EXCEPTION HANDLING

- When executing C++ code, different errors can occur: coding errors made by the programmer, errors due to wrong input, or other unforeseeable things.

- When an error occurs, C++ will normally stop and generate an error message. The technical term for this is: C++ will throw an exception (throw an error).

- Exception is an event which occurs during execution of program that disrupts normal flow of program

HOW TO HANDLE THE EXCEPTION: EXCEPTION HANDLING

- Exception Handling is a process to handle runtime errors.

- We perform exception handling so the normal flow of the application can be maintained even after runtime errors.

- exception is an event or object which is thrown at runtime.

- All exceptions are derived from **exception** class. It is a runtime error which can be handled. If we don't handle the exception, it prints exception message and terminates the program.

- Example of Exception like Divide by zero, Accessing array element beyond its limit, running out of memory etc.

- Exception handling mechanism consists of followin parts:

1) Find the problem(Hit the Exception)
2) Inform about its occurrence(Throw the exception)
3) Receive error information(Catch the exception)
4) Take proper action(Handle the exception)

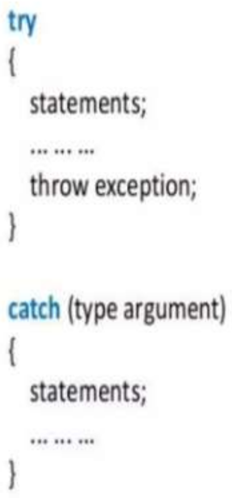
- **Exception Handling can be done in 3 ways:**

1) **<u>try block</u>**: try block is used to place the code that may occur exception. Exception are thrown inside the try block.

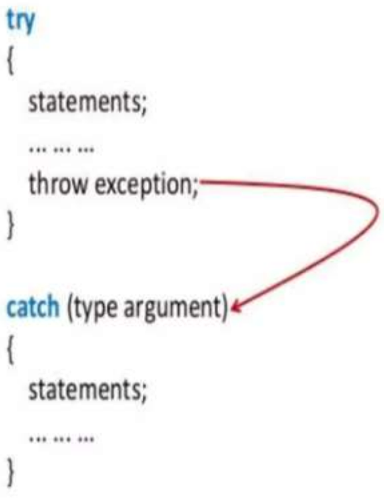
2) **<u>catch block</u>**: catches an exception which is thrown from try block. The catch keyword indicates the catching of an exception. It is used to handle the exception. It defines action to be taken when exception occur

3) **<u>throw keyword:</u>** throws an exception when a problem is detected.

SYNTAX OF EXCEPTION HANDLING

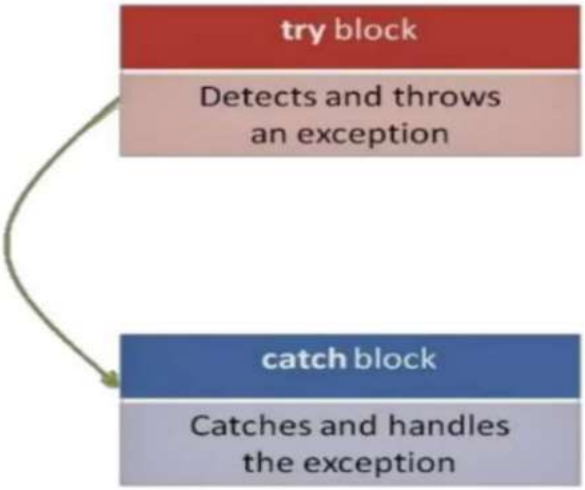


```
try
{
  statements;
  ... ... ...
  throw exception;
}

catch (type argument)
{
  statements;
  ... ... ...
}
```



```
try
{
  statements;
  ... ... ...
  throw exception;
}

catch (type argument)
{
  statements;
  ... ... ...
}
```

EXCEPTION HANDLING MECHANISM



try block

Detects and throws an exception

catch block

Catches and handles the exception

EXAMPLE WITHOUT EXCEPTION HANDLING

```
#include<iostream>
void main()
{
    int a,b;
    a=10;
    b=a/0;  //exception occurs
     cout<<”result: “<<b;
}
```

Here program will be terminated you will not get output because exception occurs at line 6 and flow of program comes out of main() function without executing line 7.

Exception Handling

```
#include<iostream>
void main()
{
    int a,b;
    a=10;
   try
  {
    b=a/0;  //exception occurs
   }
   catch(const char* e)
   {
     cout<<"Divide by zero error" //exception handler code
    }
}
```