Exception Handling in C++

Types of Errors

In C++, we can encounter with three types of errors

- Syntax Error
- Logical Error
- Runtime Error or Exception

Types of Errors

In C++, we can encounter with three types of errors

- Syntax Error
- Logical Error
- Runtime Error or Exception

Syntax Error:

- Such errors which are arises due to poor understanding of the language rules are called Syntax or compile time errors.
- E.g. Missing semicolon,undefined variables etc.

Logical Errors:

- Such errors arises due to poor understanding of the problem and solution procedure.
- Logical errors are not detected or reported by the compiler
- When we run a program which have logical error, it generates unexpected output.

Types of Errors

Runtime Error or Exception:

- Sometime we face some errors at the time of program execution, other than syntax error or logical error, such errors are called runtime error or exception.
- Compiler can not detect exceptions.
- When a program have exception then it abnormally terminated during runtime.
- The exceptions which can be handled by the program, such as array index out of range, dividing an integer by zero are called as synchronous exceptions, and the exceptions which can not be handled the program,such as input device failure etc, are called Asynchronous exceptions.
- In C++, only synchronous exceptions can be handled within a program

Types of Errors

Runtime Error or Exception:

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

int main()
{
    float a,b,c;
    cout<<"Enter two numbers : ";
    cin>>a>>b;
    c=a/b;
    cout<<a<<"/"<<b<<"="<<c<<endl;
    cout<<"Program Terminated successfully!!"<<endl;
    return(0);
}
```

Exception Handling

Exception Handling is a mechanism which allows us to detect the exception and prevent our program from abnormal termination at runtime. C++ exception handling mechanism is built upon following three keywords

try: The try statement allows you to define a block of code to be tested for errors while it is being executed.

throw: The throw keyword throws an exception when a problem is detected, which lets us create a custom error.

catch: The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.

Exception Handling

The try and catch keywords come in pairs:

```
try {

  // Block of code to try

  throw exception; // Throw an exception when a problem arise

}

catch () {

  // Block of code to handle errors

}
```

Exception Handling

```
int main()
{   int a,b,c;
    cout<<"Enter two number : ";
    cin>>a>>b;
    try
    {   if(b==0)
             throw b;
        else
        {      c=a/b;
               cout<<"Result : "<<c;
        }
    }
    catch(int x)
    {     cout<<"\nCan't  divide by "<<x;
    }
     return(0);
}
```

Exception Handling

```
int main() {
  try {
    int age = 15;
    if (age >= 18) {
      cout << "Access granted - you are old enough.";
    } else {
      throw (age);
    }
  }
  catch (int myNum) {
    cout << "Access denied - You must be at least 18 years old.\n";
    cout << "Age is: " << myNum;
  }
  return 0;
}
```

Handle Any Type of Exceptions (...)

If you do not know the throw type used in the try block, you can use the "three dots" syntax (...) inside the catch block, which will handle any type of exception:

```
int main() {
  try {
    int age = 15;
    if (age >= 18) {
      cout << "Access granted - you are old enough.";
    } else {
      throw 505;
    }
  }
  catch (...) {
    cout << "Access denied - You must be at least 18 years old.\n";
  }
  return 0;
}
```