

# More About Exceptions Solutions

- What happens if a destructor throws an exception?
  - Only one stack unwinding process can occur at a time
  - If the destructor is called after an exception has already been thrown, there is a stack unwind in progress
  - If the destructor then throws another exception which results in a stack unwinding, there will be two stack unwindings at the same time
  - Undefined behaviour

- If you have a destructor that needs some code which might throw an exception, how should you write it?
  - The destructor must handle the exception internally, without causing a stack unwinding
  - This is done by providing a try/catch block which does not rethrow the exception (or throw a fresh exception)

- What happens if a constructor throws an exception?
  - If the exception is not handled inside the constructor, the partially completed object will be destroyed, along with its data members and any base class parts
  - An object does not exist (as far as the rest of the program is concerned) until its constructor has successfully completed

- If you have a constructor that needs some code which might throw an exception, how should you write it?
  - Provided you still want the object to be created, the exception should be handled internally
  - This is done by providing a try/catch block which does not rethrow the exception (or throw a fresh exception)

- Imagine we create an instance of a class whose constructor throws an unhandled exception. We create the instance inside a try/catch block which can handle this exception

- What happens if
  - The constructor throws an exception
    - The instance is not created
    - A stack unwinding process begins
    - This will destroy all the local variables in the try block
    - The program looks for a suitable handler for the exception
  - The constructor does not throw an exception
    - The instance has been (presumably successfully) created and can be used
    - The program continues executing normally

- Write a simple class whose constructor throws an exception
- Write a program which creates an instance of this class inside a try/catch block which handles this exception
- Run your program. Compare the results when the constructor throws an exception with the results when it does not throw



- What is a catch all handler?
  - A catch all handler will handle any exception of any type
- Give an example of when a catch all handler could be useful
  - It can be useful to put a catch all handler after a complex sequence of catch blocks, or after main()
  - This will help detect any unhandled exceptions
- Write a simple program which shows how a catch all handler can be useful