Task 1:

//MAIN file (source.cpp)

#include<iostream>

#include<string>

#include"interface1.h"

using namespace std;

template <typename t>

void add(t x, t y)

{

cout << x + y<<endl;

}

template <typename t>

void mul(t a, t b)

{

cout << a \* b<<endl;

}

int main()

{

add<int>(5.76, 5);

add<float>(5.76, 5.651f);

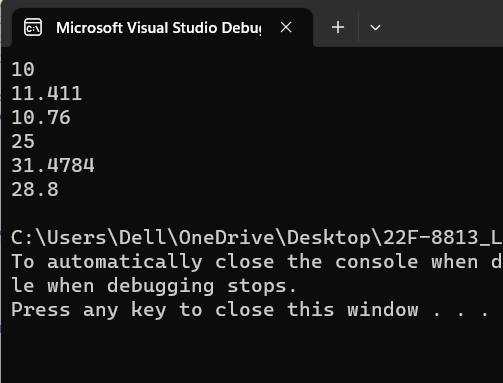
add<double>(5.76, 5);

mul<int>(5.76, 5);

mul<float>(5.76f, 5.465f);

mul<double>(5.76, 5);

}



Task 2:

//MAIN file (source.cpp)

#include<iostream>

#include<string>

#include"interface2.h"

using namespace std;

template <typename t>

void avg(t a[5], int size)

{

t sum=0;

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

{

sum += a[i];

}

sum = sum / size;

cout<< sum<<endl;

}

int main()

{

float arrfloat[5] = { 2.5461,5.345,6.153,7.65,8.54 };

int arrint[5] = { 1,2,3,4,5 };

char arrchar[5] = { 'a','b','c','d','e' };

double arrdouble[5] = { 2.5461,5.345,6.153,7.65,8.54 };

int size = sizeof(arrint) / sizeof(arrint[0]);

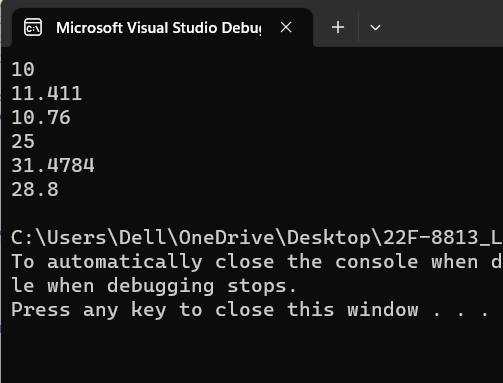
avg(arrint, size);

avg(arrfloat, size);

avg(arrchar, size);

avg(arrdouble, size);

}



Task 3:

//MAIN file (source.cpp)

#include<iostream>

#include<string>

#include"interface3.h"

using namespace std;

template <typename t>

void swap(t a, t b)

{

t temp;

temp = a;

a = b;

b = temp;

cout << "\nfirst is : " << a << "\nsecond is : " << b;

}

int main()

{

int i = 2, j = 4;

float c = 5.135, d = 354.433;

double e = 5.135, f = 354.433;

char g = 'a', h = 'b';

string str1 = "hanan", str2 = "abdul";

cout << "before swap : \n";

cout << &i << "\n" << &j << "\n" << &c << "\n" << &d << "\n" << &e << "\n" << &f << "\n" << &g << "\n" << &h << "\n" << &str1 << "\n" << &str2;

cout << "\n after swap : ";

swap(&i, &j);

