

CMSC 341

C++ & Makefile Review

Object Relations

“Uses a”

- An object uses another object by calling a public method of that object

“Has a”

- Implemented using composition (aggregation)
- i.e. object Foo has an object Bar as a data member

“Is a”

- An object builds off of a base object to extend its functionality (inheritance)
- Typically derived class is a specialized version of its base class

Inheritance

- Single Inheritance
- Multiple Inheritance
 - The “Diamond Problem”

Exceptions

- The **author** of a library/class can detect run-time errors, but does not in general know what to do with them
- The **user** of a library/class can cope with such errors, but can not detect them (otherwise they would have been handled in the users code and not left to the library to find)

Exceptions

- Notion of an **exception** is provided to deal with such problems
- General idea is that when a function/method encounters a problem it can not cope with, it *throws* an exception, hoping that its caller (indirect or direct) can handle the problem

Exception Alternatives

- Terminate the program
- Return a value representing an error
- Return a legal value and leave the program/object in an illegal state
- Call a function to be supplied in case of an error

Exception Benefits

- Removes error handling code from the code that caused the error (less clutter)
- Makes it possible to catch all kinds of errors, errors of a certain type, or errors of related types
- Usually used in situations in where the system can recover
- Used when the error will be dealt with by a different part of the program (i.e., different scope) from that which detected the error

Exception Examples

- Throwing / Catching exceptions
 - General Form
- Grouping of exceptions
- Order of catching
- Complex exceptions

Makefiles

- The **make** command allows you to manage large programs or groups of programs
- As you begin to write larger programs, you will begin to notice that you usually only work on a small section of the program, and the remainder of the program remains unchanged
- The **make** program helps you in developing large programs by keeping track of which portions of the entire program have been changed and compiling only those parts which have changed since the last compile

Makefiles

- Dependency Graph
- How it works
- Targets / Dependencies
- Main Target
- Macros
- Phony Targets