**Experiment No: 7 Date:- 24-11-2020**

**AIM: To study Exception Handling**

**THEORY:**

An exception is a problem that arises during the execution of a program. A C++ exception is a response to an exceptional circumstance that arises while a program is running, such as an attempt to divide by zero.

Exceptions provide a way to transfer control from one part of a program to another. C++ exception handling is built upon three keywords: **try, catch,** and **throw**.

* **throw** − A program throws an exception when a problem shows up. This is done using a **throw** keyword.
* **catch** − A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The **catch** keyword indicates the catching of an exception.
* **try** − A **try** block identifies a block of code for which particular exceptions will be activated. It's followed by one or more catch blocks.

try {

// protected code

}

catch( ExceptionName e1 ) {

// catch block

}

catch( ExceptionName e2 ) {

// catch block

}

catch( ExceptionName eN ) {

// catch block

}

You can list down multiple **catch** statements to catch different type of exceptions in case your **try** block raises more than one exception in different situations.

**A] Write a C++ program to implement exceptional handling concept (Divide by zero) using exception rethrow mechanism**

**#include<iostream>**

**using namespace std;**

**void divide(float a, float b)**

**{ try**

**{**

**if(b==0)**

**throw b;**

**else**

**cout<<"The result is:"<<a/b<<endl;**

**}**

**catch(float )**

**{**

**cout<<"Exception caught inside function"<<endl;**

**throw;**

**}**

**}**

**int main()**

**{**

**float a,b;**

**cout<<"Enter 2 elements to perform division"<<endl;**

**cout<<"Numerator:";**

**cin>>a;**

**cout<<"\nDenominator:";**

**cin>>b;**

**cout<<endl;**

**try**

**{**

**divide(a,b);**

**}**

**catch(float)**

**{**

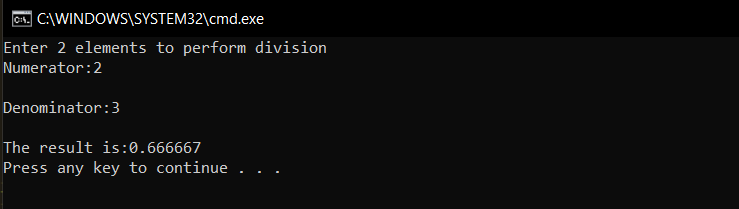
**cout<<"\nException Caught inside main"<<endl<<endl;**

**}**

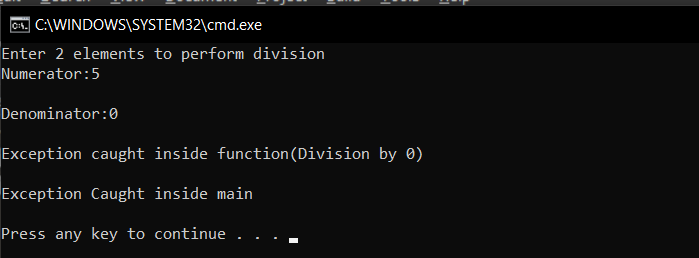
**return 0;**

**}**

**Output 1:**

****

**Output 2:**

****

**B] Write a C++ program to implement a multi catch exception handling mechanism**

#include<iostream>

using namespace std;

void func(int x)

{

try

{

if(x<0)

throw x;

else if(x==0)

throw 0.0;

else

throw 'x';

}

catch(int x)

{

cout<<"Caught an integer\n"<<endl;

}

catch(double x)

{

cout<<"Caught a double\n"<<endl;

}

catch(char x)

{

cout<<"Caught a character\n"<<endl;

}

}

int main()

{

int x;

cout<<"The value of variable, say x will be either < 0 or =0 or > 0.\nThe multiple catch statements will detect each case\n"<<endl;

cout<<"x=-1"<<endl;

func(x=-1);

cout<<"x=0"<<endl;

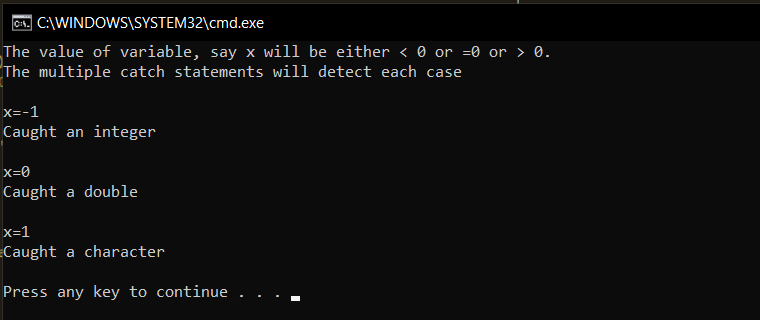
func(x=0);

cout<<"x=1"<<endl;

func(x=1);

return 0;

}



**CONCLUSION:**

* Exceptions provide a way to react to exceptional circumstances (like runtime errors) in programs by transferring control to special functions called *handlers*.
* There can be multiple catch statement, depending on the number of possible exceptions.
* If none of the catch statements are matching, then the default catch is executed.
* It makes sure that the execution of the program is not affected by the exceptions and slowly handles them without causing any issue to the program execution.