Task 1:

//MAIN file (source.cpp)

#include<iostream>

#include<string>

#include"interface1.h"

using namespace std;

int main()

{

try {

float num = 0;

float zero = 0;

cout << "enter the number : ";

cin >> num;

if (num / zero==0)

{

throw "sdf";

}

}

catch(string)

{

cout << "num is divided by zero \n";

}

try {

int num1 = 0;

float numf = 0;

char ch;

string str;

int temp=0;

cout << "what you want to enter 1. integer\n2.float\n3.character\n4.string\n ";

cin >> temp;

switch (temp)

{

case 1:

{

throw 1;

break;

}

case 2:

{

throw 1.1;

break;

}

case 3:

{

throw 'a';

break;

}

case 4:

{

throw str;

break;

}

default:

cout << "entered incorrect value ";

break;

}

}

catch (int)

{

cout << "int exception caught ";

}

catch (float)

{

cout << "float exception caught ";

}

catch (char)

{

cout << "char exception caught ";

}

catch (string)

{

cout << "string exception caught ";

}

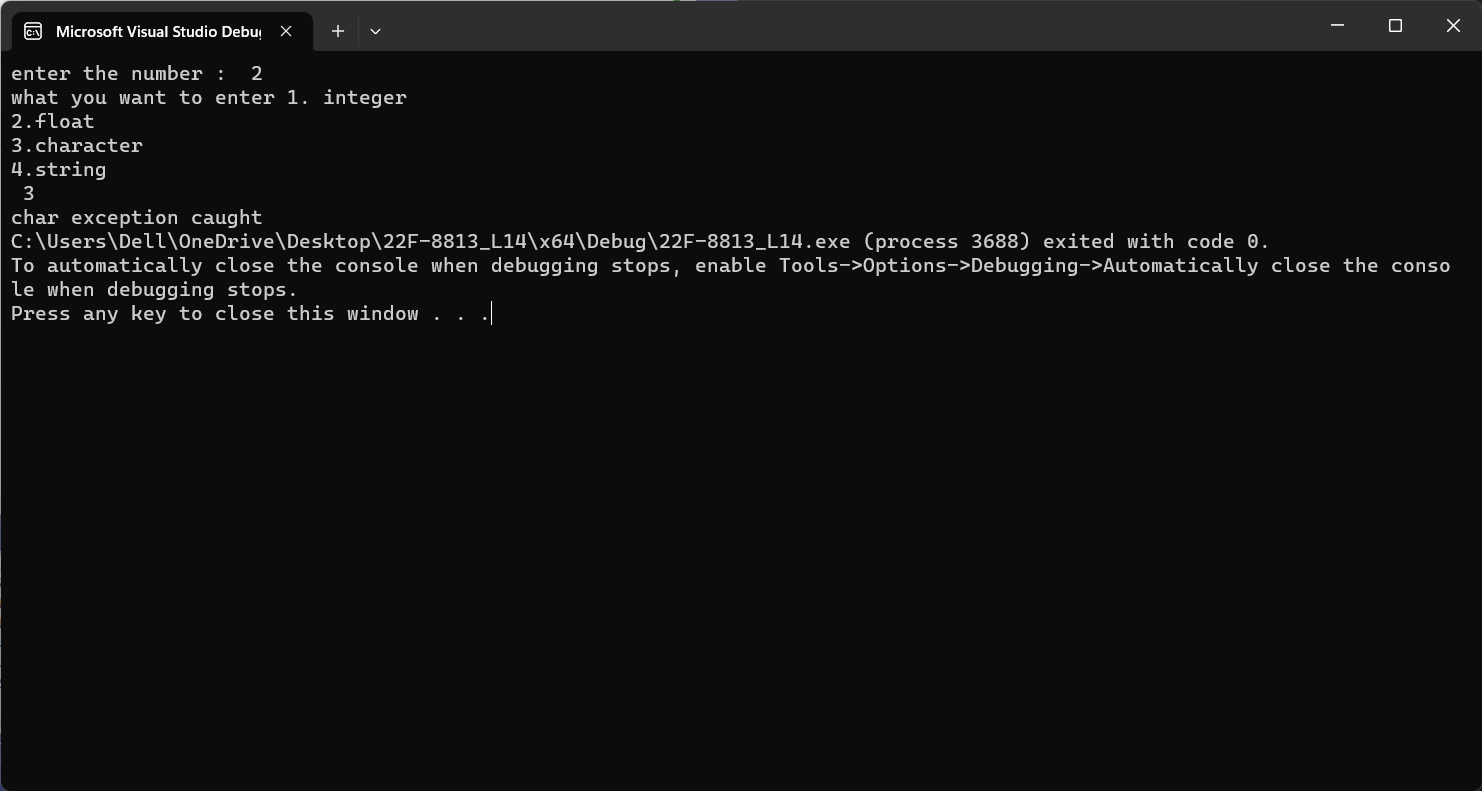
catch (...)

