Fundamentals Of Programming

Lab Manual: 10

Submitted to: Sir M. Affan

Submitted By: Nafeel Javaid

Roll No: 465554

Course: ME-15

Section: A

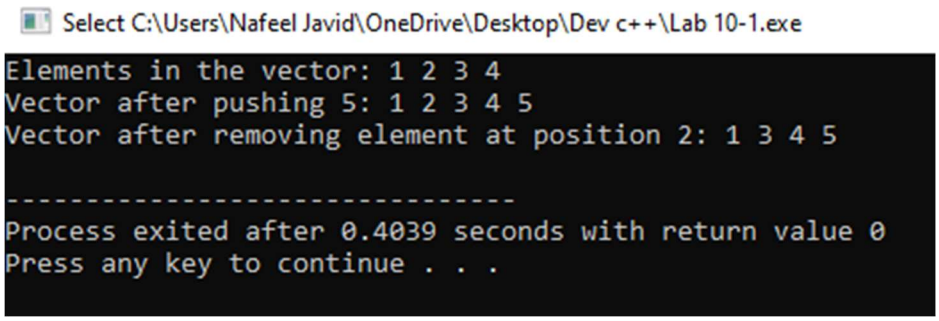
Task:01

- #include <iostream>

```
#include <vector>

using namespace std;

int main() {

   vector<int> myVector;

   for (int i = 1; i <= 4; ++i)

      myVector.push_back(i);

   cout << "Elements in the vector: ";

   for (auto y = myVector.begin(); y != myVector.end(); ++y)

      cout << *y << " ";
```

```
    cout << endl;

      myVector.push_back(5);

    cout << "Vector after pushing 5: ";

for (auto y = myVector.begin(); y != myVector.end(); ++y)

    cout << *y << " ";

    cout << endl;

if (!myVector.empty() && myVector.size() > 2) {

    auto itToRemove = myVector.begin() + 1;

    myVector.erase(itToRemove);

}

 cout << "Vector after removing element at position 2: ";

for (auto y = myVector.begin(); y != myVector.end(); ++y)

    cout << *y << " ";
```

```
    cout <<endl;


   return 0;

}
```



Task: 02

- #include <iostream>

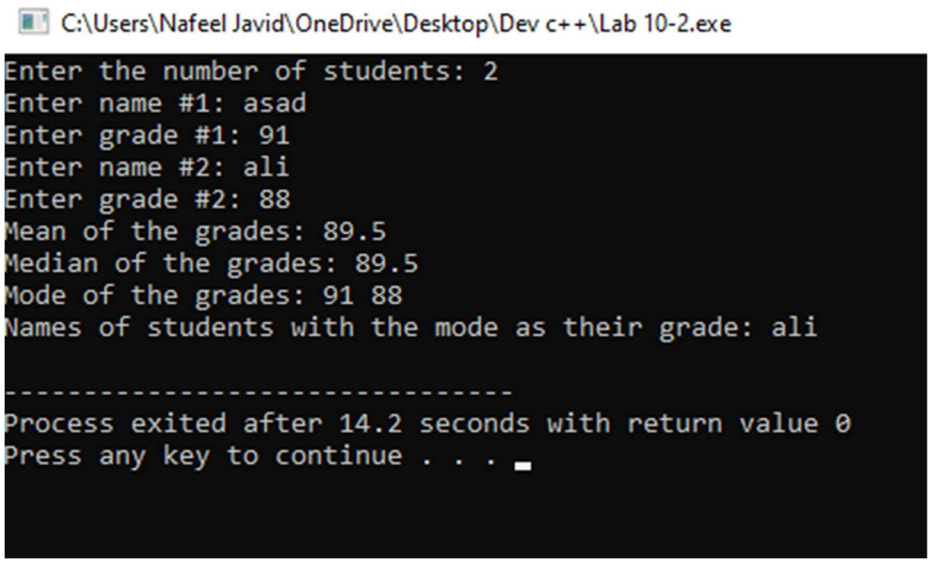
```
#include <vector>

#include <algorithm>
```

```
#include <unordered_map>

#include<numeric>

using namespace std;

int main() {

    int numStudents;

    cout << "Enter the number of students: ";

    cin >> numStudents;

    vector<string> names;

    vector<int> grades;

    for (int i = 0; i < numStudents; ++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);

}

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

cout << "Mean of the grades: " << mean << endl;

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

double median;

if (numStudents % 2 == 0) {

    median = (grades[numStudents / 2 - 1] + grades[numStudents / 2]) / 2.0;

} else {
```

```
    median = grades[numStudents / 2];

}

cout << "Median of the grades: " << median << endl;

unordered_map<int, int> frequencyMap;

int maxFrequency = 0;

for (int grade : grades) {

    maxFrequency = max(maxFrequency, ++frequencyMap[grade]);

}

cout << "Mode of the grades: ";

for (const auto& entry : frequencyMap) {

    if (entry.second == maxFrequency) {

        cout << entry.first << " ";

    }
```

```
    }

    cout << endl;

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

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

        if (grades[i] == frequencyMap.begin()->first) {

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

        }

    }

    cout << endl;

    return 0;

}
```



