Tutorial 8

CS 247

University of Waterloo

cs247@student.cs.uwaterloo.ca

June 23, 2017
Overview

1. OOP Principles
   - Open Closed Principle
   - Favour Composition Over Inheritance
   - Single Responsibility Principle

2. P1 Help
Object Oriented Programming Principles

The principles covered in class were:

- Open Closed Principle
- Favour Composition over Inheritance
- Single Responsibility Principle
- Liskov Substitutability Principle
- Law of Demeter
What’s open closed principle?
Open Closed Principle

A module should be open for extension but closed to modification.

You provide an abstract base class which gives the interface for the client to interact with. You then still have the freedom to extend the functionality through concrete classes without breaking client code.

Also known as ”Program to an Interface, not an Implementation”.
Problem:
1. We only have one class of shape (Square) at first, we want to calculate
the sum of a vector of squares.
2. We add a new class Circle derived from Shape. How can we modify
AreaCalculator::Area?
3. How to modify the code to follow open-closed principle?
Why?
We choose composition over inheritance because it is possible to modify the component at run-time while we are able to use it (whereas you can’t change your parent class at runtime.)
Problem:
A set of furniture: chair, bed and table in different colors: red, green and yellow.
1. If using inheritance,...
2. If using composition,...
3. If we add a new color now, ...
4. If we add a new type now, ...
When do you still need inheritance?

- We need a type hierarchy
- When using Polymorphism
What’s Single Responsibility Principle?
Each changeable design decision should be encapsulated in a separate module.
Single Responsibility Principle

Why:
- Code changes
- Low coupling
- Maintainability
- Testability and Debugging
Question:
1. How many responsibilities does the Camera class have?
2. How to modify it to follow the SRP?
End