{

cout << " exception caught ";

}

}



Task 2:

//MAIN file (source.cpp)

#include<iostream>

#include<string>

#include"interface2.h"

using namespace std;

int multi(int n1, int n2, int d1, int d2)

{

cout << " the multiplication is : ";

return (n1 / d1)\* (n2 / d2);

}

int add(int n1, int n2, int d1, int d2)

{

cout << " the sum is : ";

return (n1 / d1) + (n2 / d2);

}

int sub(int n1, int n2, int d1, int d2)

{

cout << " the subtraction is : ";

return (n1 / d1) - (n2 / d2);

}

int div(int n1, int n2, int d1, int d2)

{

cout << " the division is : ";

return (n1 / d1) \* (d2 / n2);

}

int main()

{

int numerator1 = 0,denominator1=0,temp=0, numerator2 = 0, denominator2 = 0;

do

{

try

{

cout << "enter numerator1 : ";

cin >> numerator1;

if (numerator1 < 0)

{

throw numerator1;

}

}

catch (int)

{

cout << " Exception caught: numerator1 must not be lass than zero! ";

}

try

{

cout << "enter denominator1 ";

cin >> denominator1;

if (denominator1 <= 0)

{

throw denominator1;

}

}

catch (int)

{

cout << " Exception caught: denominator1 must not be zero! ";

}

try

{

cout << "enter numerator2 : ";

cin >> numerator2;

if (numerator2 < 0)

{

throw numerator2;

}

}

catch (int)

{

cout << " Exception caught: numerator2 must not be lass than zero! ";

}

try

{

cout << "enter denominator2 ";

cin >> denominator2;

if (denominator2 <= 0)

{

throw denominator2;

}

}

catch (int)

{

cout << " Exception caught: denominator2 must not be zero! ";

}

} while (numerator1<0||denominator1<=0|| numerator2 < 0 || denominator2 <= 0);

do

{

cout << "\n1. addition\n2.subtraction\n3. multiplication\n4. division\n5. to Exit\n";

cin >> temp;

switch (temp)

{

case 1:

{

cout << add(numerator1, numerator2, denominator1, denominator2);

break;

}

case 2:

{

cout << sub(numerator1, numerator2, denominator1, denominator2);

break;

}

case 3:

{

cout << multi(numerator1, numerator2, denominator1, denominator2);

break;

}

case 4:

{cout << div(numerator1, numerator2, denominator1, denominator2);

break;

}

case 5:

{

break;

}

default:

