

Try and Catch Blocks Exercises

- Complete the following statement:
 - A try block can be followed by several catch blocks, each of which handles an exception instance with a different (...)
- If more than one of the catch blocks is able to handle a certain exception, how does the program decide which one to invoke?

- In the following code, which handler is invoked, and why?

```
try {  
    vector<int> v;  
    cout << v.at(2) << endl;           // May throw an exception of type std::out_of_range  
}  
catch (const exception& e) {  
    cout << "std::exception\n";  
}  
catch (const out_of_range& e) {  
    cout << "std::out_of_range\n";  
}
```

- When writing an exception handler, how should the exception instance be passed to it? Give a reason for your answer
- What guidelines should we follow when writing an exception handler?

- What happens if an exception is thrown in a try block and none of the associated catch blocks can handle it?

- Write a simple program which throws an exception inside a nested try block, which can only be handled by one of the outer catch blocks
- Write a simple program which calls a function. The function throws an exception inside a try block, which can only be handled by a catch block in the calling function