EUROPEAN UNIVERSITY OF LEFKE
Faculty of Engineering
Department of Computer Engineering

COMP218
OBJECT-ORIENTED PROGRAMMING

LAB WORK NO. 4

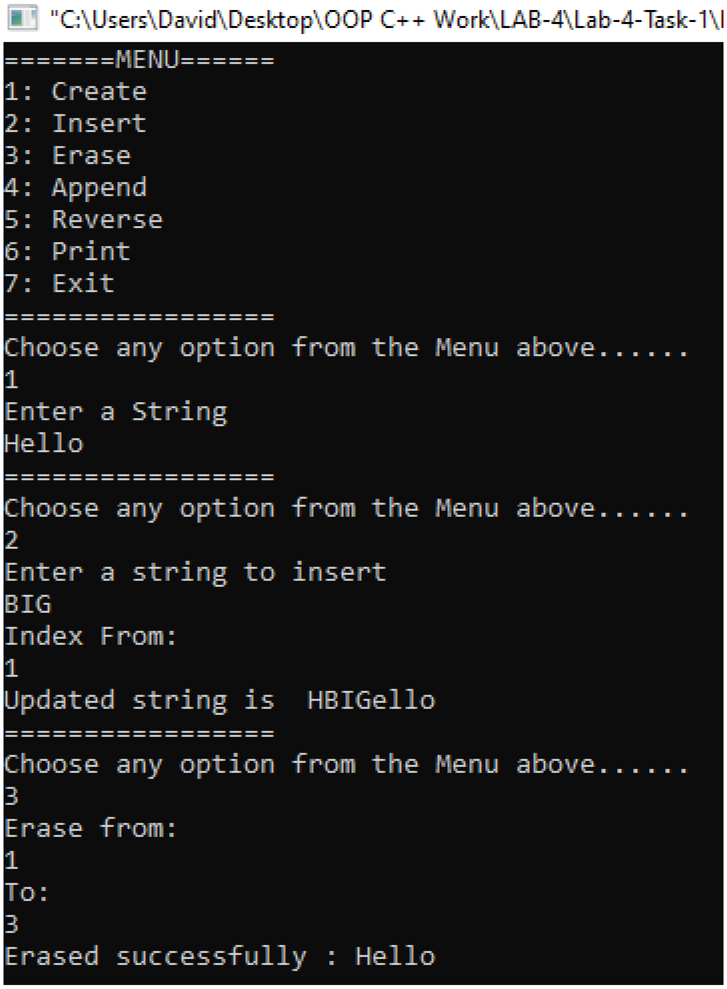
Prepared by **David O. Ladipo** (174574)
Submitted to Dr. Ferhun Yorgancıoğlu

