CS 246 Fall 2019 — Tutorial 7

November 4th, 2019

Summary

1 Class Relationships

Composition (OWNS-A): Class A *owns an* instance of class B. Class A is responsible for deleting the instance of class B when an object of class A is destroyed.

Aggregation (HAS-A): class A *has an* instance of class B. Class A is not responsible for deleting the instance of class B.

Inheritance (IS-A): class B *is a* class A. This means that an instance of class B can be used in any situation where an instance of class A can be used.

Exercise: How are each of these relations impelmented in C++?

2 Inheritance

Syntax: `class B : public A`. B *is an* A, and hence can be used as one when passing arguments to functions, etc. To pass arguments to A's constructor from B, use an entry `A{...}` at the start of the MIL.

Exercise: Try writing a simple example of inheritance, exploring cases when the parent class has and does not have a default constructor.

3 Accessiblity

Fields and methods in a class marked private are not accessible to other classes. Fields and methods marked protected are accessible to subclasses. Fields and methods marked public are accessible to all.

Question: What's the point of controlling accessibility?

4 Virtual and Pure Virtual Methods

Any function can be overridden in a subclass. It's generally a poor idea to do this for non-virtual functions, as they will not work polymorphically. Virtual functions will behave as we expect when overridden.

Pure virtual functions *must* be overridden by subclasses. Syntax: `virtual T f(...) = 0;`.

A class cannot be instantiated if it has pure virtual methods. Such classes are called *abstract*, and a class that is not abstract is *concrete*. Generally, it's a good idea to only inherit from one class and to have that class be abstract. Thus if the inheritance relations in a UML form a tree, the non-leaf nodes should typically be abstract.

Exercise: Write an abstract base class `Animal` with a pure virtual `speak` method, and child classes `Dog` and `Cat`. Write a function `meet` which takes two animals and has them speak to each other.