Week 7

Exercise 1

#include "stdafx.h"

#include<iostream>

#include<string>

#include<vector>

using namespace std;

template <class T>

bool palindrome(vector<T> a)

{

int begin, end, check;

for (begin = 0, end = a.size()- 1; begin < end; begin++, end--)

{

if (a[begin] != a[end])

{

check = 0;

break;

}

else

check = 1;

}

if (check == 0)

return false;

else

return true;

}

int main()

{

const int size = 9;

int arr[9] = { 1,2,3,4,5,6,7,8,9 };

vector <int> v1(arr, arr + size);

char a[9] = { 'a','b','c','d','e','d','c','b','a' };

vector <char> v2(a, a + size);

cout << "Exercise 1" << endl;

cout<<"v1 is "<<(palindrome(v1)?"true":"false")<<endl;

cout<<"v2 is "<< (palindrome(v2) ? "true" : "false") << endl;

    return 0;

}

Exercise 2

// Week 7 Practical.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

#include<string>

#include<vector>

using namespace std;

#include"student.h"

int \_tmain(int argc, \_TCHAR\* argv[])

{

student s1;

string a,b;

float c,d;

cout<<"Please enter the first name"<<endl;

cin>>a;

cout<<"Please enter the last name"<<endl;

cin>>b;

cout<<"Please enter the year 1 mark"<<endl;

cin>>c;

cout<<"Please enter the year 2 mark"<<endl;

cin>>d;

s1.ReadData(a,b,c,d);

s1.SpliceNames(a,b);

s1.MergeMark();

s1.Average();

s1.ShowFinal();

return 0;

}

Student.h

#pragma once

#include"student.h"

class student

{

public:

student(void);

~student(void);

void ReadData(string, string, float, float);

string SpliceNames(string, string);

float MergeMark();

float Average();

void ShowFinal();

private:

string FullName;

string FirstName;

string LastName;

float Year1Mark;

float Year2Mark;

float Finalmark;

float AverageM;

};

Student.cpp

#include "stdafx.h"

#include<iostream>

#include<string>

using namespace std;

#include "student.h"

student::student(void)

{

}

student::~student(void)

{

}

void student::ReadData(string a, string b, float c, float d)

{

FirstName='a';

LastName='b';

Year1Mark=c;

Year2Mark=d;

}

string student::SpliceNames(string a, string b)

{

FullName+=a;

FullName+=" ";

FullName+=b;

return FullName;

}

float student::MergeMark()

{

Finalmark=Year1Mark+Year2Mark;

return Finalmark;

}

float student::Average()

{

AverageM=Finalmark/2;

return AverageM;

}

void student::ShowFinal()

{

cout<<"The average mark of "<<FullName<<" is "<<AverageM;

}