Unit 3. Exception Handling (4L,14M)

➢ Concept of Exception Handling mechanism –

In C++, exceptions are runtime anomalies or abnormal conditions that a program encounters during its execution. The process of handling these exceptions is called exception handling. Using the exception handling mechanism, the control from one part of the program where the exception occurred can be transferred to another part of the code.

What is a C++ Exception?
An exception is an unexpected problem that arises during the execution of a program our program terminates suddenly with some errors/issues. Exception occurs during the running of the program (runtime).

There are two types of exceptions in C++

1. **Synchronous:** Exceptions that happen when something goes wrong because of a mistake in the input data or when the program is not equipped to handle the current type of data it's working with, such as dividing a number by zero.

1. **Asynchronous**: Exceptions that are beyond the program's control, such as disc failure, keyboard interrupts, etc.

C++ Exception Handling Keywords

In C++, we use 3 keywords to perform exception handling:

- try
- catch, and
- throw

➢ C++ try and catch

C++ provides an inbuilt feature for Exception Handling. It can be done using the following specialized keywords: try, catch, and throw with each having a different purpose.

Syntax of try-catch in C++

```
try {
    // Code that might throw an exception
```

```
    throw SomeExceptionType("Error message");
  }
  catch( ExceptionName e1 )  {
    // catch block catches the exception that is thrown from try block
  }
```

1. try in C++

The try keyword represents a block of code that may throw an exception placed inside the try block. It's followed by one or more catch blocks. If an exception occurs, try block throws that exception.

2. catch in C++

The catch statement represents a block of code that is executed when a particular exception is thrown from the try block. The code to handle the exception is written inside the catch block.

3. throw in C++

An exception in C++ can be thrown using the throw keyword. When a program encounters a throw statement, then it immediately terminates the current function and starts finding a matching catch block to handle the thrown exception.

In C++ programming, exception handling is performed using try/catch statement. The C++ **try block** is used to place the code that may occur exception. The **catch block** is used to handle the exception.

Ex.

```
#include <iostream>
using namespace std;
float division(int x, int y) {
  return (x/y);
}
int main () {
  int i = 50;
  int j = 0;
  float k = 0;
    k = division(i, j);
    cout << k << endl;
  return 0;
}
```

Output:

```
Floating point exception
```

Multiple catch statements can be used to catch different type of exceptions thrown by try block.

Why do we need Exception Handling in C++?

1. *Separation of Error Handling Code from Normal Code*
1. *Functions/Methods can handle only the exceptions they choose*
1. *Grouping of Error Types*

// C++ program to demonstate the use of try,catch and throw in exception handling.

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

double division(int a, int b) {
  if( b == 0 ) {
     throw "Division by zero condition!";
  }
  return (a/b);
}

int main () {
  int x = 50;
  int y = 0;
  double z = 0;

  try {
     z = division(x, y);
     cout << z << endl;
  } catch (const char* msg) {
    cerr << msg << endl;
  }

  return 0;
}
```