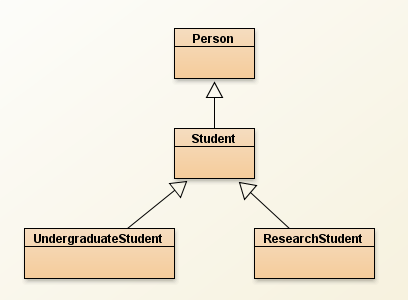
**KV4001: Week 4 Lab Exercises**

**Introduction**

The object of this lab is to further practice the concepts of inheritance; in particular subtyping and the uses of ‘super’.

The code you will write represents students taking undergraduate and research courses (Note: Not at Northumbria!).



**Person** is a class representing a person with a first and last name. It has deliberately been left as simple as possible.

**Student** represents a ‘general’ student. It should extend Person and adds course and id attributes together with their required accessor and mutator methods. These attributes should be initialised via the constructors. It also holds variables to hold the overall marks for each of years 1 – 3 which should be set to zero (0) by the constructors, and later given values via their mutator methods.

The class also holds methods to calculate the total mark over the three years and the overall result mark (the average of the 3 marks) which is the general starting point for each type of course.

**UndergraduateStudent** Is a subclass of Student which adds an alternate calculation of the overall result based upon the average of year 2 & 3 marks, if the result would be higher. The method should use ‘super’ to discriminate between the super/subclass versions of the getResultMark() method.

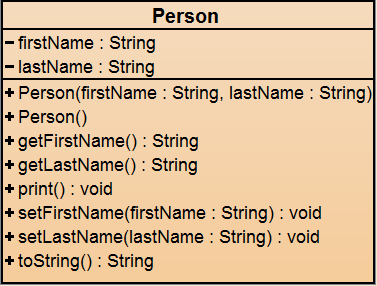
**ResearchStudent** is a subclass of Student where a student (a) has a named supervisor and (b) a research mark – both of these should be initialised to a default value in the constructors of the class. Both attributes require accessors and mutators.

A research student’s result mark is calculated on the basis of 80% of the year1 – 3 average (Student has a method to get the three year total which you should utilise) and 20% of the research mark.

**General** All classes should have a toString() method to return the contents of an object of that type. They should use ‘super’ where appropriate to take advantage of their superclass’ version of the method. They should also have a print method to write these values to the console window in the same format as the equivalent toString() method.

**Task 1**

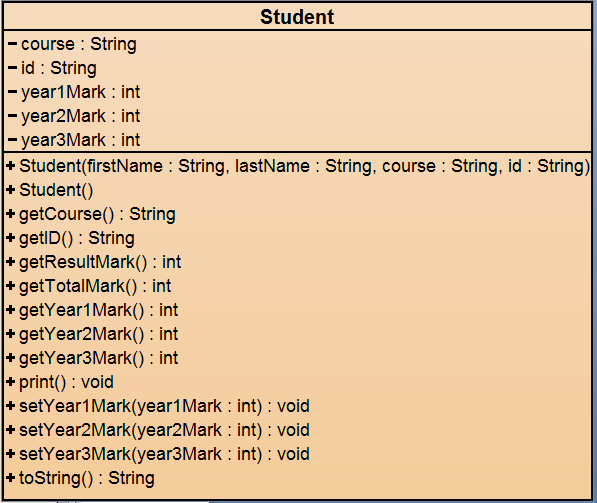
Open a new project in BlueJ. Add a new class Person and complete it as per the following class diagram. Fully comment your code as you go.



Compile the class, fix any errors. Create one or more objects and fully test the methods.

**Task 2**

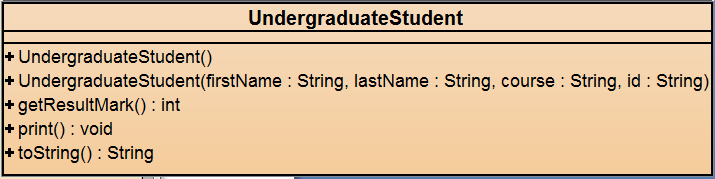
Add a new class Student and complete it as per the following class diagram. The class should have Person as a superclass.



Compile the class, fix any errors. Create one or more objects and fully test the methods. Check it against the notes on page 1. Does it perform as it should? If not amend your code and retest.

**Task 3**

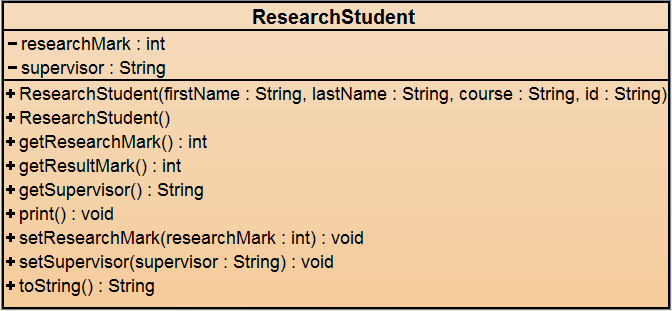
Add a new class UndergraduateStudent and complete it as per the following class diagram. The class should have Student as a superclass. This class uses the variables already defined by its superclass and does not add any others. Its purpose is to redefine how the result is calculated as per the note on page 1.



Compile the class, fix any errors. Create one or more objects and fully test the methods. Check it against the notes on page 1. Does it perform as it should? If not amend your code and retest.

**Task 4**

Add a new class ResearchStudent and complete it as per the following class diagram. The class should also have Student as a superclass. This class adds variables for researchMark and supervisor as well as the variables already defined by its superclass. Its purpose is to add the extra functionality required for research students and also to redefine how the result is calculated as per the note on page 1.



Compile the class, fix any errors. Create one or more objects and fully test the methods. Check it against the notes on page 1. Does it perform as it should? If not amend your code and retest.

**Task 5**

Finally retest your application to ensure that everything works as expected. Think about the code you have written. What would you want to change if you had written the specification?