Fundamentals of Programing

Lab Manual # 10

**Lab Instructor:** Muhammad Affan

## Student Name: Muhammad Ali Shahzadah

## CMS ID: 466353

# Lab Manual # 10

**Vectors and Object Oriented Programming**

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

#include <bits/stdc++.h>

using namespace std;

int main() {

cout << "Enter elements for the vector (enter a non-integer to stop): "<<endl;

vector<int> myVector;

int num;

while (cin >> num) {

myVector.push\_back(num);

}

cout << "Vector elements: ";

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

cout << \*it << " ";

}

myVector.push\_back(5);

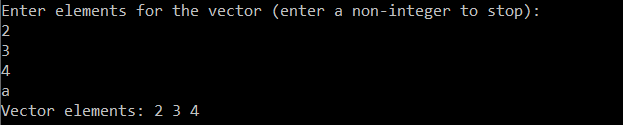
if (!myVector.empty() && myVector.size() >= 5) {

myVector.erase(myVector.begin() + 4);

}

return 0;

}



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

#include <iostream>

#include <vector>

using namespace std;

void bubbleSort(vector<int>& values) {

int n = values.size();

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (values[j] > values[j + 1]) {

int temp = values[j];

values[j] = values[j + 1];

values[j + 1] = temp;

}

}

}

}

int calculateMedian(vector<int>& values) {

bubbleSort(values);

int n = values.size();

int median;

if (n % 2 == 0) {

median = (values[n / 2 - 1] + values[n / 2]) / 2;

} else {

median = values[n / 2];

}

return median;

}

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

int maxFrequency = 0;

int modeValue = 0;

for (int i = 0; i < values.size(); i++) {

int frequency = 0;

for (int j = 0; j < values.size(); j++) {

if (values[i] == values[j]) {

frequency++;

}

}

if (frequency > maxFrequency) {

maxFrequency = frequency;

modeValue = values[i];

}

}

return modeValue;

}

void printEqualGrades(const vector<string>& names, const vector<int>& grades, int mode) {

cout << "Students with Grades Equal to Mode Value: ";

for (int i = 0; i < names.size(); i++) {

if (grades[i] == mode) {

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

}

}

cout << endl;

}

int main() {

vector<string> studentNames;

vector<int> studentGrades;

int num, mean, medianValue, modeValue;

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

cin >> num;

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

string name;

int grade;

cout << "Name of student: ";

cin >> name;

studentNames.push\_back(name);

cout << "Enter the grade percentage: ";

cin >> grade;

studentGrades.push\_back(grade);

}

int sum = 0;

for (int i = 0; i < studentGrades.size(); i++) {

sum += studentGrades[i];

}

mean = sum / studentGrades.size();

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

medianValue = calculateMedian(studentGrades);

cout << "Median: " << medianValue << endl;

modeValue = calculateMode(studentGrades);

cout << "Mode: " << modeValue << endl;

printEqualGrades(studentNames, studentGrades, modeValue);

return 0;

}