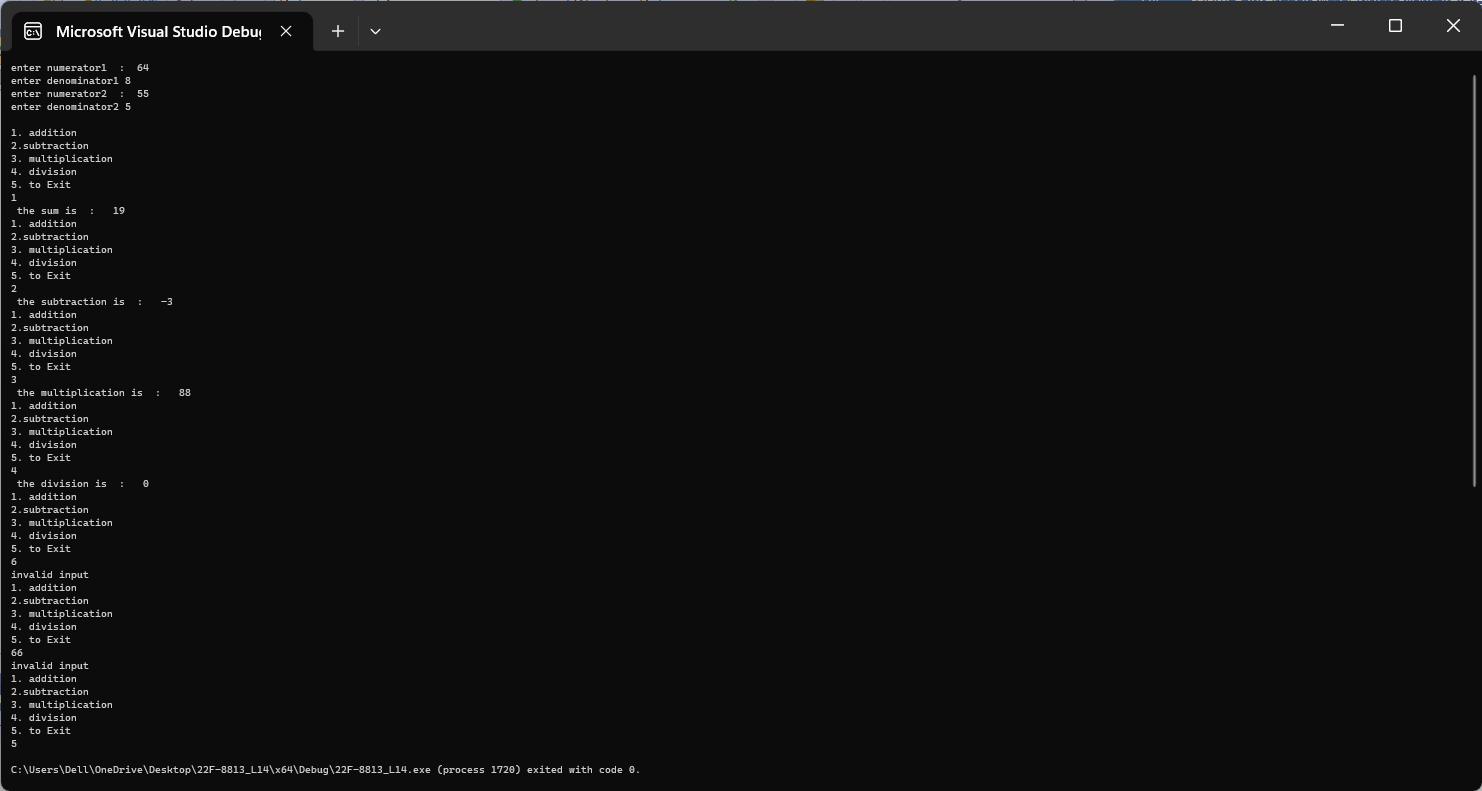
cout << "invalid input";

break;

}

} while (temp!=5);

}



Task 3:

//MAIN file (source.cpp)

#include<iostream>

#include<string>

#include"interface3.h"

using namespace std;

int main()

{

int lowerLimit=0;

try

{

cout << "Entering the try block." << endl;

cin >> lowerLimit;

if (lowerLimit < 100)

throw exception("Lower limit violation.");

cout << "Exiting the try block." << endl;

}

catch (exception eObj)

{

cout << "Exception: " << eObj.what() << endl;

}

cout << "After the catch block" << endl;

/\*

a.The value of lowerLimit is 50 ?

when lower limit is 50 than it throws an exception of exception class and catch block catches the exception of exception class, this makes an object of it and executes the statement written in the catch block and then exceutes the outer cout;it also don't execute the statement written below throw bcz after throw statement execution it does not execute the remainig below it;

b.The value of lowerLimit is 150 ?

