**Question 1:**

#include<iostream>

using namespace std;

template <class t1>

t1 add(t1 a, t1 b) {

return a + b;

}

template <class t2>

t2 mul(t2 a, t2 b) {

return a \* b;

}

int main(){

cout << "THE SUM IS = " << add(4, 5)<<endl;

cout << "THE SUM IS = " << add(4.2, 5.5) << endl;

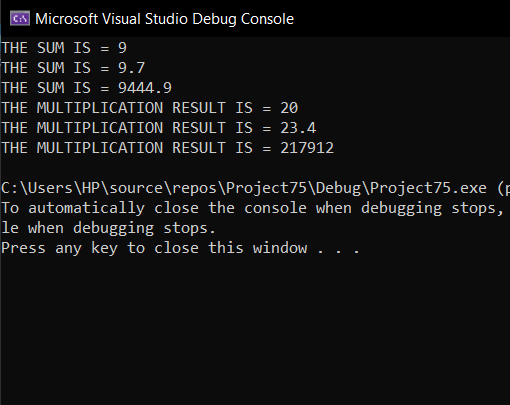
cout << "THE SUM IS = " << add(4000.5, 5444.4) << endl;

cout << "THE MULTIPLICATION RESULT IS = " << mul(4, 5)<<endl;

cout << "THE MULTIPLICATION RESULT IS = " << mul(4.5, 5.2) << endl;

cout << "THE MULTIPLICATION RESULT IS = " << mul(400.5, 544.1) << endl;

}



**Question 2:**

#include<iostream>

using namespace std;

template <class t,class s>

class triangle {

private:

public:

t area(t a, t b) {

return 0.5 \* (a \* b);

}

t paramter(t a, t b)

{

return (a \* b);

}

};

int main(){

triangle <int,int> arham;

cout << "BOTH PARAMETERS ARE INT " << endl;

cout <<"AREA ANSWER IS ="<< arham.area(4, 5)<<endl;

cout << "PARAMTER ANSWER IS =" << arham.paramter(4, 5) << endl << endl;

cout << "BOTH PARAMETERS ARE FLOAT " << endl;

triangle <float,float> arham1;

cout << "AREA ANSWER IS =" << arham1.area(4.2, 3.4)<<endl;

cout << "PARAMETER ANSWER IS =" << arham1.paramter(4.2, 3.4)<<endl<<endl;

cout << "BOTH PARAMETERS ARE INT,FLOAT " << endl;

triangle <int,float> arham2;

cout << "AREA ANSWER IS =" << arham2.area(444.2, 333.4) << endl;

cout << "PARAMETER ANSWER IS =" << arham2.paramter(444.2, 333.3) << endl<<endl;

cout << "BOTH PARAMETERS ARE FLOAT,DOUBLE " << endl;

triangle <float,double> arham3;

cout << "AREA ANSWER IS =" << arham3.area(444.2, 333.4) << endl;

cout << "PARAMETER ANSWER IS =" << arham3.paramter(444.2, 333.3) << endl << endl;

cout << "BOTH PARAMETERS ARE DOUBLE,INT " << endl;

triangle <double,int> arham4;

cout << "AREA ANSWER IS =" << arham4.area(444.2, 333.4) << endl;

cout << "PARAMETER ANSWER IS =" << arham4.paramter(444.2, 333.3) << endl << endl;

cout << "BOTH PARAMETERS ARE FLOAT,DOUBLE " << endl;

triangle <float,double> arham5;

cout << "AREA ANSWER IS =" << arham5.area(444.2, 333.4) << endl;

cout << "PARAMETER ANSWER IS =" << arham5.paramter(444.2, 333.3) << endl << endl;

