**M. Ali. Arslan**

**19F-0348**

**Last Lab**

**Program # 1:**

#include<iostream>

using namespace std;

template <typename T , typename X>

T add(T y, X x)

{

return x + y;

}

template <typename U , typename C>

U multiply(U x, C t)

{

return (x \* t);

}

int main()

{

cout << " Add(int,int) = " << add<int, int>(10, 5) << endl;

cout << " Add(float,float) = " << add<float, float>(9.68, 12.90) << endl;

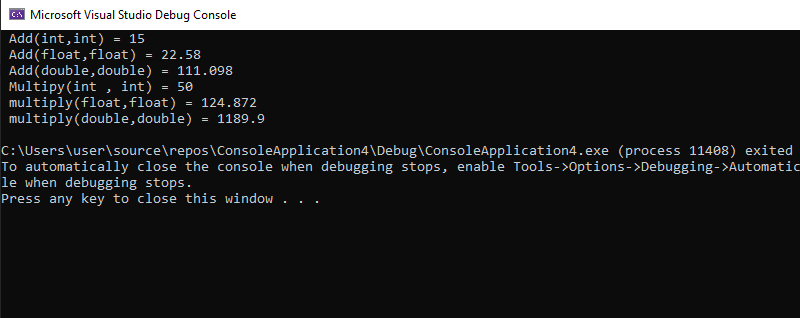
cout << " Add(double,double) = " << add<double, double>(99.09, 12.0083) << endl;

cout << " Multipy(int , int) = " << multiply<int, int>(10, 5) << endl;

cout << " multiply(float,float) = " << multiply<float, float>(9.68, 12.90) << endl;

cout << " multiply(double,double) = " << multiply<double, double>(99.09, 12.0083) << endl;

}



**Program # 2:**

#include<iostream>

using namespace std;

template <typename T, typename U>

class Triangle

{

private:

T length;

U width;

public:

Triangle()

{

length = 0;

width = 0;

}

void set\_data(T l, U w)

{

length = l;

width = w;

}

T get\_area()

{

return (0.5 \* (length \* width));

}

T get\_perimeter()

{

return (0.5 \* (length + width));

}

};

int main()

{

Triangle <int, int>obj1;

Triangle <float, float>obj2;

Triangle <float, double>obj3;

Triangle<double, int>obj4;

Triangle <float, double>obj5;

Triangle <double, double>obj6;

obj1.set\_data(10, 10);

cout << " Area(int,int) = " << obj1.get\_area() << endl;

obj2.set\_data(0.99, 0.99);

cout << " Area (float,float) = " << obj2.get\_area() << endl;

obj3.set\_data(3.56f, 4.5);

cout << " Area (float,double) = " << obj3.get\_area() << endl;

obj4.set\_data(4.5, 9);

cout << " Area(double,int) = " << obj4.get\_area() << endl;

obj5.set\_data(3.56f, 4.5);

cout << " Area (float,double) = " << obj5.get\_area() << endl;

obj6.set\_data(99.99, 89.89);

cout << " Area (double,double) = " << obj6.get\_area() << endl;

cout << " Perimeter(int,int) = " << obj1.get\_perimeter() << endl;

cout << " Permieter(float,float) = " << obj2.get\_perimeter() << endl;

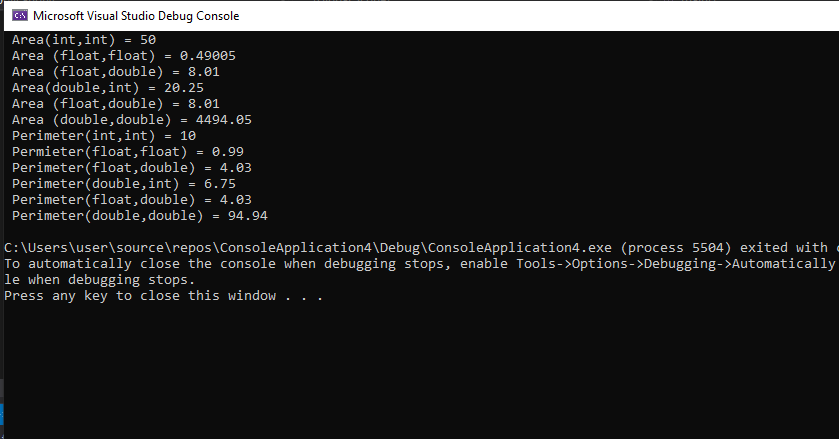
cout << " Perimeter(float,double) = " << obj3.get\_perimeter() << endl;

cout << " Perimeter(double,int) = " << obj4.get\_perimeter() << endl;

cout << " Perimeter(float,double) = " << obj5.get\_perimeter() << endl;

cout << " Perimeter(double,double) = " << obj6.get\_perimeter() << endl;

}



**Program # 3:**

// Part(A)

#include<iostream>

using namespace std;

int checK\_values(int deno)

{

if (deno == 0)

{

throw "Error";

}

else

{

return deno;

}

}

int main()

{

int number;

int denominator;

int result = 0;

cout << " Enter the Number :- " << endl;

cin >> number;

cout << " Enter the Denominator :- " << endl;

cin >> denominator;

try

{

if (checK\_values(denominator))

{

result = number / denominator;

cout << " The Quotient is :- " << result << endl;

}

}

// catches any type of exception.

catch (...)