Task-1: Write a C++ program that implements a menu-driven approach to facilitate several string handling functions. The list of options shall be to create, insert, erase, append, reverse and print. Each option shall be implemented as a user-defined C++ function trying the hide certain details from the user. For instance, when the user chooses to erase a certain part of the string, the program may print the string first for reminding and then ask the user to specify from which index to which index to erase. Similar approaches can be taken to try and experience the usage of necessary member functions of the string class.

```
#include <iostream>
#include <string>
#include <bits/stdc++.h>
using namespace std;

//Created a class called userString
class userString{
private:
    string str;
public:
    //Function prototypes
    void createString();
    void insertString();
    void eraseString();
    void appendString();
    void reverseString();
    string printString();
};
//create string function to class userString
void userString::createString()
{
    cin.ignore(); //clearing one or more character from the input buffer
    cout<< "Enter a String" <<endl; // output
    getline(cin,str);//takes input from user, stores it in str
}
//insert string function to class userString
void userString::insertString()
{
    cin.ignore();//clearing one or more buffer from the input buffer
    string newString;
    int index;
    cout<< "Enter a string to insert" << endl;
    cin >> newString; //receives string from user and stores in newString
    //an index variable initialized to store index from the user
    //where the string should be inserted
    cout<< "Index From: "<<endl;
    cin>> index;
    str.insert(index, newString); //c++ insert function is called on str
    //and inserts the newstring at the specified index
    cout<< "Updated string is " << " "<< str<< endl; // outputs the updated
string

}
//erase string function to class userString
void userString::eraseString()
{
    int start, finish;
    cout << "Erase from: " << endl;
    cin>> start;
    cout<< "To: " << endl;
    cin >> finish;
    str.erase(start,finish); // c++ erase function called on str..
    //This will erase strings from the specified start to finish index
    cout << "Erased successfully : " <<str <<endl;
}
//append string function to class userString
void userString::appendString()
{
    string strAppend;
    cout << "Enter the string you wish to append" << endl;
    cin >> strAppend;
    str.append(strAppend); //c++ string append function is called.. This appends
    //the strAppend string from the user at the end of the initial string
    cout << "Appended string is: "<< str <<endl;
}
//reverse string function to class userString
void userString::reverseString()
{
    reverse(str.begin(), str.end()); // this c++ string reverse function
    //reverses the string from end to front
    cout << "Reverse string is: " << str <<endl; //outputs the reversed string
}
string userString::printString()
{
    return str; //simply returns the string
}
//function for the menu
void menu(){
  cout<<"=======MENU======"<<endl;
  cout<<"1: Create"<<endl;
  cout<<"2: Insert"<<endl;
  cout<<"3: Erase"<<endl;
  cout<<"4: Append"<<endl;
  cout<<"5: Reverse"<<endl;
  cout<<"6: Print"<<endl;
  cout<<"7: Exit"<<endl;
  cout<<"================="<<endl;
}
int main()
{
    //created an object of class userString
    userString s;
    int option; //holds the value that will be used for the switch case
    menu();
    while(1){

    cout<< "Choose any option from the Menu above......" << endl;
    cin>> option;
    //switch case function is initialized and the object will be called on the
    //various functions of the class
    switch(option)
    {
        case 1:
        s.createString();
        cout<<"================="<<endl;
        break;

        case 2:
        s.insertString();
        cout<<"================="<<endl;
        break;

        case 3:

          s.eraseString();
          cout<<"================="<<endl;
            break;
        case 4:
            s.appendString();
            cout<<"================="<<endl;
            break;
        case 5:
            s.reverseString();
            cout<<"================="<<endl;
```

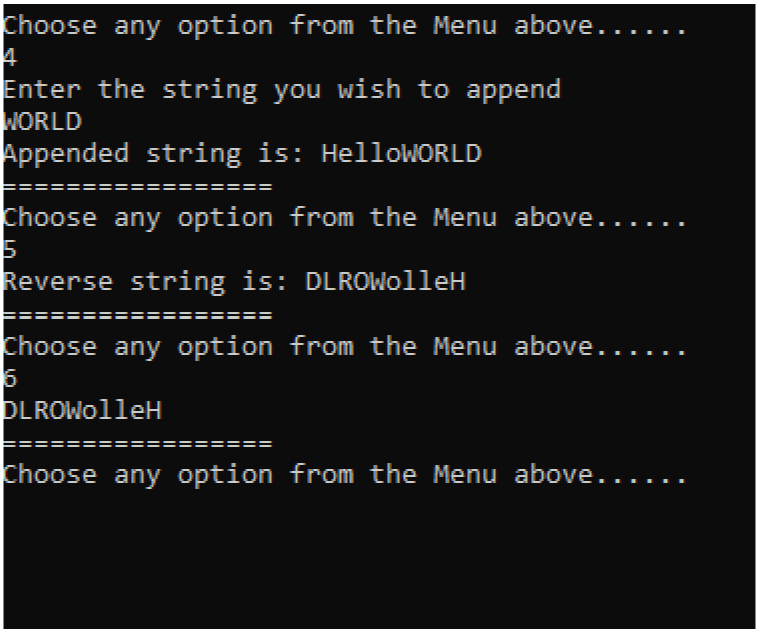
```
      break;
    case 6:
     cout<<  s.printString() << endl;
     cout<<"================="<<endl;
      break;
    case 7:
      cout << "Program Finished..." <<endl;
      return 0;
      break;
    default:
      cout << "You're out of bound, Please select a valid option above.."
<< endl;
      break;

        }

  }
  return 0;
}
```