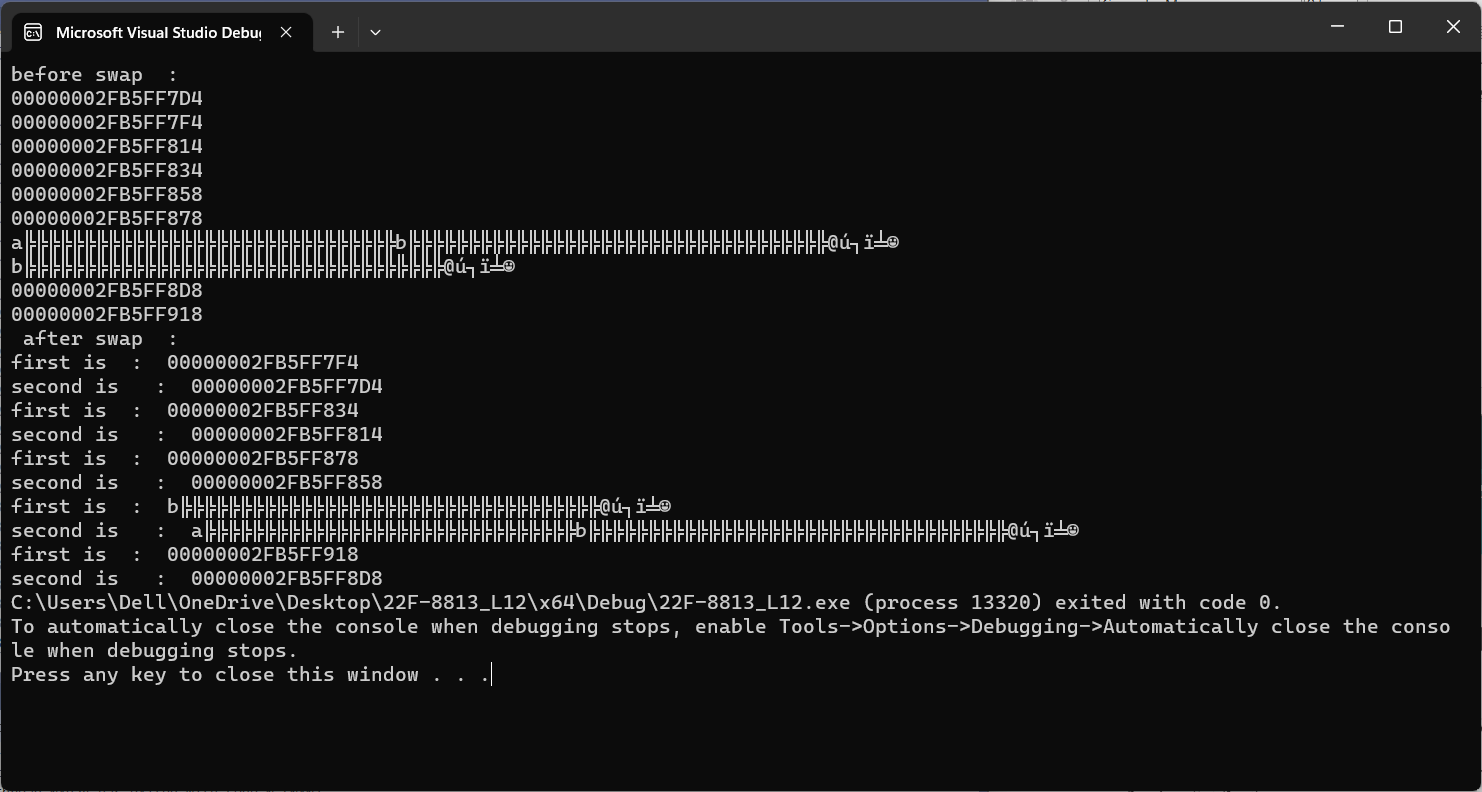
swap(&c,&d);

swap(&e,&f);

swap(&g,&h);

swap(&str1,&str2);

}



Task 4:

// Class.h file (class header file)

#pragma once

//#ifndef CLASS\_H

#include<string>

#include<iostream>

using namespace std;

//#define CLASS\_H

template<class T>

class triangl

{

public:

T length;

T width;

triangl() :length(0), width(0) {}

void set(T a, T b)

{

length = a;

width = b;

}

T area()

{

T temp;

temp = length \* width;

return (0.5) \* temp;

}

T perimeter()

{

return length + width;

}

};

//MAIN file (source.cpp)

#include<iostream>

#include<string>

#include"Header4.h"

using namespace std;

int main()

{

triangl<int> iobj;

triangl<float> fobj;

triangl<double> dobj;

iobj.set(3, 4);

cout << "\n int area is : " << iobj.area();

cout << "\n int perimeter is : " << iobj.perimeter();

fobj.set(3.531, 531.54);

cout << "\n float area is : " << fobj.area();

cout << "\n float perimeter is : " << fobj.perimeter();

dobj.set(31.353, 315.31);

cout << "\n double area is : " << dobj.area();

cout << "\n double perimeter is : " << dobj.perimeter();

iobj.set(3, 45.31351f);

cout << "\n int float area is : " << iobj.area();

cout << "\n int float perimeter is : " << iobj.perimeter();

dobj.set(31.353f, 315.31);

cout << "\n float double area is : " << dobj.area();

