

# FIT2081

## Laboratory for Week 06

### Relates to Lecture 05

This Lab is one of 11 you will attend during the semester. Each lab is worth 4 of your final marks for the unit. The best 10 marks will be used.

In the last 30 minutes of the each lab (or before if you ask and the tutor has time) your tutor will mark you. You will be asked a few questions to determine whether you **confidently** understand the work you are presenting. You will be awarded a mark of 0 if **any** trace of plagiarism is detected during this questioning. You may prepare your solution before the lab in which case you can ask to be marked at the earliest opportunity.

Purpose:

- Check your understanding of inheritance and interfaces

Important

- Each question should be answered in its own class in its own project or within a common project. Please note the compiler compiles all classes in a project at once (this can be changed but not recommended for beginners)

Download from Moodle the zip file “Week 5 Code”. Unzip it. It’s an IntelliJ project containing the code you are required to enhance and modify in this laboratory.

Modify the downloaded code according to the following specification.

### Q1 – Modifying InheritancePlus (1 mark)

Code another subclass of Employee called PieceWorkEmployee. Use the existing subclasses as a guide. This class represents employees who are paid according to what they produce each week (so called piece-work employees). Each has a target number of pieces to make each week and a payment per piece. If they exceed their target the payment per piece has a 10% bonus for all pieces made in excess of the target. If they do not make their target they are fined a set penalty amount. Each piece-work employee has their own piecesMade, target, paymentPerPiece and penaltyAmount values.

### Q2 – Modifying InheritancePlus (1 mark)

Modify the driver class (PayrollSystemTest.java) to exercise/test a piece-work employee the same as the other employee types.

**Q3 – Modifying InterfacesPlus (1 mark)**

Add the newly coded PieceWorkEmployee class to the project. Fix the compile error this causes.

Modify the driver class (PayableInterfaceTestTest.java) to exercise/test a piece-work employee the same as the other employee types.

**Q4 – Modifying InterfacesPlus (1 mark)**

Add another instance variable to the Invoice class called invoiceID of type String. Modify the class to accommodate this new instance variable.

Add another method to the Payable interface called getID(). It returns a String that identifies the object the method is invoked on. Now fix all the compile errors this creates USING THE ABSOLUTE MINIMUM AMOUNT OF CODE (THINK CAREFULLY).

Now modify the driver class (PayableInterfaceTest.java) to include invocation of this new interface method. Also include a piece-work employee in the driver's processing.