Semester 1: Tutorial 3

The purpose of this week's tutorial is to test your understanding of inheritance in java, in particular:

1. Design of classes using an appropriate hierarchy
2. Overriding methods

Try thinking about the solutions in small groups.

1. Consider a library which wishes to store information about all the items it holds in stock. It only stocks books, journals and newspapers. You have been asked to write classes to store this information using an appropriate inheritance hierarchy.

   a) For each of the classes, identity the attributes (fields) that each class should contain – think about what properties you would look for in the library catalogue or on a book buying website for each class type.

   b) Are there any attributes which will are common across all three classes?

   Any fields shared by all the classes can be collected together into a generic `Item` class.

   c) Sketch an inheritance hierarchy for the `Item`, `Book`, `Journal` and `Newspaper` classes.

   d) Use your answer to (b) to write the constructor for `Item`.

   e) Write a `toString()` method for the `Item` class – design it so that it can be used by all of the subclasses of `Item`.

   f) Write the header (the first line) for the class `Book`.

   g) What attributes will the `Book` class have in addition to the attributes inherited from `Item`. Write the constructor method for `Book`.

   h) Write a `toString()` method for the `Book` class – design it so that it can be make use of the `toString()` of `Item`.

   i) Both `Journal` and `Newspaper` items are published at regular intervals e.g. daily or quarterly – could you make use of this to introduce a new class to your inheritance hierarchy as a superclass of `Journal` and `Newspaper`. What fields might this new class have?

2. Consider the person who wants to keep a record of their digital media. They own DVDs and CDs. The attributes that they wish to store include:

Title
Age certificate
Number of tracks
Running time
Lead Actor
Genre
Artist
Director
MoreThan1Disc

a) Identity the attributes that each disc might have in common and use this to define a class called `Disc`.

b) Write a constructor and a `toString()` method for `Disc`.

c) What attributes will the `DVD` class have in addition to the attributes inherited from `Disc`. Write the constructor method for `DVD`.

d) Write a `toString()` method for the `DVD` class – design it so that it can be make use of the `toString()` of `Disc`.

e) Carry out the steps of (c) and (d) for the class `CD`.

f) Where would a `BlueRayDisc` class fit into the class hierarchy? What extra attributes would a `BlueRayDisc` require? Write a `toString()` method for the class `BlueRayDisc`.

g) How could you adapt your code to identify both physical discs and electronic media?