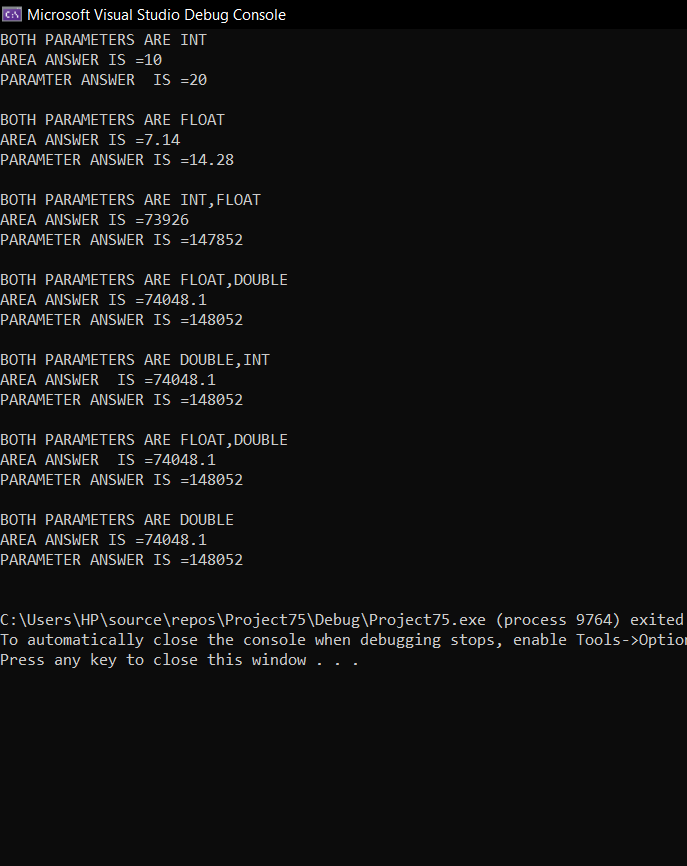
cout << "BOTH PARAMETERS ARE DOUBLE " << endl;

triangle <double, double> arham6;

cout << "AREA ANSWER IS =" << arham6.area(444.2, 333.4) << endl;

cout << "PARAMETER ANSWER IS =" << arham6.paramter(444.2, 333.3) << endl << endl;

}



**Question 3**

#include<iostream>

#include<conio.h>

#include<string>

using namespace std;

void fun() {

int a, b;

cout << "enter no 1= " << endl;

cin >> a;

cout << "enter no 2= " << endl;

cin >> b;

int k;

try

{

if (b == 0)

cout << "WHICH EXCEPTION YOU WANT TO SEE ? " << endl;

cout << "1.INTEGER" << endl << "2.CHAR" << endl << "3.FLOAT" << endl << "4.STRING" << endl;

cin >> k;

switch (k)

{

case 1:

throw 1;

break;

case 2:

throw char ('c');

break;

case 3:

throw double (1.5);

break;

case 4:

throw string ("abc");

break;

default:

cout << "invalid command" << endl;

}

}

catch (int a)

{

cout << "int type exception caught" << endl;

}

catch (char c)

{

cout << "char type exception caught" << endl;

}

catch (double f)

{

cout << "float type exception caught" << endl;

}

catch(string s)

{

cout << "string type exception caught" << endl;

