University of HUDDERSFIELD

CFS2160: Programming Stream

Tutorial/Practical 15

Using Inheritance

Introduction

There is no Log Book this term.

The exercises in the lab will gradually build up a library of code that you will be able to draw on when you start work on the second assignment.

Activities

Make sure your work so far this term is up to date. If you need to ask about anything, do so.

Last week you developed a collection of classes that could have been used in some banking application. The spec was as follows:

1. A `CurrentAccount` class that has a balance, allows deposits and withdrawals, may allow an overdraft, but does not pay any interest. (You probably have most of this class already.)
2. A `DepositAccount` that is the same as a `CurrentAccount`, but which pays interest and which cannot go overdrawn.
3. A `StudentAccount` that has a fixed £500 overdraft.
4. A `YoungSaversAccount` that is the same as a `DepositAccount` but cannot have a balance of more than £100 (unless this happens because of interest being added).

You should be able to test each class by running a simple `main` method in each (or you might have developed a separate class to do this, following the model used to test the social media network from the lecture).

Now refactor your solution to use inheritance. It is probably best to do this in a new package so that you keep the original, but be careful not to mix up package names.

If you finish early, implement a class for the Bank itself - call it `Bank` - that contains a collection of bank accounts, and can display the sum of all the balances of these accounts.