School Of Mechanical & Manufacturing Engineering, NUST

Department of Mechanical Engineering

CS-114 - Fundamentals of Programing

Course Instructor: Dr Jawad Khan

Lab Instructor: Sir Affan

Student Name: Juveriah Waqqas

CMS ID: 460510

ME-15/A

LAB MANUAL 10

TASKS

Q-No # 1

1. Iterate Through Vector Using Iterators and print all pushed elements. Next you need to push integer 5 and remove element at that position.

Code

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{

//defining vector and adding elements
    vector<int> myvector ;

    myvector.push_back(1); myvector.push_back(2); myvector.push_back(3);
    myvector.push_back(4); myvector.push_back(6); myvector.push_back(7);

//Displaying the elements
    cout<<"The elements of the Vector are : "<<endl;
    for (vector<int>::iterator it = myvector.begin(); it != myvector.end(); ++it)
    {
        cout << *it <<" ";
    }
    cout<<endl;

//Pushing 5 and displaying the elements
    myvector.push_back(5);
    cout<<"The elements after adding 5 are : "<<endl;
    for (vector<int>::iterator it = myvector.begin(); it != myvector.end(); ++it)
    {
        cout << *it <<" ";
    }
    cout<<endl;

//Removing element at pos 5 and displaying
    cout<<"The vector after removing integer at position 5 is : "<<endl;
    myvector.erase(myvector.begin()+4 );
    for (vector<int>::iterator it = myvector.begin(); it != myvector.end(); ++it)
    { cout << *it <<" ";    }

return 0;
}
```

Execution (Example)

```
The elements of the Vector are :
1 2 3 4 6 7
The elements after adding 5 are :
1 2 3 4 6 7 5
The vector after removing integer at position 5 is :
1 2 3 4 7 5
--------------------------------
Process exited after 0.1596 seconds with return value 0
Press any key to continue . . .
```

Q-No # 2

2. Write a complete C++ program that uses 2 vectors, 1 for names (string) and 1 for grades (int)
    a. Ask the user for the number of name/grade pairs that will be entered.
    b. Display the mean of the grades.
    c. Display the median of the grades.
    d. Display the mode of the grades.
    e. Display the names of the students with the mode as their grade.

Code

```cpp
#include<iostream>
#include<vector>
using namespace std;
int main()
{
    vector<string> names;
    vector<int> marks;

    cout<<"Enter the total number of Students : "<<endl;
    int students;
    cin>>students;

//inputting the students and marks
    for(int i=0; i<students; i++)
    {
        string name;
        int mark;
        cout<<"Student Name : ";
        cin>>name;
        cout<<"Student Marks : ";
        cin>>mark;
        while(mark > 100 || mark < 0)
        {
            cout<<"Invalid marks! Enter a valid value : ";
            cin>>mark;
        }
        names.push_back(name);
        marks.push_back(mark);
    }
```

```cpp
//calculating mean
    float mean;
    int sum=0;
    for(vector<int>::iterator i = marks.begin(); i !=marks.end(); i++)
    {
        sum += *i;
    }
    mean = static_cast<float>(sum) / students;
    cout<<"The Mean is : "<<mean<<endl;

//calculating median
    float median;
    for (int i = 0; i < students - 1; i++) {
        for (int j = 0; j < students - i - 1; j++) {
            if (marks[j] > marks[j + 1]) {
                swap(marks[j], marks[j + 1]);
                swap(names[j], names[j +1 ]);
            }
        }
    }
```

```
    if(students % 2 == 0)
    {
        median = (marks[students/2 -1] + marks[students/2]) / 2;
    }
    if(students % 2 != 0)
    {
        median = marks [students / 2 ];
    }
    cout<<"The Median is : "<<median<<endl;

//calculating mode
    int mode = -1,  modefrequency = 0, currentfrequency = 1;
    for(int i=0; i<students; i++)
    {
        if(marks[i] ==  marks[i+1])
        {
            currentfrequency += 1;
        }
        else
        {
            currentfrequency = 1;
        }
        if(currentfrequency > modefrequency)
        {
            mode = marks[i];
            modefrequency = currentfrequency;
        }
    }
    if(modefrequency >= 2)
    { cout<<"The Mode is : "<<mode<<endl;


//outputting the student whose marks are equal to mode
    cout<<"The students with mode as their marks are : ";
    for(int i=0; i<students; i++)
    {
        if(marks[i] == mode)
        {
            cout<<names[i]<<",";
        }
    }
    }

    else{
        cout<<"Since none of the values are repeated , therefore mode does not exist ";
    }
return 0;
}
```

Execution (Example)