OUTPUT:



"C:\Users\David\Desktop\OOP C++ Work\LAB-4\Lab-4-Task-1\I

```
=======MENU======
1: Create
2: Insert
3: Erase
4: Append
5: Reverse
6: Print
7: Exit
=================
Choose any option from the Menu above......
1
Enter a String
Hello
=================
Choose any option from the Menu above......
2
Enter a string to insert
BIG
Index From:
1
Updated string is  HBIGello
=================
Choose any option from the Menu above......
3
Erase from:
1
To:
3
Erased successfully : Hello
```



```
Choose any option from the Menu above......
4
Enter the string you wish to append
WORLD
Appended string is: HelloWORLD
=================
Choose any option from the Menu above......
5
Reverse string is: DLROWolleH
=================
Choose any option from the Menu above......
6
DLROWolleH
=================
Choose any option from the Menu above......
```

Task-2: Write a C++ program that lets the user enter a list of integers for processing. The program shall implement a menu-driven approach to let the user specify integers and store them in a vector object. The inputting process can continue if the user wants to enter more. The program shall calculate the minimum, the maximum, the mean, the median, and the standard deviation. To calculate the median the program shall sort the list in increasing order as well.

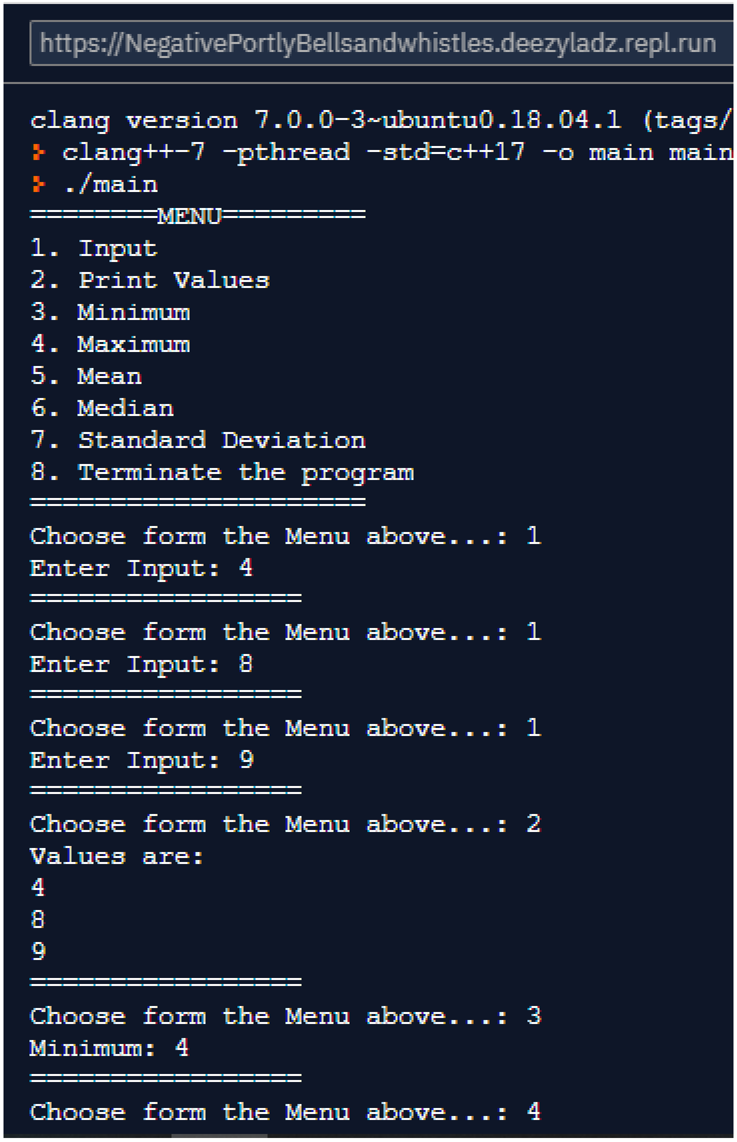
```cpp
#include <iostream>
#include <vector>
#include <cmath>
using namespace std;

//Created a class called Calculate
class Calculate{
private:
    vector<int>myVec;
    int counter;
    int average;
    int sum;

public:
    //Function prototypes
    void Insert();
    void Delete();
    int Mean();
    void Minmum();
    void Maximum();
    void standardDeviation();
    void Median();
    void PrintValues();
};
