Exception Handling

Exception Handling

- An exception is a problem that arises during the execution of a program.
- **Exception handling in C++** is one of the powerful mechanism to handle the runtime errors so that normal flow of the application can be maintained.
- 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

C++ exception handling is built upon three keywords: try, catch, and throw.

- **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.

- **throw:** A program throws an exception when a problem shows up. This is done using a throw keyword.

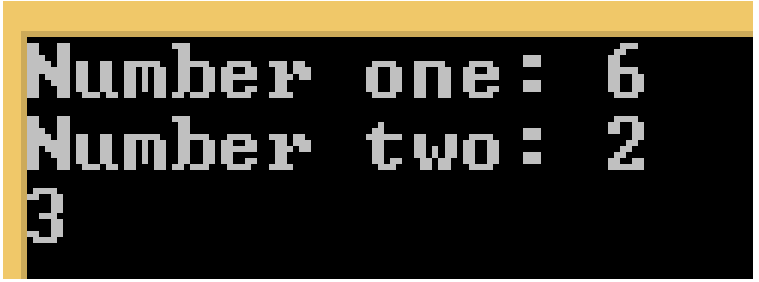
- **catch:** A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The catch keyword indicates the catching of an exception.

Syntax

```
try
{     // protected code     }

catch( ExceptionName e1 )
{    // catch block     }

catch( ExceptionName e2 )
{    // catch block     }

catch( ExceptionName eN )
{    // catch block    }
```

Normal problem example

```
#include <iostream>
 using namespace std;
 int divide_numbers(int a, int b)
{
    return a/b;
}
 int main()
{
    int a, b;
    cout << "Number one: ";
    cin >> a;
    cout << "Number two: ";
    cin >> b;
     cout << divide_numbers(a, b);
     return 0;
}
```



Exception Handling Example

```
#include <iostream>
 using namespace std;

int divide_numbers(int a, int b)
{
    if(b==0)
        throw 1;
    return a/b;
}
```

```
int main()
{
    int a, b;
    cout << "One: ";
    cin >> a;
    cout << "Two: ";
    cin >> b;
    try {
        cout << divide_numbers(a, b);
    }
    catch(int  code) {
        cout << "ERROR CODE: " << code;
    }
    //Continue doing whatever afterwards like normal
    return 0;
}
```

```
One: 6
Two: 2
3
```

```
One: 6
Two: 0
ERROR CODE: 1
```

Exception Handling Example(cont..)

```
#include <iostream>
using namespace std;
int main()
{
cout << "Start\n";
try { // start a try block
    cout << "Inside try block\n";
    throw 100;                  // throw an error
    cout << "This will not execute";
}
```

```
catch (int i) { // catch an error
    cout << "Caught an exception -- value is: ";
    cout << i << "\n";
}
    cout << "End";
return 0;
}
```

Exception Handling Example(cont..) is not work

```
#include <iostream>
using namespace std;
int main()
{
cout << "Start\n";
try {              // start a try block
    cout << "Inside try block\n";
    throw 100;              // throw an error
    cout << "This will not execute";
}
```

```
catch (double i) {  // won't work for an int exception
    cout << "Caught an exception -- value is: ";
    cout << i << "\n";
}
    cout << "End";
return 0;
}
```

Exception Handling Looping Example

```
#include <iostream>
using namespace std;
int main ()
{
   int x = 50;
   int y = 0;
   int i;
   double z;
   try {
     for (i=1;i >0;i++){
      cout << "Value is  " << i  << endl;
    // throw i;
     }
}
```

```
 catch (int e) {
     cout << "Throwing Exception occur " << endl;
   }


   return 0;
}
```

Catching All Exceptions

```
#include <iostream>
using namespace std;
void Xhandler(int test)
{
try{
if(test==0) throw test;          // throw int
if(test==1) throw 'a';           // throw char
if(test==2) throw 123.23;        // throw double
}
catch(...) { // catch all exceptions
cout << "Caught One!\n";
}
}
```

```
int main()
{
cout << "Start\n";
Xhandler(0);
Xhandler(1);
Xhandler(2);
cout << "End";
return 0;
}
```

```
Start
Caught One!
Caught One!
Caught One!
End
```

Catching All Exceptions

```
#include <iostream>
using namespace std;
void Xhandler(int test)   {
try{
if(test==0) throw test;                // throw int
if(test==1) throw 'a';                 // throw char
if(test==2) throw 123.23;              // throw double
}
catch(int i) {                 // catch an int exception
cout << "Caught an integer\n";
}
catch(...) {                   // catch all other exceptions
cout << "Caught One!\n";
}   }
```

```
int main()
{
cout << "Start\n";
Xhandler(0);
Xhandler(1);
Xhandler(2);
cout << "End";
return 0;
}
```