Lec 9. Chapter 16

Nov. 8, 2025

- Ch. 15 Quiz opens today until Tuesday
- W3
  - ↳ use C++ Reference page
- Final
  - ↳ know Error management well (Chapter 16)
- Exceptions
  - ↳ something unexpected occurred
- Terminology
  - Exception
  - throw an exception
  - catch/handle an exception: process the exception; interpret the signal
- Keywords
  - ↳ throw
  - ↳ try
  - ↳ catch: Takes a parameter that matches the type thrown
- Match will create an execution (match related to exception thrown)
- catch is your interaction with the user
- Program 16-1*
- write fxn for catch block in helper fxn
- your calling is in try
- the throw part is in the fxn that try calls
- static_cast<datatype>(numerator)
  - ↳ converts one data type to another
- exceptions
  - ↳ new no name needed for parameter def, catch block param def does need type of exception being caught

- two situations where except not caught
  - thrown outside try block
  - no matching exception type
- exception class
  - can have more than one exception class
  - no members (signal) vs members (pass data)
    - usually to signal to user
- Rectangle files (v1)
- 16-2 (use the class method on final)
- unwinding stack
  - if you program correctly, return 0
  - essentially undoing the stacked fxns it processed
- nested try blocks
  - can have throws and catches layered
- Function Templates
  - pattern that can work w/ many data types
- template <class T> (template prefix)

  T times10 (T num) generic data type

  {return 10*num;} must be "T"

- can overload templates
- Class Templates (use on final)
  - unlike w/ fxns, classes are instantiated by supplying the type info
- class Square : public Rectangle <T>

HW

Nov. 22 will get project

- 15-7
- 16-11 (simple Vector)
- 15-12 c