COMP 3111
SOFTWARE ENGINEERING

LECTURE 18
SYSTEM ANALYSIS AND DESIGN
DESIGN PATTERNS EXERCISE

EXERCISE 1: PUT ON YOUR THINKING CAP

Which of the following are disadvantages of using *inheritance* to provide Duck behaviour? (Choose all that apply.)

☐ A. Code is duplicated across subclasses.

☑ B. Runtime behaviour changes are difficult.

E.g., we cannot change the flying behaviour of a duck at runtime.

☐ C. We can't make ducks dance.

☑ D. Hard to gain knowledge of all duck behaviours.

E.g., we need to look at all the duck subclasses to determine all the different quacking behaviours.

☐ E. Ducks can't fly and quack at the same time.

☑ F. Changes can unintentionally affect other ducks.

E.g., adding flying behaviour to the superclass allowed all ducks to fly, even those that should not be able to fly.

EXERCISE 2: DESIGN PRINCIPLE CHALLENGE

How does the observer pattern use the following principles?

Design Principle

Identify the aspects of your application that vary and separate them from what stays the same.

The number and types of observers vary; observers are separated from the state of the subject.

Design Principle

Program to an interface, not an implementation.

Subjects and observers know each other via interfaces.

Design Principle

Favour composition over inheritance.

Observers are composed with their subject; no inheritance is used.