EUROPEAN UNIVERSITY OF LEFKE

Faculty of Engineering

# Department of Computer Engineering



# COMP218

OBJECT-ORIENTED PROGRAMMING

## Lab Work No. 4

Prepared by Seward Richard Mupereri (20140175)

Submitted to Dr. Ferhun Yorgancıoğlu

### Task (1)

#include<iostream>

#include<iomanip>

#include<vector>

#include<algorithm>

#include<cmath>

using namespace std;

int main()

{

    int value = 0;

    double sum = 0.0;

    int n = 0;

    vector<int> data;

    cout << "===================================" << endl;

    cout << "       STATISTICS CALCULATOR       " << endl;

    cout << "===================================" << endl;

    cout << "         ENTER Q TO EXIT         " << endl;

    cout << "-----------------------------------" << endl;

    cout << setw(20) << "ENTER VALUE: ";

    while ( cin >> value )

    {

        cout << setw(20) << "ENTER VALUE: ";

        data.push\_back( value );

    }

    for (size\_t i = 0; i < data.size(); i++)

    {

        sum += data[i];

    }

    cout << "-----------------------------------" << endl;

    cout << setw(14) << "UNSORTED DATA:";

    for (size\_t i = 0; i < data.size(); i++)

    {

        cout << setw(3) << data[i];

    }

    cout << endl;

    int i, hold, j;

    for ( i = 1; i < data.size(); i++)

    {

        hold = data[i];

        j = i - 1;

        while (j >= 0 && data[j] > hold)

        {

            data[j + 1] = data[j];

            j = j - 1;

        }

        data[j + 1] = hold;

    }

    cout << setw(14) << "SORTED DATA:";

    for (size\_t i = 0; i < data.size(); i++)

    {

        cout << setw(3) << data[i];

    }

    cout << endl;

    cout << "-----------------------------------" << endl;

    cout << setw(20) << "MEAN = " << setw(3) <<  sum / data.size() << endl;

    double mid;

    if ( data.size() % 2 != 0 )

    {

        mid = (double) data[ data.size() / 2];

    }

    else

    {

        mid = (double)( data[ ( data.size() - 1) / 2 ] + data[ data.size() / 2] ) / 2.0;

    }

    cout << setw(20) << "MEDIAN = " << setw(3) << mid << endl;

    cout << setw(20) << "MINIMUM = " << setw(3) << data.front() << endl;

    cout << setw(20) << "MAXIMUM = "  << setw(3) << data.back() << endl;

    double stdDvn = 0.0;

    double stdDvnAns;

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

    {

        stdDvn += pow(data[i] - ( sum / data.size() ), 2);

    }

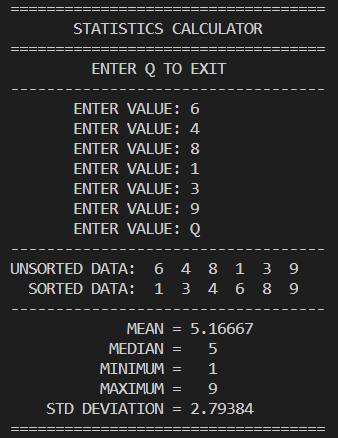
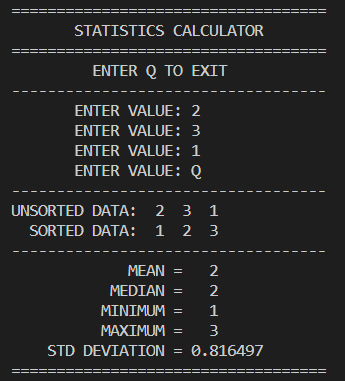
    stdDvnAns = sqrt( stdDvn / data.size() );

    cout << setw(20) << "STD DEVIATION = "  << setw(3) << stdDvnAns << endl;

    cout << "===================================" << endl;

    return 0;

}

### Task (2)

#include<iostream>

#include<iomanip>

#include<vector>

#include<algorithm>

#include<cmath>

using namespace std;

void insertionSort(vector<int> &data)

{

    int i, hold, j;

    for ( i = 1; i < data.size(); i++)

    {

        hold = data[i];

        j = i - 1;

        while (j >= 0 && data[j] > hold)

        {

            data[j + 1] = data[j];

            j = j - 1;

        }

        data[j + 1] = hold;

    }

}

double mean (vector<int> &data, int sum)

{

    return (sum / data.size() );

}

double median(vector<int> &data)

{

    double mid;

    if ( data.size() % 2 != 0 )

    {

        return (double) data[ data.size() / 2];

    }

    else

    {

        return (double)( data[ ( data.size() - 1) / 2 ] + data[ data.size() / 2] ) / 2.0;

    }

}

int minimum(vector<int> &data)

{

    return data.front();

}

int maximum(vector<int> &data)

{

    return data.back();

}

double standardDeviation(vector<int> &data, int sum)

{

    double stdDvn = 0.0;

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

    {

        stdDvn += pow(data[i] - mean(data, sum), 2);

    }

    return sqrt(stdDvn / data.size() );

}

int main()

{

    int value = 0;

    double sum = 0.0;

    int n = 0;

    vector<int> data;

    cout << "===================================" << endl;

    cout << "       STATISTICS CALCULATOR       " << endl;

    cout << "===================================" << endl;

    cout << "         ENTER Q TO EXIT         " << endl;

    cout << "-----------------------------------" << endl;

    cout << setw(20) << "ENTER VALUE: ";

    while (  cin >> value )

    {

        cout << setw(20) << "ENTER VALUE: ";

        data.push\_back( value );

    }

    for (size\_t i = 0; i < data.size(); i++)

    {

        sum += data[i];

    }

    cout << "-----------------------------------" << endl;

    cout << setw(14) << "UNSORTED DATA:";

    for (size\_t i = 0; i < data.size(); i++)

    {

        cout << setw(3) << data[i];

    }

    cout << endl;

    insertionSort(data);

    cout << setw(14) << "SORTED DATA:";

    for (size\_t i = 0; i < data.size(); i++)

    {

        cout << setw(3) << data[i];

    }

    cout << endl;

    cout << "-----------------------------------" << endl;

    cout << setw(20) << "MEAN = " << setw(3) <<  mean(data, sum) << endl;

    cout << setw(20) << "MEDIAN = " << setw(3) << median(data) << endl;

    cout << setw(20) << "MINIMUM = " << setw(3) << minimum(data) << endl;

    cout << setw(20) << "MAXIMUM = "  << setw(3) << maximum(data) << endl;

    cout << setw(20) << "STD DEVIATION = "  << setw(3) << standardDeviation(data, sum) << endl;

    cout << "===================================" << endl;

    return 0;

}

