

# Computing Science and Mathematics

## CSCU9T4 Practical (29<sup>th</sup> January)

Spring 2018

Building on the training record program of last week, this week you need to enhance the program to handle entries for someone who is training triathletes. That is, they will be running, cycling and swimming. This will involve building an object model of the new requirements and then implementing that to replace the class Entry. As last week, your extensions should be coded as elegantly and efficiently as possible.

The trainer looks after athletes training for the triathlon. This means that in addition to training runs (as recorded last week, with times and distance), there are now other training sessions to record: sprints, long and short distance cycles, swims. For example:

- a sprints session might consist of 4 \* 40m sprints (with 3 minutes recovery between), or 4 \* 300m sprints (with 6 minutes recovery between),
- a cycle session might include information on distance, terrain, time and tempo.
- a swim session should record similar information to that on a run (distance, time), but you might also want to know if it was in the pool or outdoors.

You are reminded that while it is OK to discuss general points with your colleagues, copying is a serious offence. If there is any suggestion of sharing of code between class members then you will be asked to submit your code and it will be checked for plagiarism.

You will be adding to this code next week.

### Checkpoint

**Demonstrate to a tutor your modified TrainingRecord program. It should include new buttons to add the new entry types (or some way of distinguishing them. The existing buttons (FindAllByDate and LookUp) should still work, and you should ensure suitable display of training entries.**

**Test your program thoroughly before calling the tutor: add some training records and make sure you get the right results for cases where zero entries are returned, where one entry is returned, and where multiple entries are returned. Also ensure that you display suitable output for different types of entries, especially when they are mixed together.**

**If you don't finish this work today, finish it off in your own time and get it checked at the next class.**

**THE ABSOLUTE DEADLINE FOR CHECKING THIS WORK IS  
THE LAB ON MONDAY 12th FEBRUARY 2018.**

P.T.O

# Computing Science and Mathematics

1. Create a suitable object model for this information by drawing a class diagram for the superclass Entry and its subclasses. What information (and therefore attributes) do you need for each type of session? Remember that you might use a text field, e.g. to store all the exercises in the sprints session. Which attributes will be part of Entry and which part of a subclass? What about operations on the (sub)classes? Are any of those operations in the superclass abstract methods?

*Note: since this is quite a simple setup, you could just use Entry, with an extra text field to code for type of activity. Don't do that: the point here is to exercise your use of inheritance.*

2. Implement these classes. You need to replace Entry by the new superclass and add new subclasses. Consider carefully the use of *redefinition* of methods to make preparing output very straightforward for each class.
3. Amend the GUI to handle the new entry types. How will you add new entries? Do you need three buttons or is there another way to deal with this? Think about your reasons for making the choice the way you have. What are the design considerations? The practical considerations? How will you justify your choice?
4. (Optional) Add a button to find all entries for a given name. How are you going to handle string matching? Do you require an exact match? A case-independent match? A partial match?