//create an insert function to class Calculate
void Calculate:: Insert()
{
    cout << "Enter Input: "; //outputs Enter input to the user
    int input;
    cin >> input; //takes the user input and saves it in input variable
    myVec.push_back(input); //push_back vector function called on myVec
    //to enable elements to be inserted in the vector
}
//create a mean function to class Calculate
//calculates the mean of the elements in the vector
int Calculate::Mean()
```

```
{
    sum = counter= average = 0;
    //for loop to iterate through all the elements and sum them up,
    //at the same time keeping count with the counter variable
    for (size_t i=0; i < myVec.size(); i++)
    {
        sum+=myVec[i];
        counter++;
    }
    average = sum/counter;  //the sum is divided by the
    //number of counts and assigned to average
     return average;
}
//create a minimum function to class Calculate
//checks for the minimum value in the vector
void Calculate::Minmum()
{
    int min;
    min = myVec[0];
    for (size_t i=0; i < myVec.size(); i++)
    {
        if (min > myVec[i]) //compares first element withe second and so  on...
        min = myVec[i];
    }
    cout << "Minimum: " << min <<endl;
}
//create a Maximum function to class Calculate
//checks for the maximum value in the vector
void Calculate::Maximum()
{
    int max;
    max = myVec[0];
    for (size_t i=0; i < myVec.size(); i++)//compares first element withe second
and so  on...
        {
        if (max < myVec[i])
            max = myVec[i];
    }

    cout << "Maximum is: " << max << endl;
}
//create a standard deviation function to class Calculate
//standard deviation is a quantity expressing by how much the
//members of a group differ from the mean value for the group.
void Calculate::standardDeviation()
```

