

## CS4013 Lab 3

1. Implement the solution to Tutorial 2, Q2. It has been decided that `TestResult` should be able to hold scores for any number of tests. You may add an additional constructor to `TestResult` which takes an `int` as parameter, representing the number of tests a `TestResult` must hold. The test scores could be stored as an array of double values but given that the data will be hidden inside the class if you can identify another suitable data structure to store the data you can use that data structure. However, such a change should not affect the class contract. The no-arg constructor could assume that there will be three test scores.
2. Design a `StudentResult` class which has data fields to store a student ID as a `String` and a `TestResult` object.
  - Add a constructor with a parameter representing the studentID.
  - Add a constructor with parameters representing the studentID and the number of test scores.
  - Add an accessor method for the student ID.
  - Add a method `addResult(int testId, double value)` to add a test result for a particular student.
  - Add the method `String toString()` which returns the student ID, each of the test scores and the grade as a comma separated string (As part of the solution you could add the `String toString()` method to the `TestResult` class).
3. Design a `GradeBook` class which has a `String` data field representing the name of the `GradeBook`, an `int` representing the number of tests associated with this `GradeBook` and which contains a list of `StudentResults`.
  - Add a constructor to the class which takes a parameter of type `String` representing the name and an `int` value as a parameter which specifies the number of tests that are associated with this `GradeBook`.
  - Add a method `void addStudentResult(String id, int testId, double value)`. If the student id has already been added to the grade book then insert the test result for this student. Otherwise add this student id and result to the grade book.
  - Add a method `String toString()` which returns a `String` containing the student results of all students separated by an end of line character.
4. Design a `TestGradeBook` class which creates 2 `GradeBook` objects and adds results associated with at least two students. Print the `GradeBooks` to standard output.

### Submission to Sulis:

Please submit solution for Lab 3 to Sulis in a compressed file with:

`TestResult.java`

StudentResult.java

GradeBook.java

TestGradeBook.java

Please ensure that each java class is tested independently.