when lower limit is 150 than it doesn't throws an exception of exception class ,it also execute the statement written below throw bcz when throw statement isn't executed then it executes the statemnet below it and outer catch block;

\*/

}

Task 4:

//MAIN file (source.cpp)

#include<iostream>

#include<string>

//#include"interface4.h"

using namespace std;

int main()

{

char ch = 'a';

int feet = 0, inch = 0, cm = 0;

do

{

try

{

cout << "\nplease enter feets : ";

cin >> feet;

ch = feet;

if (feet <= 0/\*||ch<'/'&&ch>':'\*/)

{

throw feet;

}

}

catch (int)

{

cout << "Exception caught : feet is less than or equal to zero ";

}

try

{

cout << "\npleas enter inch : ";

cin >> inch;

if (inch <= 0 /\*|| ch < '/' && ch>':'\*/)

{

throw inch;

}

}

catch (int)

{

cout << "Exception caught : inch is less than or equal to zero ";

}

} while (feet<=0||inch<=0);

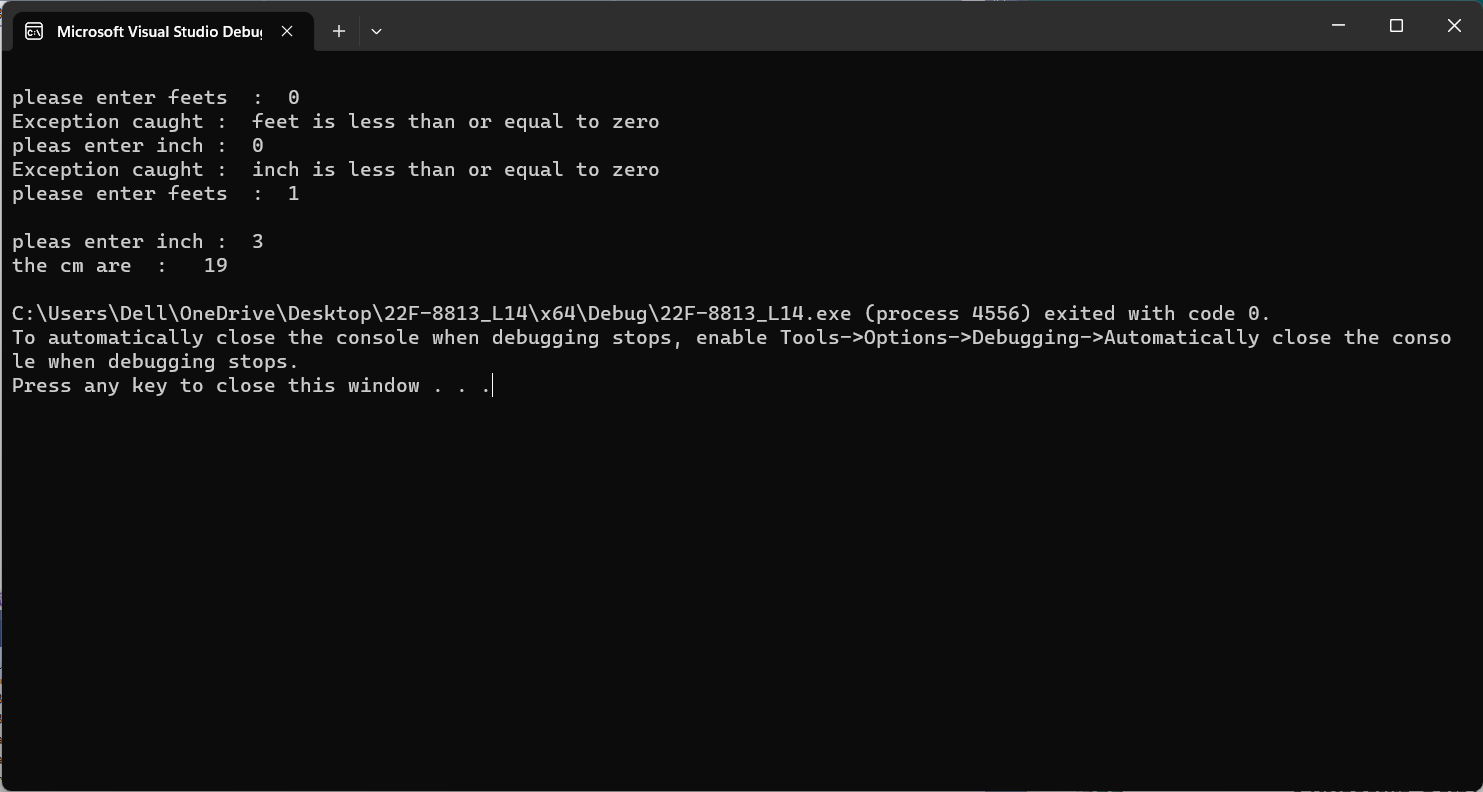
feet \*= 12;

