

|  |  |
| --- | --- |
| Name | Abdul Rafeh |
| Class | ME-15(C) |
| CMS | 417648 |

Q1:

#include<iostream> #include <vector>

usingnamespacestd;

intmain(){

vector<int>numbers={1,4,2,8,5};

//Iteratethroughthevectorusingiterators and print elements

cout<<"Originalvectorelements:";

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

cout<<\*it<<"";

***}***

// Push integer 5 numbers.push\_back(5);

//Removetheelementatthepositionofthe newly added 5

intindex\_to\_remove=numbers.size()-1;// Get the index of the last element

numbers.erase(numbers.begin()+ index\_to\_remove);

// Print the modified vector elements cout<<"\nModifiedvectorelements:"; for (int num : numbers) {

cout<<num<<"";

***}***

cout<<endl;

return0;

***}***

Q2:

#include <iostream> #include <vector> #include <string> #include<algorithm>

#include<unordered\_map> using namespace std;

//Functiontocalculatethemeanofavectorof integers

doublecalculateMean(constvector<int>& grades) {

intsum=0;

for(intgrade:grades){

sum+=grade;

***}***

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

***}***

//Functiontocalculatethemedianofavectorof integers

doublecalculateMedian(vector<int>&grades){ sort(grades.begin(), grades.end());

intsize=grades.size(); if (size % 2 == 0) {

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

/2.0;

}else{

returngrades[size/2];

***}***

***}***

//Functiontocalculatethemodeofavectorof

integers

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

unordered\_map<int,int>frequency; for (int grade : grades) {

frequency[grade]++;

***}***

vector<int> modes; intmaxFrequency=0;

for (const auto& pair : frequency) { if(pair.second>maxFrequency){

maxFrequency=pair.second; modes.clear();

modes.push\_back(pair.first);

}elseif(pair.second==maxFrequency){ modes.push\_back(pair.first);

***}***

***}***

returnmodes;

***}***

intmain(){

intnumPairs;

cout<<"Enterthenumberofname/grade pairs: ";

cin>>numPairs;

vector<string>names; vector<int> grades;

//Inputname/gradepairs

for(inti=0;i<numPairs;++i){ string name;

intgrade;

cout<<"Entername"<<i+1<<":"; cin >> name;

names.push\_back(name);

cout<<"Entergradefor"<<name<<":"; cin >> grade;

grades.push\_back(grade);

***}***

//Calculatemean

double mean = calculateMean(grades); cout<<"Meanofthegrades:"<<mean<<

endl;

//Calculatemedian

double median = calculateMedian(grades); cout<<"Medianofthegrades:"<<median<<

endl;

//Calculatemode

vector<int>modes=calculateMode(grades); cout << "Mode of the grades: ";

for(intmode:modes){

cout<<mode<<"";

***}***

cout<<endl;

//Displaynameswiththemodeastheirgrade cout << "Names with the mode as their grade:

***";***

for (size\_t i = 0; i < grades.size(); ++i) { if(find(modes.begin(),modes.end(),

grades[i]) != modes.end()) { cout<<names[i]<<"";

***}***

***}***

cout<<endl;

return0;

***}***

Q4:

#include<iostream> #include <string>

usingnamespacestd;

struct Employee { string name; double salary;inthoursWorked;

***};***

voidincreaseSalary(Employee&employee){ if (employee.hoursWorked >= 12) {

employee.salary+=150;

}elseif(employee.hoursWorked>=10){ employee.salary += 100;

}elseif(employee.hoursWorked>=8){ employee.salary += 50;

***}***

***}***

intmain(){

const int numEmployees = 10; Employeeemployees[numEmployees];

//Inputemployeeinformation

for(inti=0;i<numEmployees;++i){

cout<<"Enternameofemployee"<<i+1

***<<":";***

cin>>employees[i].name;

cout<<"Entersalaryofemployee"<< employees[i].name << ": ";

cin>>employees[i].salary;

cout<<"Enterhoursofworkperdayof employee " << employees[i].name << ": ";

cin>>employees[i].hoursWorked;

***}***

//Increasesalarybasedonhoursworkedper day

for(inti=0;i<numEmployees;++i){ increaseSalary(employees[i]);

***}***

//Displayemployees'namesandfinalsalaries cout << "\nEmployees and their final salaries

afterincrease:\n";

for (int i = 0; i < numEmployees; ++i) { cout<<employees[i].name<<":$"<<

employees[i].salary<<endl;

***}***

return0;

***}***

Q3:

#include<iostream>

#include<cmath>

usingnamespacestd; class Triangle {

private:

doubleside1,side2,side3;

public:

Triangle(doubles1,doubles2,doubles3): side1(s1), side2(s2), side3(s3) {}

double calculatePerimeter() { returnside1+side2+side3;

***}***

doublecalculateArea(){

doubles=calculatePerimeter()/2.0;

returnsqrt(s\*(s-side1)\*(s-side2)\*(s-

side3));

***}***

voidprintDetails(){

cout<<"Trianglewithsides:"<<side1<<" m, " << side2 << " m, " << side3 << " m\n";

cout<<"Perimeter:"<<calculatePerimeter()

<<"m\n";

cout<<"Area:"<<calculateArea()<<" square meters\n";

***}***

***};***

intmain(){

doubleside1=3.0,side2=4.0,side3=5.0;

Triangletriangle(side1,side2,side3); triangle.printDetails();

return0;

***}***