

## C++ Exception Handling

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

void main()
{
    int x,y;
    cout<<"Enter the value of x:";
    cin>>a>>b;

    cout<<"Enter the value of y:";
    cin>>b;

    try
    {
        if(y!=0)
        {
            cout<<"Result=" << x/y;
        }
        else
        {
            throw(y);
        }
    }
    catch(int ex)
    {
        cout<<"Caught an exception which is:" << y ;
    }
}
```

```
    }  
}  
}
```

**OUTPUT:**

Enter the value of a:

4

Enter the value of b;

0

Caught an exception which is:0

**First we should know what is exception handling?**

Ans. When in a program a situation occurred when program is not able to take decision or result is out of control. Than such conditions are needed to be handled to prevent our program. The reason behind such conditions is known as exceptions.

Some of the examples of exceptions are:

- When divided by zero(o), which is an exception.
- When doing operation of a number with a character, is also an exception.
- Pressing a login button may time after entering credentials, could also be an exception.

So, for handling exceptions in a program which may be in C++, JAVA etc. Concept of exception handling is used.

In exception handling, try and catch blocks are used. In try, throw is also used.

The try block will execute the program, and if any exception occurs it will throw the exception and program control will entered into catch block.

The try block is always used to execute the program and catch block will always handled the exceptions thrown by try block.

### MCQs on Exception Handling

Q1. The class at the top of exception class hierarchy is \_\_\_\_\_.

- A. Arithmetic exception
- B. Throwable
- C. Object
- D. Exception

### MCQs Answers

Q1. (B)

### References:

1. Sebesta, "Concept of programming Language", Pearson Edu
2. Loudon, "Programming Languages: Principles & Practices" , Cengage Learning
3. Tucker, "Programming Languages: Principles and paradigms ", Tata McGraw -Hill.
4. E Horowitz, "Programming Languages", 2nd Edition, Addison Wesley

### Related Posts:

1. Sequence Control & Expression | PPL

2. PPL:Named Constants
3. Parse Tree | PPL | Prof. Jayesh Umre
4. Basic elements of Prolog
5. Loops | PPL | Prof. Jayesh Umre
6. Subprograms Parameter passing methods | PPL | Prof. Jayesh Umre
7. Programming Paradigms | PPL | Prof. Jayesh Umre
8. Subprograms Introduction | PPL | Prof. Jayesh Umre
9. Phases of Compiler | PPL | Prof. Jayesh Umre
10. Parse Tree | PPL
11. Influences on Language design | PPL | Prof. Jayesh Umre
12. Fundamentals of Subprograms | PPL | Prof. Jayesh Umre
13. Programming Paradigm
14. Influences on Language Design
15. Language Evaluation Criteria
16. OOP in C++ | PPL
17. OOP in C# | PPL
18. OOP in Java | PPL
19. PPL: Abstraction & Encapsulation
20. PPL: Semaphores
21. PPL: Introduction to 4GL
22. PPL: Variable Initialization
23. PPL: Conditional Statements
24. PPL: Array
25. PPL: Strong Typing
26. PPL: Coroutines
27. PPL: Exception Handler in C++
28. PPL: OOP in PHP

29. PPL: Character Data Type
30. PPL: Exceptions
31. PPL: Heap based storage management
32. PPL: Primitive Data Type
33. PPL: Data types
34. Programming Environments | PPL
35. Virtual Machine | PPL
36. PPL: Local referencing environments
37. Generic Subprograms
38. Local referencing environments | PPL | Prof. Jayesh Umre
39. Generic Subprograms | PPL | Prof. Jayesh Umre
40. PPL: Java Threads
41. PPL: Loops
42. PPL: C# Threads
43. Pointer & Reference Type | PPL
44. Scope and lifetime of variable
45. Design issues for functions
46. Parameter passing methods
47. Fundamentals of sub-programs
48. Subprograms
49. Design issues of subprogram
50. Garbage Collection
51. Issues in Language Translation
52. PPL Previous years solved papers
53. Type Checking | PPL | Prof. Jayesh Umre
54. PPL RGPV May 2018 solved paper discussion| Prof. Jayesh Umre
55. PPL Viva Voce

- 56. PPL RGPV June 2017 Solved paper | Prof. Jayesh Umre
- 57. Concurrency
- 58. Basic elements of Prolog
- 59. Introduction and overview of Logic programming
- 60. Application of Logic programming
- 61. PPL: Influences on Language Design
- 62. Language Evaluation Criteria PPL
- 63. PPL: Sequence Control & Expression
- 64. PPL: Programming Environments
- 65. PPL: Virtual Machine
- 66. PPL: Programming Paradigm
- 67. PPL: Pointer & Reference Type
- 68. try-catch block in C++