Task: 03

- #include <iostream>

#include <cmath>

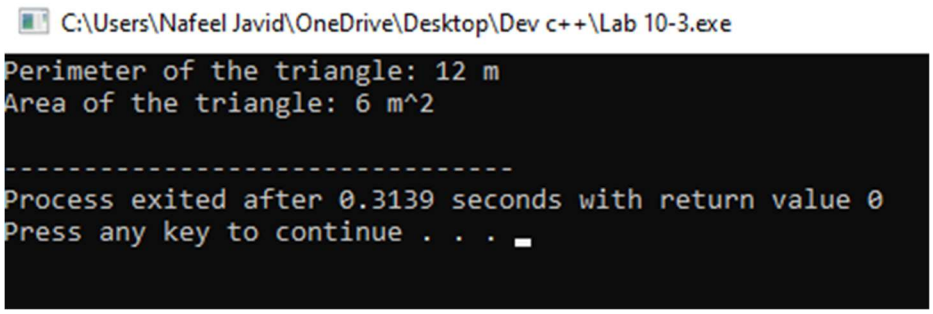
using namespace std;

class Triangle {

private:

```
 double a, b, c;

public:

 Triangle(double side1, double side2, double side3) {

    a = side1;

    b = side2;

    c = side3;

 }

                double getPerimeter() {

    return a + b + c;

 }

 double getArea() {

    double s = (a + b + c) / 2;

    return sqrt(s * (s - a) * (s - b) * (s - c));
```

```
    }

};

  int main() {

   Triangle t(3, 4, 5);

   cout << "Perimeter of the triangle: " << t.getPerimeter() << " m" << endl;

   cout << "Area of the triangle: " << t.getArea() << " m^2" << std::endl;

   return 0;

}
```



Task: 04

- #include <iostream>

#include <cmath>

using namespace std;

struct Employee {

    std::string name;

    double salary;

    int hours_worked;