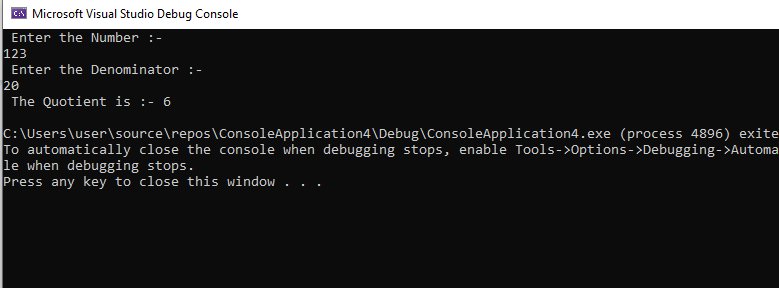
{

cout << " Exception Occoured " << endl;

}

return 0;

}



// Part(B)

#include<iostream>

#include<string>

using namespace std;

int main()

{

int x = -5;

char a = 'F';

string str = "GTH";

float b = -8.90;

try

{

if (x < 0)

{

throw x;

}

if (!a == 'G')

{

throw a;

}

if (str == "GTH")

{

throw str;

}

if (b < 0)

{

throw b;

}

}

catch (int x)

{

cout << " Exception Occoured " << endl;

}

catch (char x)

{

cout << " Invalid Character Choosed " << endl;

}

catch (string x)

{

cout << " Invalid input , Exception Occoured " << endl;

}

catch (float x)

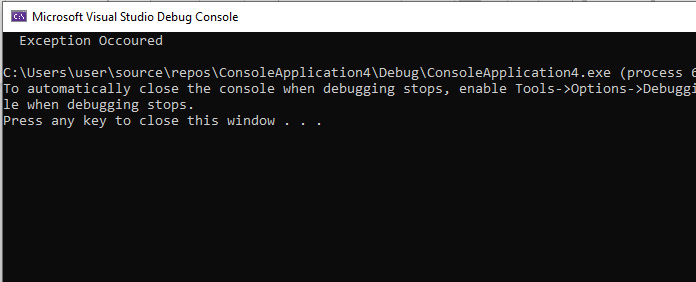
{

cout << " value cannot be Smaller than zero " << endl;

}

return 0;

}



**Program # 4:**

#include<iostream>

using namespace std;

void function\_menu();

void add\_fractions(int a, int b, int c, int d);

void multiply\_fractions(int a, int b, int c, int d);

void divide\_fractions(int a, int b, int c, int d);

void subtract\_fractions(int a, int b, int c, int d);

int nr; // numirator result

int dr; // denominator result

int main()

{

int choice;

int numirator1, numirator2, denominator1, denominator2;

cout << endl;

function\_menu();

try

{

cout << " Enter the Values of The First Fraction Fraction :- " << endl;

cin >> numirator1;

cin >> denominator1;

if (denominator1 == 0)

{

throw denominator1;

}

cout << " Enter the values of the Second Fraction " << endl;

cin >> numirator2;

cin >> denominator2;

if (denominator2 == 0)

{

throw denominator2;

}

}

catch (...)

{

cout << " EXCEPTION OCCOURED ! " << endl;

cout << " Enter the Values of F1 Again " << endl;

cin >> numirator1;

cin >> denominator1;

if (denominator1 == 0)

{

cout << " Invalid Input Try once more " << endl;

while (denominator1 == 0)

{

cout << " Zero Is An Invalid Number For This fraction " << endl;

cin >> denominator1;

}

}

cout << " Enter the Values of F2 Again " << endl;

cin >> numirator2;

cin >> denominator2;

if (denominator2 == 0)

{

cout << " Invalid Input Try once more " << endl;

while (denominator2 == 0)

{

cout << " Zero Is An Invalid Number For This fraction " << endl;

cin >> denominator2;

}

}

}

cout << " The Fraction entered is F1 = : " << numirator1 << "/" << denominator1 << endl;

cout << " The Fraction entered is F2 = : " << numirator2 << "/" << denominator2 << endl;

cout << endl;

cout << " Enter which Task you wan the Program to Perform Or Press -1 to Quit" << endl;

while (1)

{

cout << endl;

function\_menu();

cout << endl;

cin >> choice;

if (choice == -1)

{

break;

}

switch (choice)

{

case 1:

cout << " The result is " << endl;

add\_fractions(numirator1, denominator1, numirator2, denominator2);

cout << nr << "/" << dr << endl;

break;

case 2:

cout << " The result is " << endl;

multiply\_fractions(numirator1, denominator1, numirator2, denominator2);

cout << nr << "/" << dr << endl;

break;

case 3:

cout << " The result is " << endl;

divide\_fractions(numirator1, denominator1, numirator2, denominator2);

cout << nr << "/" << dr << endl;

break;

case 4:

cout << " The result is " << endl;

subtract\_fractions(numirator1, denominator1, numirator2, denominator2);

cout << nr << "/" << dr << endl;

break;

default:

break;

}

}

return 0;

}

void function\_menu()

{

cout << " This Function can do The Following Tasks " << endl;

cout << " 1) Add Two Fractions " << endl;

cout << " 2) Multiply Fractions " << endl;

cout << " 3) Divide Fractions " << endl;

cout << " 4) Subtract Fractions " << endl;

}

void add\_fractions(int a, int b, int c, int d)

{

cout << " Performing Addition " << endl;

nr = a + c;

dr = b + d;

}

void multiply\_fractions(int a, int b, int c, int d)

{

cout << " Performing Multiplication " << endl;

nr = a \* c;

dr = b \* d;

}

void divide\_fractions(int a, int b, int c, int d)

{

nr = a / c;

dr = b / d;

}

void subtract\_fractions(int a, int b, int c, int d)

{

nr = a - c;

dr = b - d;

}

