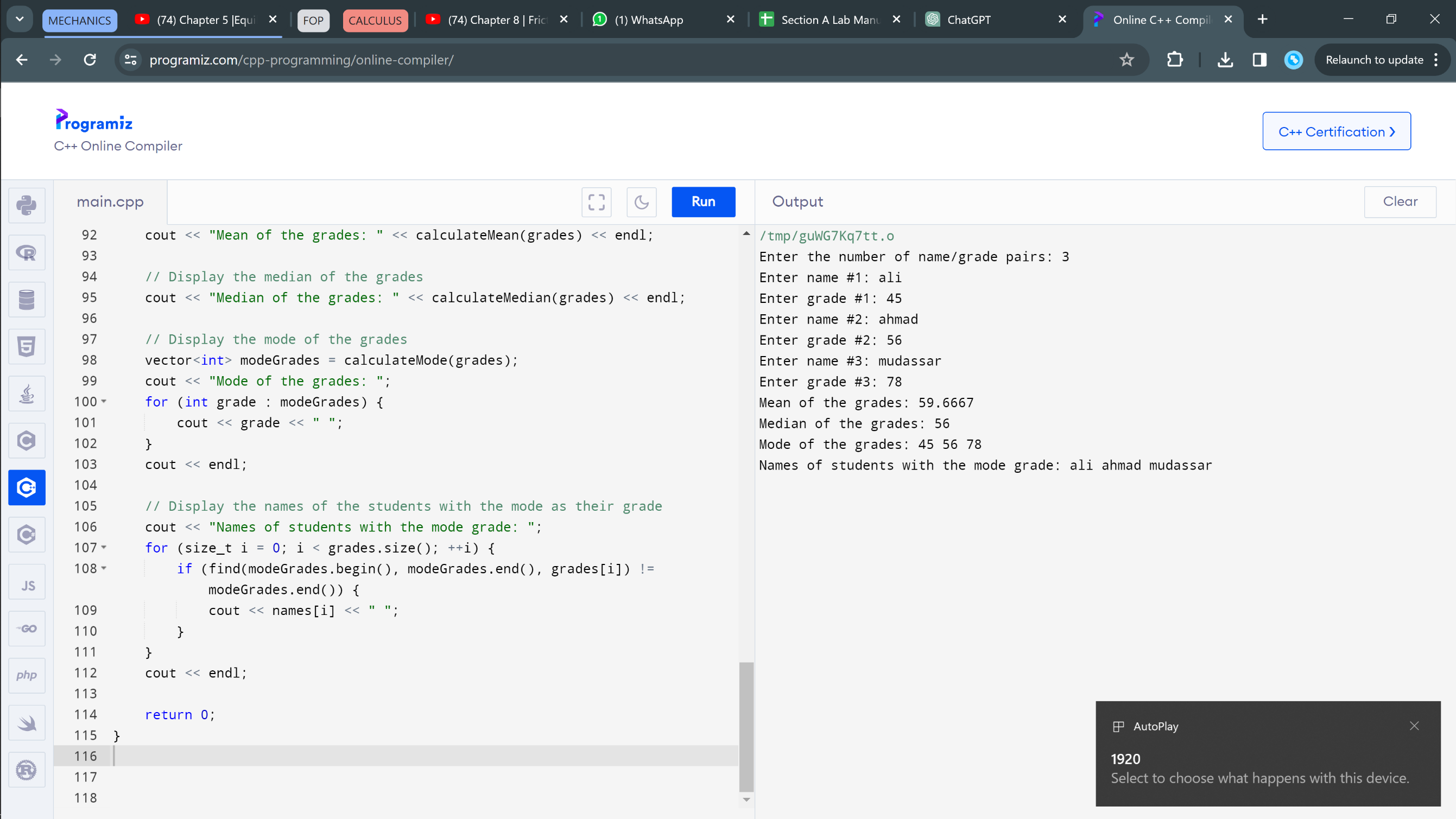
}

}

cout << endl;

return 0;

}



TASK 3:

#include <iostream>

#include <cmath>

using namespace std;

class Triangle {

private:

double side1, side2, side3;

public:

Triangle(double s1, double s2, double s3) : side1(s1), side2(s2), side3(s3) {}

double calculatePerimeter() const {

return side1 + side2 + side3;

}

double calculateArea() const {

double s = calculatePerimeter() / 2.0;

return sqrt(s \* (s - side1) \* (s - side2) \* (s - side3));

}

void printInfo() const {

cout << "Triangle with sides: " << side1 << " m, " << side2 << " m, " << side3 << " m" << endl;

cout << "Perimeter: " << calculatePerimeter() << " m" << endl;

cout << "Area: " << calculateArea() << " square meters" << endl;

