ECS414U/A – Object Oriented Programming
Week 9 lab session (5 marks)

Rules & instructions: This assignment is individual work. You must complete it by yourself. You can discuss it with your demonstrator and other students in your lab session, and you are in fact encouraged to do so. However, you must not share your solution with other students, so that they can submit it as their own, or request that others share their solution with you. This would result in disciplinary action.

This exercise is about writing code which uses Java's collection type `ArrayList<E>`. You can use the built-in methods that Java provides for `ArrayList` to implement the methods asked for..

You must submit your files to QM+ before 23:59 Thursday, 7 April.

R1) (1 mark) Write a static method which takes an `ArrayList` of `String`s and an integer and changes the `ArrayList` destructively to remove all `String`s whose length is less than the integer argument. So, if the integer argument is 4 and the `ArrayList` is:

tomato	cheese	chips	fruit	pie	butter	tea	buns

the `ArrayList` should be changed to:

tomato	cheese	chips	fruit	butter	buns

R2) (1 mark) Write a static method which performs the same operation as the one in R1, but constructively rather than destructively. Show, using a front-end which gives aliasing, that your method for R1 is destructive and your method for R2 is not.

R3) (1 mark) Write generic static methods which take an `ArrayList` and two objects of its element type. The methods must replace the first occurrence of an object equal to the first object by the second object. As with parts R1 and R2, give a destructive and a constructive version. For example, if a call takes an `ArrayList` of integers `[5,12,4,16,4,2,2]` and the integers `4` and `7`, one method should change the actual `ArrayList` object to `[5,12,7,16,4,2,2]` and the other should return a new `ArrayList` which has that value.

Note the significance of the words "occurrence of an object equal to". It means equality testing using the method `equals`, as opposed to testing for an occurrence of the actual object using the alias test given by the == symbol.

R4) (2 marks) Write a class which stores an `ArrayList` of `Account` objects where `Account` is as was asked for in Lab Exercise 2, you may use the class `Account` which has now been given as its solution. Give the class that stores the `ArrayList` of `Account` objects the following the methods that can be called on it, one for each of these tasks:

- Add a new `Account` object to the list
- Returns the `Account` object in the list that has the highest balance
- Returns the total of adding the balances of all the `Account` objects in the list
- Take an integer that is a value, deposit it to the `Account` in the list that has the lowest balance, and return that `Account`
- Take an integer that represents a unique identifier, and return the `Account` that has that identifier, or throw an exception if there is no `Account` in the list that has that identifier
- Take an integer that represents a unique identifier, and remove from the list the `Account` that has that identifier, or throw an exception if there is no `Account` that has that identifier