

**NUST**

**School of Mechanical & Manufacturing Engineering**

**FOP Lab Manual-10:**

**Name:** Muhammad Muzammil Riaz

**Qalam:** 467817

**Batch:** ME-15

**Section:** A

**Course Instructor:** Dr. Jawad

**Lab instructor:** Sir. Affan

**Qno.1:**

#include <iostream>

#include <vector>

int main() {

std::vector<int> myVector;

myVector.push\_back(1);

myVector.push\_back(2);

myVector.push\_back(3);

myVector.push\_back(4);

std::cout << "Original Vector: ";

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

std::cout << \*it << " ";

}

std::cout << std::endl;

myVector.push\_back(5);

if (!myVector.empty()) {

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

myVector.erase(removeIterator);

}

std::cout << "Modified Vector: ";

for (int element : myVector) {

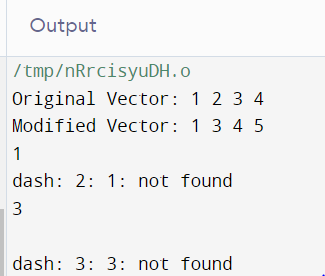
std::cout << element << " ";

}

std::cout << std::endl;

return 0;

}



**Qno.2:**

#include <iostream>

#include <vector>

#include <algorithm>

#include <numeric>

using namespace std;

int main() {

int numPairs;

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

cin >> numPairs;

vector<string> names(numPairs);

vector<int> grades(numPairs);

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

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

cin >> names[i];

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

cin >> grades[i];

}

double mean = accumulate(grades.begin(), grades.end(), 0.0) / grades.size();

cout << "Mean grade: " << mean << endl;

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

double median = grades.size() % 2 == 0 ? (grades[grades.size() / 2 - 1] + grades[grades.size() / 2]) / 2.0 : grades[grades.size() / 2];

cout << "Median grade: " << median << endl;

int mode = grades[0], modeCount = 1;

int currentCount = 1;

for (int i = 1; i < grades.size(); ++i) {

if (grades[i] == grades[i - 1]) {

currentCount++;

} else {

if (currentCount > modeCount) {

mode = grades[i - 1];

modeCount = currentCount;

}

currentCount = 1;

}

}

if (currentCount > modeCount) {

mode = grades[grades.size() - 1];

}

cout << "Mode grade: " << mode << endl;

cout << "Students with the mode grade: ";

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

if (grades[i] == mode) {

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

}

}

cout << endl;

return 0;

}

A computer screen with white text

Description automatically generated

**Qno.3:**

#include <iostream>

#include <cmath>

using namespace std;

class Triangle {

public:

double side1, side2, side3;

Triangle(double a, double b, double c) {

side1 = a;

side2 = b;

side3 = c;

}

void printArea() {

double s = (side1 + side2 + side3) / 2.0;

double area = sqrt(s \* (s - side1) \* (s - side2) \* (s - side3));

cout << "Area of the triangle: " << area << " sq. m\n";

}

void printPerimeter() {

double perimeter = side1 + side2 + side3;

cout << "Perimeter of the triangle: " << perimeter << " m\n";

}

};

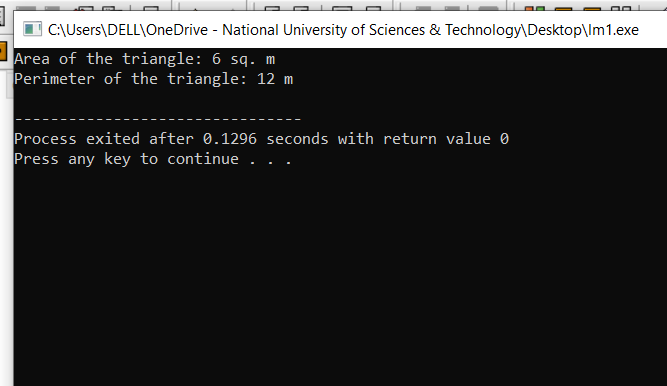
int main() {

triangle.printArea();

triangle.printPerimeter();

return 0;

}



**Qno.4:**

#include <iostream>

#include <vector>

using namespace std;

struct Employee {

string name;

double salary;

int hoursPerDay;

};

int main() {

vector<Employee> employees(10);

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

cout << "Enter details for employee " << i + 1 << ":" << endl;

cout << "Name: ";

cin >> employees[i].name;

cout << "Salary: $";

cin >> employees[i].salary;

cout << "Hours of work per day: ";

cin >> employees[i].hoursPerDay;

}

for (Employee& employee : employees) {

if (employee.hoursPerDay == 8) {

employee.salary += 50;

} else if (employee.hoursPerDay == 10) {

employee.salary += 100;

} else if (employee.hoursPerDay >= 12) {

employee.salary += 150;

}

}

cout << "\nEmployee Details with Final Salaries:\n";

for (const Employee& employee : employees) {

cout << "Name: " << employee.name << endl;

cout << "Final Salary: $" << employee.salary << endl;

cout << "------------------------------------\n";

}

return 0;

}