DHA SUFFA UNIVERSITY
Department of Computer Science
Object Oriented Programming
CS-1002L
Spring 2020

Lab 10
Exception Handling in C++

Objectives

To learn about exception handling in C++.

Exception Handling

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.

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.

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.

Syntax

```
try
{
      // protected code
}     catch( ExceptionName e1 )
{
      // catch block
}     catch( ExceptionName e2 )
{
      // catch block
}     catch( ExceptionName eN )
{
      // catch block
}
```

Example 10.1

```
#include<iostream>
int main()
{
   //initialzing
   int year=0;
   //taking input
   std::cout <<"Enter your year of birth";
   std::cin >>year;
   //checking if year is valid
   try
   {
      if (year>=2019)
      {
         //throw year if it is greater than or equal to 2019
         throw year;
         std::cout << "After throw (Never executed)" << std::endl;
      }
   }
   //catch year
   catch (int &year )
   {
      std::cout <<year << "\tis not a valid year" << std::endl;
   }
   return 0;
}
```

Output

```
Enter your year of birth 2020
2020 is not a valid year
```

C++ Standard exceptions
C++ provides a list of standard exceptions defined in <exception> which we can use in our programs. A small description of some of these types is listed below.

std::bad_alloc : This can be thrown by new.
std::invalid_argument : This is thrown due to invalid arguments.
std::length_error : This is thrown when a too big std::string is created.
std::out_of_range : This can be thrown by the 'at' method for example when array index value is out of range.
std::invalid_argument : This is thrown due to invalid arguments.
std::range_error : This occurs when you try to store a value which is out of range.

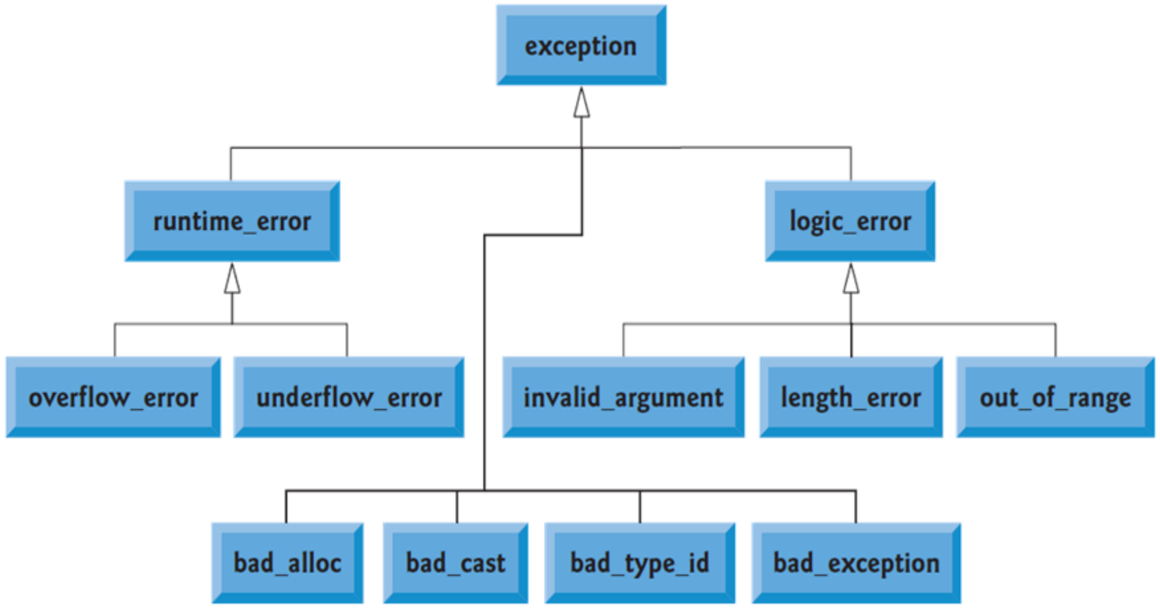

Figure 10.1 - Types of Exception

Example 10.2

```
// out_of_range example
#include <iostream>
#include <exception>
#include <array>
int main ()
{
    //creating std::array of size 10
    std::array<int,10> my_array;

    try
    {
        /*assigning value to index 15 which is invalid as array size is 10*/
        my_array.at(15)=100;
    }
    catch (std::out_of_range& err)
    {
        std::cerr << "Out of Range error: " << err.what() << std::endl;
    }

    return 0;
}
```

Output

```
Out of Range error: array::at: __n (which is 15) >= _Nm (which is 10)
```

Example 10.3

```
// bad_alloc example
#include <iostream>     // std::cout
#include <new>          // std::bad_alloc

int main () {
  try
  {
    int* myarray= new int[1000000*1000000];
  }
  catch (std::bad_alloc& ba)
  {
    std::cout << "bad_alloc caught: " << ba.what() << '\n';
  }
  return 0;
}
```

Output

```
bad_alloc caught: bad allocation
```

Example 10.4
overflow_error

```
#include <iostream>
int main( )
{
 try
 {
    int n1 = 2147483647;      //range of integer
    std::cout <<" Overflow the integer range and set in range : " << n1 + 1
    <<std:: endl;

  }
 catch (std::exception &e)
 {
   std::cout << "Caught " << e.what( ) << std::endl;

 };
}
```

Output:

```
Overflow the integer range and set in minimum range : -2147483648
```

Example 10.5

```
#include <iostream>
#include<exception>

class Exception {

public:
   Exception(const std::string& msg) : msg_(msg) {}
  ~Exception() {}

  std::string getMessage() const {return(msg_);}

private:
   std::string msg_;
};
void exceptionFunction() {
      //throw exception : In function
            try
            {
                  std::cout << "Throwing exception in exception function "
                  << std::endl;
                  throw (Exception("Example of Rethrowing Exception"));
            }
            catch (Exception &e)
            {
                  std::cout << "Exception Caught in Function:: " <<
                  e.getMessage( ) << std::endl;
                  throw;

            };
}
int main() {

   std::cout << "Program for Rethrowing Exception Handling \n";
   // try block - for exception code
   try {
      // Inside try block
      exceptionFunction();
   }// catch block
   catch (Exception &e) {
      // Code executed when exception caught
      std::cout<<"Exception Caught in main ::"<< e.getMessage() <<
std::endl;
   }

   return 0;
}
```

Output

```
Program for Rethrowing Exception Handling
Throwing exception in exception function
Exception Caught in Function:: Example of Rethrowing Exception
Exception Caught in main ::Example of Rethrowing Exception
```

DHA SUFFA UNIVERSITY
Department of Computer Science
Object Oriented Programming
CS-1002L
Spring 2020

Lab Tasks

Task 10.1

Define two functions FunctionA() and FunctionB(). In definition of FunctionA() and FunctionB() throw exceptions using try-catch block, try block of FunctionB() call FunctionA(). In main function define try block that calls FunctionB() and a catch block.

Task 10.2

Write a C++ program to throw bad allocation error using try-catch blocks.

Task 10.3

Write a C++ program to throw range_error using try-catch blocks.

Submission Instructions

1. Number your solution folders as question number e.g. Q1, Q2, etc. (Q is in upper case)
2. Every folder should contain three files (one header, one implementation and one driver)
3. Create a new folder named cs152abc where abc is your 3 digit roll #. e.g. cs152111.
4. Copy all the project folders into this folder.
5. Now make sure a zip file named cs152abc.zip is created e.g. cs152111.zip
6. Upload the assignment solution on LMS under the assignment named Lab 09
7. Assignment – XX, where XX is your section name.