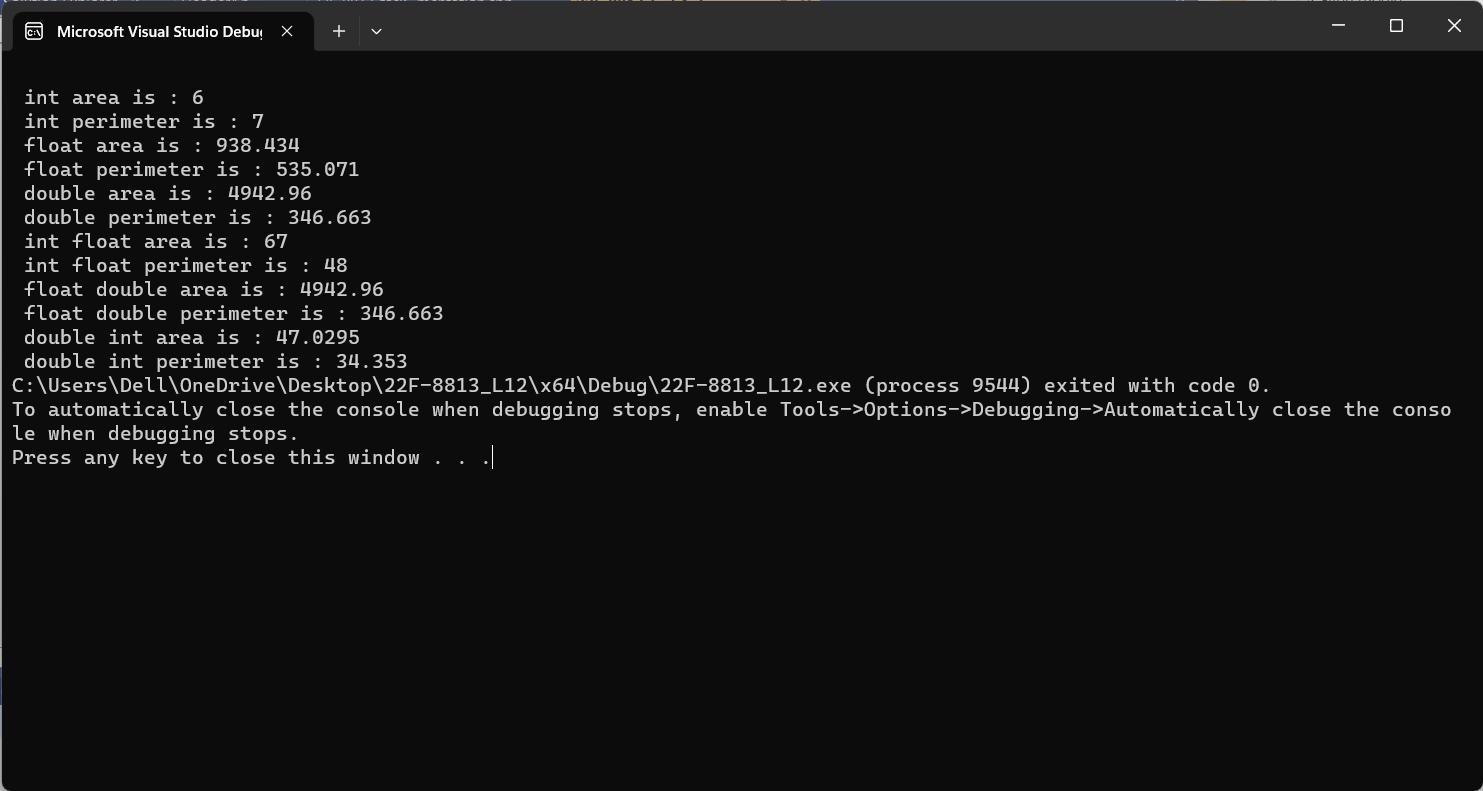
cout << "\n float double perimeter is : " << dobj.perimeter();

dobj.set(31.353, 3);

cout << "\n double int area is : " << dobj.area();

cout << "\n double int perimeter is : " << dobj.perimeter();

}



Task 5:

//MAIN file (source.cpp)

#include<iostream>

#include<string>

#include"interface5.h"

using namespace std;

int main()

{

calculator <int> iobj;

calculator <float> fobj;

iobj.set(2, 5);

cout << "for integers : \n";

iobj.add();

iobj.mul();

iobj.sub();

iobj.div();

iobj.sq();

fobj.set(221.135, 513.531);

cout << "for floats : \n";

fobj.add();

fobj.mul();

fobj.sub();

fobj.div();

fobj.sq();

}

// Class.h file (class header file)

#pragma once

//#ifndef CLASS\_H

#include<string>

#include<iostream>

using namespace std;

//#define CLASS\_H

template<typename T>

class calculator

{

T num1;

T num2;

public:

void set(T a, T b)

{

num1 = a;

num2 = b;

}

void add();

void mul();

void sub();

void div();

void sq();

};

template<typename T>

void calculator<T>::add()

{

cout << num1 + num2<<endl;

}

template<typename T>

void calculator<T>::mul()

{

cout << num1 \* num2 << endl;

}

template<typename T>

void calculator<T>::sq()

{

cout << sqrt(num1)<<endl;

cout << sqrt(num2) << endl;

}

template<typename T>

void calculator<T>::div()

{

cout << num1 / num2 << endl;

}

template<typename T>

void calculator<T>::sub()

{

cout << num1 - num2 << endl;

}