inch \*= 2.54;

cm = feet + inch;

cout << "the cm are : " << cm << endl;

}



Task 5:

#include<iostream>

#include"interface5.h"

#include<string>

#include<stdexcept>

using namespace std;

void main()

{

string time;

int hour=0,min=0,sec=0,temp=0,m=0;

try

{

cout << "Enter time (for pm= 100 for am=100) (format: hour-minutes-sec-pm): ";

scanf\_s("%d-%d-%d-%d", &hour, &min, &sec,&m);

if (hour < 1 || hour>12)

{

throw Invalidhr();

}

if (min < 1 || min>60)

{

throw Invalidmin();

}

if (sec < 0||sec>60)

{

throw Invalidsec();

}

if (m==110)

{

hour += 12;

}

cout <<hour <<" , " << min << ", " << sec << endl;

}

catch (const exception& ex)

{

cout << "Error: " << ex.what() << endl;

}

}

#include"interface5.h"

////////const char\* Invalidsec::what() const noexcept

////////{

//////// return "Invalid seconds";

////////}

////////const char\* Invalidmin:: what() const noexcept

////////{

//////// return "Invalid minutes";

////////}

////////const char\* Invalidhr::what() const noexcept

////////{

//////// return "Invalid hour";

////////}

// Class.h file (class header file)

#pragma once

//#ifndef CLASS\_H

#include<string>

#include<iostream>

using namespace std;

//#define CLASS\_H

class Invalidhr : public exception

{

public:

const char\* what() const noexcept override

{

return "Invalid hour";

}

};

class Invalidmin : public exception

{

public:

const char\* what() const noexcept override

{

return "Invalid minutes";

}

};

class Invalidsec : public exception

{

public:

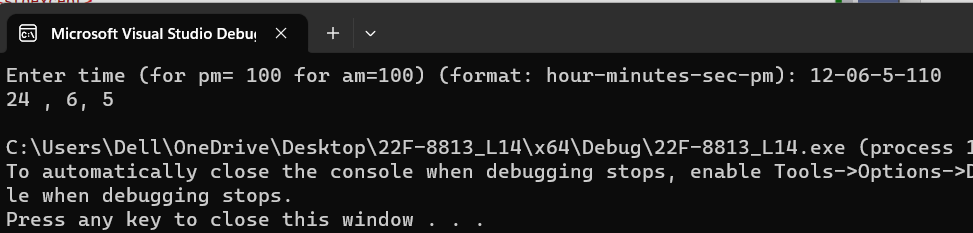
const char\* what() const noexcept override

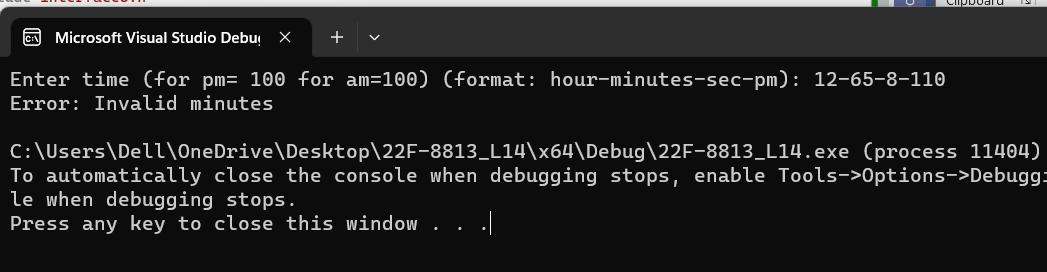
{

return "Invalid seconds";

