Task 7: Exception Handling Basics in C++

Objective:
To understand and implement basic exception handling in C++ by handling division by zero using try-catch blocks. This task demonstrates how runtime errors can be handled gracefully without crashing the program.

1. Concept Overview

Exception handling in C++ is used to handle runtime errors and abnormal conditions. The try block contains code that may generate an exception, the throw statement throws an exception, and the catch block handles the exception.

2. Program Code

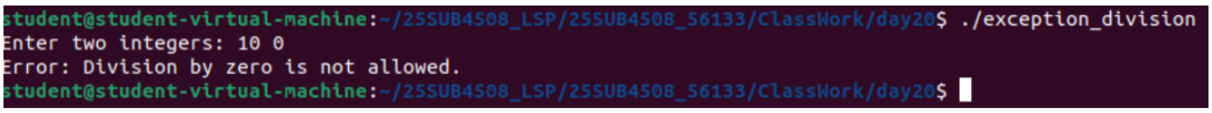
```
#include <iostream>
 using namespace std;

 int divide(int a, int b) {
        if (b == 0)
        throw b;
        return a / b;
 }

 int main() {
        int x, y;
        cout << "Enter two integers: ";
        cin >> x >> y;

        try {
        int result = divide(x, y);
        cout << "Result = " << result << endl;
        }
        catch (int e) {
        cout << "Error: Division by zero is not allowed." << endl;
        }

        return 0;
 }
```

3. Compilation Instructions

Compile the program using a C++ compiler:

```
g++ exception_division.cpp -o exception_division
```

4. Sample Outputs



```
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day20$ ./exception_division
Enter two integers: 10 0
Error: Division by zero is not allowed.
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day20$ 
```

5. Observations & Explanation

1. The divide() function checks if the denominator is zero.
2. If zero is detected, an exception is thrown using the throw keyword.
3. The try block attempts to execute the division operation.
4. The catch block catches the exception and displays an error message.
5. The program continues execution without crashing.

6. Advantages of Exception Handling

- Prevents program termination due to runtime errors
- Improves program reliability
- Separates error-handling logic from normal logic

7. Conclusion

This task demonstrates the use of basic exception handling in C++. By using try-catch blocks, runtime errors such as division by zero can be handled safely and efficiently.