}

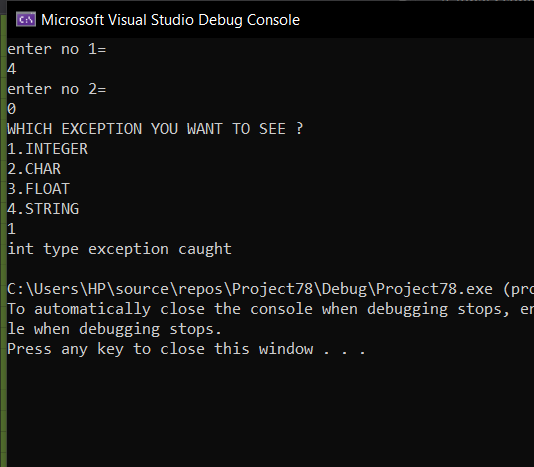
}

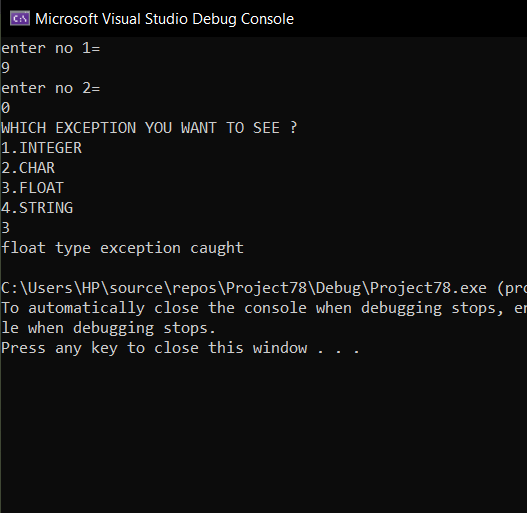
void main()

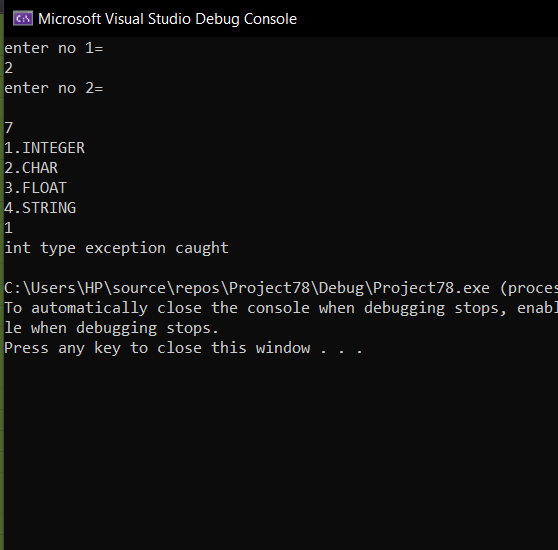
{

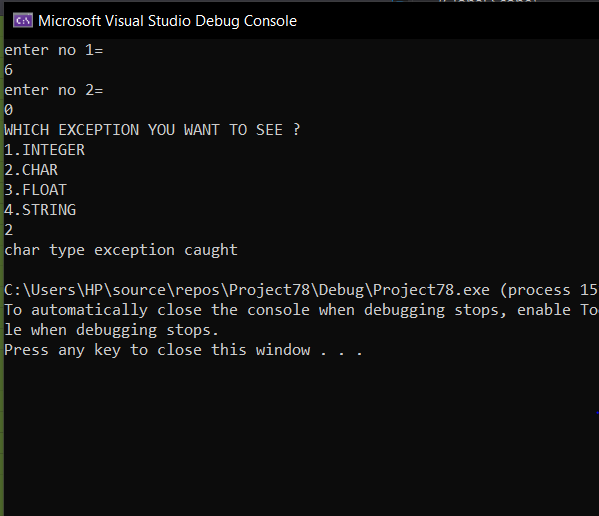
fun();

}









**Question 4:**

#include<iostream>

#include<conio.h>

#include<string>

using namespace std;

void add(int a,int b)

{

cout << a +b << endl;

}

void sub(int a,int b)

{

cout << a - b<<endl;

}

void divi(int a,int b) {

cout << a / b << endl;

}

void fun() {

int a, b,c,d;

int temp1, temp;

cout << "FRACTION 1 = " << endl;

cout << "enter denomintor = " << endl;

cin >> a;

cout << "enter numinator = " << endl;

cin >> b;

cout << "FRACTION 2 = " << endl;

cout << "enter demonintor 2= " << endl;

cin >> c;

cout << "enter numinator = " << endl;

cin >> d;

char k;

try

{

if (d == 0 || b == 0)

throw 1;

else {

temp1 = a / b;

temp = c / d;

cout << "1. ADD '+'" << endl << "2. SUB '-'" << endl << "3.DIV '/'" << endl;

cin >> k;

switch (k)

{

case '+':

add(temp1, temp);

break;

case '-':

sub(temp1, temp);

break;

case '/':

divi(temp1, temp);

break;

default:

cout << "invalid command" << endl;

}

}

}

catch (int a)

{

cout << "THE NOMONINATOR CANNOT BE ZERO" << endl;

}

}

void main()

{

fun();

}