}

};

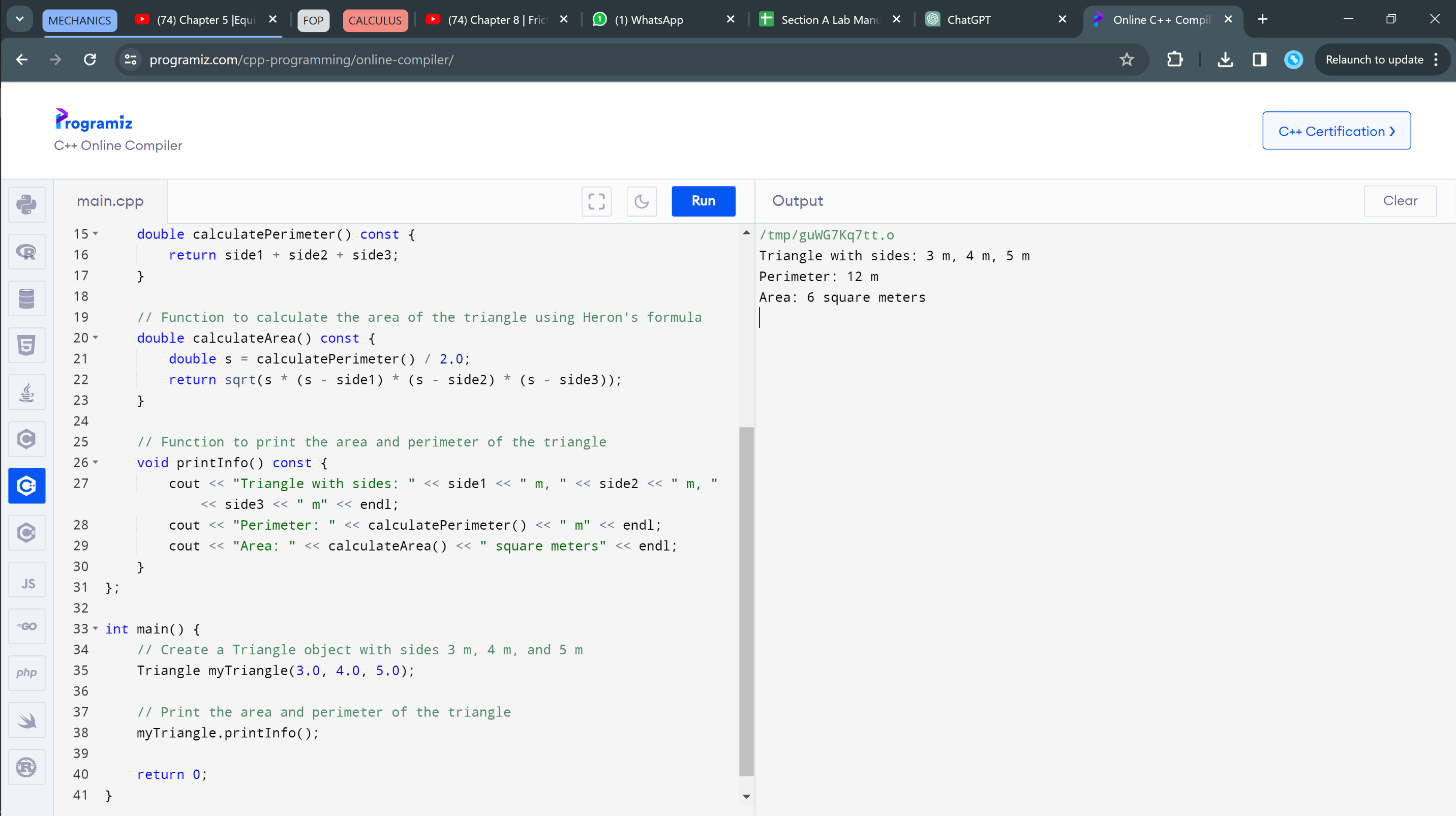
int main() {

Triangle myTriangle(3.0, 4.0, 5.0);

myTriangle.printInfo();

return 0;

}



TASK 4:

#include <iostream>

#include <string>

using namespace std;

struct Employee {

string name;

double salary;

int hoursOfWorkPerDay;

};

void increaseSalary(Employee& employee) {

if (employee.hoursOfWorkPerDay >= 12) {

employee.salary += 150;

} else if (employee.hoursOfWorkPerDay >= 10) {

employee.salary += 100;

} else if (employee.hoursOfWorkPerDay >= 8) {

employee.salary += 50;

