

## Objectives:

- To gain an understanding of Exception handling in C++.

## Tools and Libraries Used:

- Programming Language: C++
- IDE: G++
- Libraries: include <iostream>, include <string>

## Theory:

Exception handling in C++ is a mechanism that allows a program to detect and manage runtime errors gracefully, without crashing. It separates error-handling code from the main logic, improving readability and robustness.

C++ uses three main keywords:

- try: Defines a block of code to test errors.
- throw: Signals the occurrence of an exception.
- catch: Defines a block of code that handles the error.

Syntax:

```
try {  
    throw "Error occurred";  
}  
catch (const char* msg) {  
    cout << msg;  
}
```

You can throw and catch different types: integers, strings, objects, or even custom exception classes.

Benefits:

- Handles errors without terminating the program.
- Promotes clean, maintainable code.

## Lab Questions:

### Q no 1:

Write a C++ program to demonstrate multiple catch blocks handling different data types. Throw and handle int, char, and string type exceptions in separate catch blocks.

Code:

```
1. #include <iostream>
2. #include <string>
3. using namespace std;
4. int main() {
5.     try {
6.         int choice;
7.         cout << "Enter 1 to throw int,2 to throw char,3 to throw string:
8.         ";
9.         cin >> choice;
10.        if (choice == 1)
11.            throw 100;
12.        else if (choice == 2)
13.            throw 'W';
14.        else if (choice == 3)
15.            throw string("This is a string exception.");
16.        else
17.            cout << "Invalid choice!" << endl;
18.    }
19.    catch (int e) {
20.        cout << "Integer exception: " << e << endl;
21.    }
22.    catch (char e) {
23.        cout << "Char exception: " << e << endl;
24.    }
25.    catch (string &e) {
26.        cout << "String exception: " << e << endl;
27.    }
28.    cout << "Program continues after exception handling." << endl;
29.    return 0;
30. }
```

Output:

```
Enter 1 to throw int,2 to throw char,3 to throw string: 1
Integer exception: 100
Program continues after exception handling.
PS C:\Users\HP\Downloads\Personal\2nd Semester\OOP\Lab Programs\
Enter 1 to throw int,2 to throw char,3 to throw string: 2
Char exception: W
Program continues after exception handling.
PS C:\Users\HP\Downloads\Personal\2nd Semester\OOP\Lab Programs\
Enter 1 to throw int,2 to throw char,3 to throw string: 3
String exception: This is a string exception.
Program continues after exception handling.
```

**Q no 2:**

Write a program using catch-all handler (catch(...)) to handle any kind of exception.

Illustrate a case where an unexpected data type is thrown and caught generically.

**Code:**

```
1. #include <iostream>
2. using namespace std;
3. int main() {
4.     try {
5.         int choice;
6.         cout << "Enter 1 for int, 2 for double, 3 for char, 4 for
bool: ";
7.         cin >> choice;
8.         if (choice == 1)
9.             throw 100;
10.        else if (choice == 2)
11.            throw 3.14;
12.        else if (choice == 3)
13.            throw 'Z';
14.        else if (choice == 4)
15.            throw true;
16.        else
17.            throw "Unknown type";
18.    }
19.    catch (int e) {
20.        cout << " Int exception: " << e << endl;
21.    }
22.    catch (...) {
23.        cout << "Here is an exception of unknown or unexpected type!"
<< endl;
24.    }
25.    cout << "Now the program continues." << endl;
26.    return 0;
27. }
```

**Output:**

```
Enter 1 for int, 2 for double, 3 for char, 4 for bool: 1
Int exception: 100
Now the program continues.
PS C:\Users\HP\Downloads\Personal\2nd Semester\OOP\Lab Programs
Enter 1 for int, 2 for double, 3 for char, 4 for bool: 2
Here is an exception of unknown or unexpected type!
Now the program continues.
PS C:\Users\HP\Downloads\Personal\2nd Semester\OOP\Lab Programs
Enter 1 for int, 2 for double, 3 for char, 4 for bool: 3
Here is an exception of unknown or unexpected type!
Now the program continues.
PS C:\Users\HP\Downloads\Personal\2nd Semester\OOP\Lab Programs
Enter 1 for int, 2 for double, 3 for char, 4 for bool: 4
Here is an exception of unknown or unexpected type!
Now the program continues.
```

## **Conclusion:**

Understanding exception handling in C++ is essential for writing robust and error-resilient programs. Through this lab, we learned how to:

- Use try, throw, and catch blocks to handle runtime errors gracefully.
- Catch and manage different types of exceptions, including custom types.
- Maintain program flow without abrupt termination during unexpected errors.

These concepts are crucial for developing reliable and maintainable applications. Mastery of exception handling ensures that programs can deal with errors effectively, improving both user experience and code quality.