

# Exceptions Introduction Solutions

# Exception Example (unhandled)

- Write a simple program in which there is a run-time error condition, but there is no exception-handling code
- What happens when this program is run? Explain your results
  - An exception object is created when the error occurs
  - Because there is no code to handle the exception, the program is terminated

# Exception Handling

- What keyword do we use to tell the compiler to generate exception-related code?
  - `try`
- What keyword do we use to inform the compiler that some code is an exception handler?
  - `catch`

# Catch Block

- How do we specify which type of exception the code should handle?
  - The type of exception we are going to handle goes in brackets after the catch keyword
- We want to write an exception handler that can deal with different sub-types of an exception class. Is there an easy way to do this?
  - Use the base class as the exception type in the catch block
- How can we use dynamic binding in an exception handler?
  - Use a reference to the base class as the exception type in the catch block

# Exception Example (handled)

- Rewrite your program from the first exercise so that it handles the exception
- Make it print out a message which describes the error condition

# Uncaught Exceptions

- What happens if an exception is thrown in a try block and none of the associated catch blocks can handle it?
  - The program will leave the current scope and look in the enclosing scope for a suitable handler
  - If it does not find one there, it will leave that scope and look in that scope's enclosing scope
  - If the program leaves main()'s scope without finding a suitable handler, the program terminates (by default)