```
Enter the total number of Students :
6
Student Name : Ayesha
Student Marks : 88
Student Name : Juveriah
Student Marks : 90
Student Name : Ahmed
Student Marks : 88
Student Name : Abdullah
Student Marks : 80
Student Name : Sana
Student Marks : 50
Student Name : Shumaila
Student Marks : 33
The Mean is : 71.5
The Median is : 84
The Mode is : 88
The students with mode as their marks are : Ayesha,Ahmed,
--------------------------------
Process exited after 44.79 seconds with return value 0
Press any key to continue . . .
```

Q-No # 3

3. Write a program to print the area and perimeter of a triangle having sides of 3 m, 4 m and 5 m by creating a class named 'Triangle' with a function to print the area and perimeter.

Code

```cpp
#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() {
        return side1 + side2 + side3;
    }

    double calculateArea() {
        double s = calculatePerimeter() / 2;
        return sqrt(s * (s - side1) * (s - side2) * (s - side3));
    }

    void printDetails() {
        cout << "Perimeter: " << calculatePerimeter() << " meters" << endl;
        cout << "Area: " << calculateArea() << " square meters" << endl;
    }
};

int main() {
    Triangle myTriangle(3, 4, 5);

    myTriangle.printDetails();

    return 0;
}
```

Execution

```
Perimeter: 12 meters
Area: 6 square meters

--------------------------------
Process exited after 0.1224 seconds with return value 0
Press any key to continue . . .
```

Q-No # 4

4. Write a structure to store the names, salary, and hours of work per day of 10 employees in a company. Write a program to increase the salary depending on the number of hours of work per day as follows and then print the name of all the employees along with their final salaries.

Hours of work per day	8	10	>=12
Increase in Salary	$50	$100	$150

Code

```cpp
#include <iostream>
#include <string>
#include <climits>

using namespace std;

struct Employee {
    string name;
    double salary;
    int hoursWorked;
};

void increaseSalary(Employee& emp) {
    if (emp.hoursWorked >= 12) {
        emp.salary += 150;
    } else if (emp.hoursWorked >= 10) {
        emp.salary += 100;
    } else if (emp.hoursWorked >= 8) {
        emp.salary += 50;
    }
}

int main() {
    const int numEmployees = 10;
    Employee employees[numEmployees];

    for (int i = 0; i < numEmployees; ++i) {
        cout << "Enter name of employee #" << i + 1 << ": ";
        getline(cin, employees[i].name);

        cout << "Enter salary of employee #" << i + 1 << ": ";
        cin >> employees[i].salary;

        cout << "Enter hours of work per day for employee #" << i + 1 << ": ";
        cin >> employees[i].hoursWorked;

        cin.ignore(INT_MAX, '\n');

        increaseSalary(employees[i]);
    }

    cout << "\nEmployee Details:\n";
    for (int i = 0; i < numEmployees; ++i) {
        cout << "Name: " << employees[i].name << "\tFinal Salary: $" << employees[i].salary << endl;
    }
return 0;
}
```

Execution (Example)

```
Enter name of employee #1: A
Enter salary of employee #1: 700
Enter hours of work per day for employee #1: 7
Enter name of employee #2: B
Enter salary of employee #2: 800
Enter hours of work per day for employee #2: 9
Enter name of employee #3: C
Enter salary of employee #3: 1200
Enter hours of work per day for employee #3: 10
Enter name of employee #4: D
Enter salary of employee #4: 1110
Enter hours of work per day for employee #4: 14
Enter name of employee #5: E
Enter salary of employee #5: 500
Enter hours of work per day for employee #5: 4
Enter name of employee #6: F
Enter salary of employee #6: 300
Enter hours of work per day for employee #6: 1.5
Enter name of employee #7: G
Enter salary of employee #7: 700
Enter hours of work per day for employee #7: 8
Enter name of employee #8: H
Enter salary of employee #8: 1200
Enter hours of work per day for employee #8: 15
Enter name of employee #9: I
Enter salary of employee #9: 1500
Enter hours of work per day for employee #9: 6
Enter name of employee #10: I
Enter salary of employee #10: 200
Enter hours of work per day for employee #10: 1
```

```
Employee Details:
Name: A Final Salary: $700
Name: B Final Salary: $850
Name: C Final Salary: $1300
Name: D Final Salary: $1260
Name: E Final Salary: $500
Name: F Final Salary: $300
Name: G Final Salary: $750
Name: H Final Salary: $1350
Name: I Final Salary: $1500
Name: I Final Salary: $200

--------------------------------
Process exited after 72.77 seconds with return value 0
Press any key to continue . . .
```