}

};





Task 6:

#include <iostream>

#include"interface6.h"

#include <string>

using namespace std;

int main()

{

invalidday objd;

invalidmonth objm;

int d = 0, m = 0,y=0;

string dateOfBirth;

int month, day, year;

////two codes for the programme

/////as like syntax

try

{

cout << "Enter date of birth (format: MM-DD-YYYY): ";

scanf\_s("%d-%d-%d", &month, &day, &year);

if (month < 1 || month>12)

{

throw invalidmonth();

}

if (day < 1 || day>31)

{

throw invalidday();

}

if (year < 0)

{

throw InvalidYear();

}

switch (month)

{

case 1:

cout << "January ";

break;

case 2:

cout << "February ";

break;

case 3:

cout << "March ";

break;

case 4:

cout << "April ";

break;

case 5:

cout << "May ";

break;

case 6:

cout << "June ";

break;

case 7:

cout << "July ";

break;

case 8:

cout << "August ";

break;

case 9:

cout << "September ";

break;

case 10:

cout << "October ";

break;

case 11:

cout << "November ";

break;

case 12:

cout << "December ";

break;

}

cout << day << ", " << year << endl;

}

catch (const exception& ex)

{

cout << "Error: " << ex.what() << endl;

}

do

{

try

{

cout << "please enter the day : ";

cin >> d;

if (d <= 1&&d >= 31 )

{

throw objd;

}

}

catch (invalidday d)

{

d.excepd();

}

try

{

cout << "please enter the month : ";

cin >> m;

if (m >= 12 && m <= 1)

{

throw objm;

}

}

catch (invalidmonth m)

{

m.excep();

}

cout << "please enter the year : ";

cin >> y;

} while (d<=1||d>=31||m<=1||m>=12);

switch (m)

{

case 1:

{

cout << "january " << d << " , " << y;

break;

}

case 2:

{

cout << "february " << d << " , " << y;

break;

}

case 3:

{

cout << "march " << d << " , " << y;

break;

}

case 4:

{

cout << "april" << d << " , " << y;

break;

}

case 5:

{

cout << "may " << d << " , " << y;

break;

}

case 6:

{

cout << "june " << d << " , " << y;

break;

}

case 7:

{

cout << "july " << d << " , " << y;

break;

}

case 8:

{

cout << "august " << d << " , " << y;

break;

}

case 9:

{

cout << "september " << d << " , " << y;

break;

}

case 10:

{

cout << "october " << d << " , " << y;

break;

}

case 11:

{

cout << "november " << d << " , " << y;

break;

}

case 12:

{

cout << "december " << d << " , " << y;

break;

}

default:

cout << "invalid input";

break;

}

}

#include"interface6.h"

void invalidday::set(int d)

{

day = d;

}

int invalidday::get()

{

return day;

}

void invalidmonth::setm(int m)

{

month= m;

}

int invalidmonth::getm()

{

return month;

}

void invalidday::excepd()

{

cout << "exception caught";

}

void invalidmonth::excep()

{

cout << "exception caught";

}

// Class.h file (class header file)

#pragma once

//#ifndef CLASS\_H

#include<string>

#include<iostream>

using namespace std;

//#define CLASS\_H

class invalidday

{

protected:

int day;

public:

invalidday(int d = 0) :day(d){

cout << "";

}

~invalidday()

{

}

void excepd();

void set(int d);

int get();

};

class invalidmonth

{

protected:

int month;

public:

invalidmonth(int m = 0) :month(m) {}

~invalidmonth()

{

cout << "";

}

void setm(int m);

int getm();

void excep();

};

class InvalidYear : public exception

{

public:

const char\* what() const noexcept override

{

return "Invalid year";

}

};