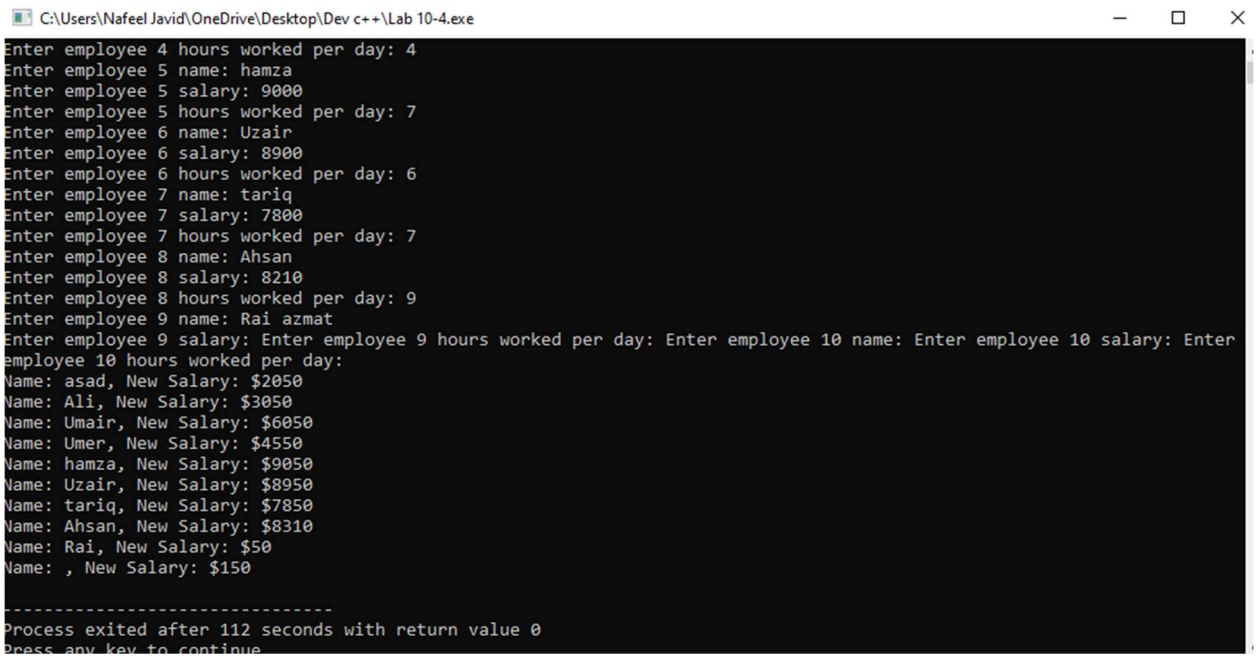
};

void increaseSalary(Employee& employee, int hours_worked) {

    if (hours_worked <= 8) {

```
        employee.salary += 50;
    } else if (hours_worked <= 10) {
        employee.salary += 100;
    } else {
        employee.salary += 150;
    }
}
int main() {
    Employee employees[10];
    for (int i = 0; i < 10; i++) {
        cout << "Enter employee " << i + 1 << " name: ";
        cin >> employees[i].name;
        cout << "Enter employee " << i + 1 << " salary: ";
```

```
        cin >> employees[i].salary;

        cout << "Enter employee " << i + 1 << " hours worked per day: ";

        cin >> employees[i].hours_worked;

    }

      cout <<endl;

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

        increaseSalary(employees[i], employees[i].hours_worked);

        cout << "Name: " << employees[i].name << ", New Salary: $" << employees[i].salary <<endl;

    }

    return 0;

}
```


C:\Users\Nafeel Javid\OneDrive\Desktop\Dev c++\Lab 10-4.exe
Enter employee 4 hours worked per day: 4
Enter employee 5 name: hamza
Enter employee 5 salary: 9000
Enter employee 5 hours worked per day: 7
Enter employee 6 name: Uzair
Enter employee 6 salary: 8900
Enter employee 6 hours worked per day: 6
Enter employee 7 name: tariq
Enter employee 7 salary: 7800
Enter employee 7 hours worked per day: 7
Enter employee 8 name: Ahsan
Enter employee 8 salary: 8210
Enter employee 8 hours worked per day: 9
Enter employee 9 name: Rai azmat
Enter employee 9 salary: Enter employee 9 hours worked per day: Enter employee 10 name: Enter employee 10 salary: Enter employee 10 hours worked per day:
Name: asad, New Salary: $2050
Name: Ali, New Salary: $3050
Name: Umair, New Salary: $6050
Name: Umer, New Salary: $4550
Name: hamza, New Salary: $9050
Name: Uzair, New Salary: $8950
Name: tariq, New Salary: $7850
Name: Ahsan, New Salary: $8310
Name: Rai, New Salary: $50
Name: , New Salary: $150

Process exited after 112 seconds with return value 0
Press any key to continue