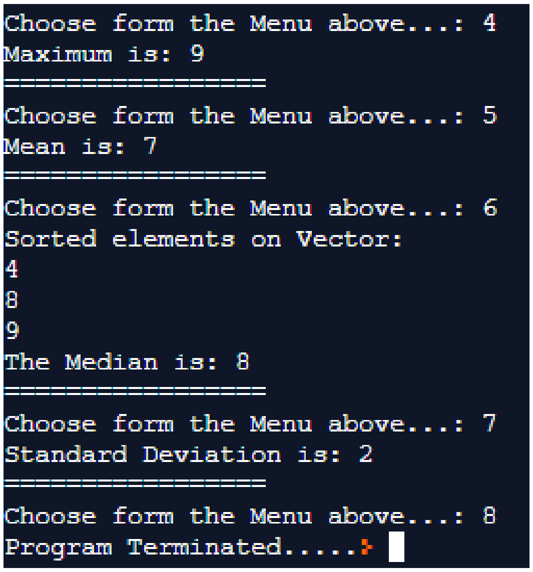
```
{
   int avrg = Mean(),n,c=0,ans;
      for(unsigned int i = 0;i < myVec.size();i++){
        n += pow((myVec[i] - avrg),2); // calculates the each element in the
        //vector minus the average to the power of 2 and adds it to a variable n
        c++; // keeps count
      }
      ans = sqrt(n/c); //finally finds to square root of n/c and assigns to a
variable ans
        cout << "Standard Deviation is: " << ans << endl;
}
//create a median function to class Calculate
void Calculate::Median()
{
    //insertion sort to sort the integers in the vector
    int i, key, j,n=0;
    for (i = 1; i < myVec.size(); i++) {
      key = myVec[i];
      j = i - 1;
      while (j >= 0 && myVec[j] > key) {
        myVec[j + 1] = myVec[j];
        j = j - 1;
      }
      myVec[j + 1] = key;
    }
    //prints out the sorted elements in the vector
    cout << "Sorted elements on Vector: " << endl;
    for(i = 0; i < myVec.size(); i++){
      cout<<myVec[i]<<" " <<endl;
      n++;
    }
    //calculating the median - median is the middle number in a set of given
numbers
    //if the set of numbers is odd - the median is the middle one
    //if its even - then the median is the sum of the two numbers in the middle
    int middle;
    double median;

    middle = myVec.size()/2;
    //checks if the vector is odd or even
    if (myVec.size()%2==1)
    {
        median = myVec[middle]; //if odd, then the median is the middle element

        cout << "The Median is: " << median << endl;
    }
    else
    {
        //if even, then both numbers in the middle divided by 2
        median = (myVec[middle]+ myVec[middle - 1])/2.0;
        cout << "The Median is: " << median << endl;
    }

  }
  //create a print function to class Calculate
  //this function simply prints out the elements in the vector
void Calculate::PrintValues()
{
    cout << "Values are: " << endl;
    for (size_t i=0; i < myVec.size(); i++)
        cout <<  myVec[i] << " " << endl;
}

//Menu function
void menu(){
  cout<<"========MENU========="<<endl;
  cout<<"1. Input"<<endl;
  cout<<"2. Print Values"<<endl;
  cout<<"3. Minimum"<<endl;
  cout<<"4. Maximum"<<endl;
  cout<<"5. Mean"<<endl;
  cout<<"6. Median"<<endl;
  cout<<"7. Standard Deviation"<<endl;
  cout<<"8. Terminate the program"<<endl;
  cout<<"====================="<<endl;
}

int main()
{
    Calculate x; //created an object x of class calculate
    int option;//holds the value that will be used for the switch case
    menu(); // menu function is called

    while(1){
        cout << "Choose form the Menu above...: ";
        cin >> option;
        //switch case function is initialized and the object will be called on
the
        //various functions of the class
        switch(option)
```

```
                    {
        case 1:
        x.Insert();
        cout<<"================="<<endl;
        break;

        case 2:
        x.PrintValues();
        cout<<"================="<<endl;
        break;

        case 3:
          x.Minmum();
          cout<<"================="<<endl;
          break;
        case 4:
           x.Maximum();
           cout<<"================="<<endl;
           break;
        case 5:
         cout << "Mean is: "<<  x.Mean() << endl;
            cout<<"================="<<endl;
            break;
        case 6:
          x.Median();
          cout<<"================="<<endl;
            break;
        case 7:
            x.standardDeviation();
            cout<<"================="<<endl;
            break;
        case 8:
            cout<<"Program Terminated.....";
            return 0;
            break;
        default:
            cout << "Invalid Option!! Please Select from the option above" <<
endl;
            break;

                }
    }
    return 0;
}
```

OUTPUT:



```
https://NegativePortlyBellsandwhistles.deezyladz.repl.run

clang version 7.0.0-3~ubuntu0.18.04.1 (tags/
› clang++-7 -pthread -std=c++17 -o main main
› ./main
========MENU=========
1. Input
2. Print Values
3. Minimum
4. Maximum
5. Mean
6. Median
7. Standard Deviation
8. Terminate the program
=====================
Choose form the Menu above...: 1
Enter Input: 4
=================
Choose form the Menu above...: 1
Enter Input: 8
=================
Choose form the Menu above...: 1
Enter Input: 9
=================
Choose form the Menu above...: 2
Values are:
4
8
9
=================
Choose form the Menu above...: 3
Minimum: 4
=================
Choose form the Menu above...: 4
```



```
Choose form the Menu above...: 4
Maximum is: 9
=================
Choose form the Menu above...: 5
Mean is: 7
=================
Choose form the Menu above...: 6
Sorted elements on Vector:
4
8
9
The Median is: 8
=================
Choose form the Menu above...: 7
Standard Deviation is: 2
=================
Choose form the Menu above...: 8
Program Terminated.....›
```