}

}

int main() {

Employee employees[10];

for (int i = 0; i < 10; ++i) {

cout << "Enter name for employee #" << i + 1 << ": ";

cin >> employees[i].name;

cout << "Enter salary for employee #" << i + 1 << ": ";

cin >> employees[i].salary;

cout << "Enter hours of work per day for employee #" << i + 1 << ": ";

cin >> employees[i].hoursOfWorkPerDay;

increaseSalary(employees[i]);

}

cout << "\nFinal Salaries:\n";

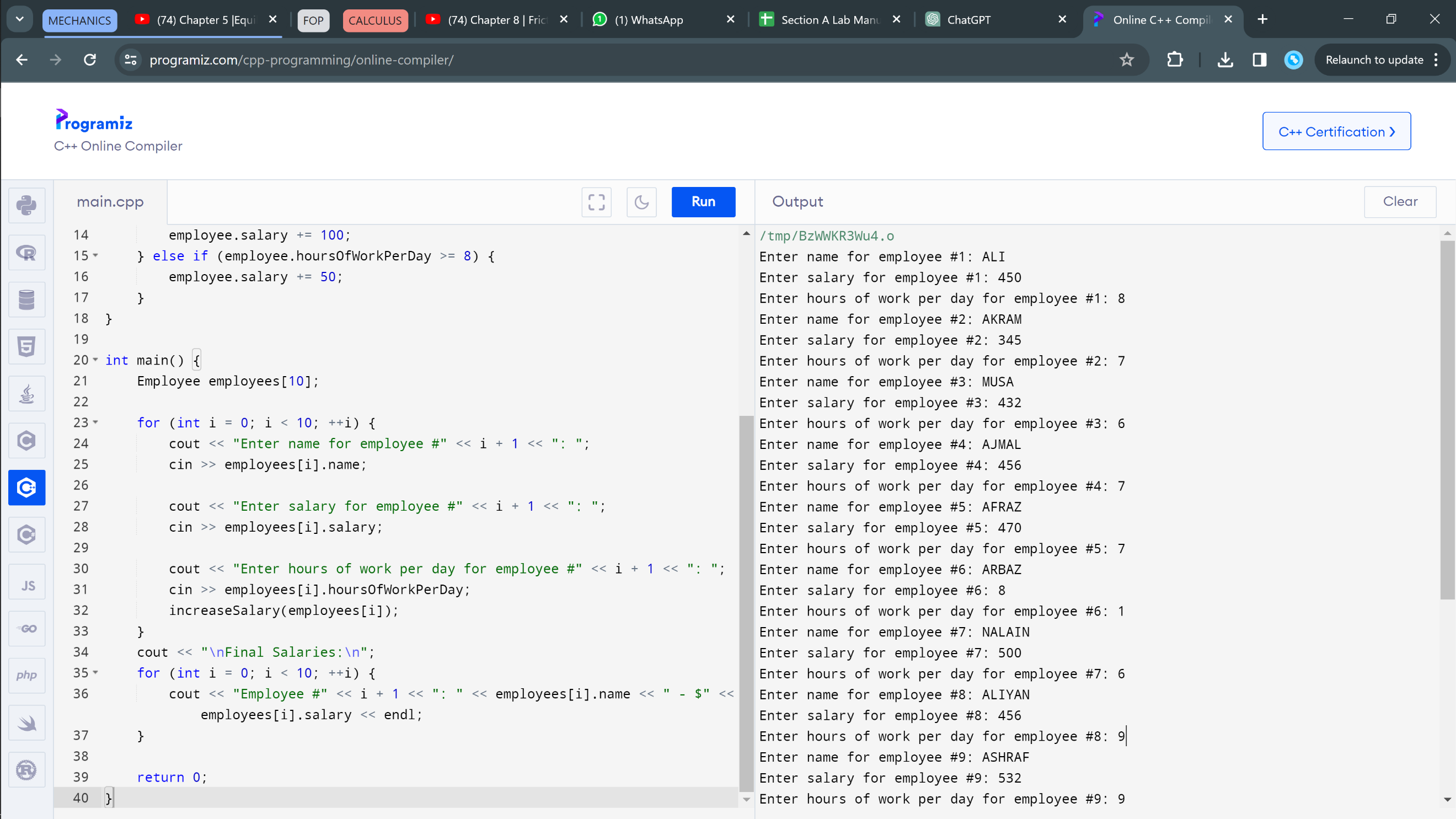
for (int i = 0; i < 10; ++i) {

cout << "Employee #" << i + 1 << ": " << employees[i].name << " - $" << employees[i].salary << endl;

}

return 0;

}



A screenshot of a computer

Description automatically generated