

CSC8001 “Programming and Data Structures”

Coursework Part 1

This work is due for electronic submission through NESS by **4pm on Tuesday 14th November 2017**. It represents an individual assignment (no group work) and can be carried out during independent study and during weekly practical classes.

This is the first half of the module’s coursework, which constitutes 15% of the final mark (overall coursework contributes 30% of the total assessment). The coursework is marked out of 40.

1. Objectives

The objectives of this coursework are for you to demonstrate that you can:

- Define classes, fields, constructors, and methods in Java
- Use appropriate types, including collections
- Implement basic algorithms using collections
- Write a main method including console I/O
- Devise appropriate test cases
- Document your solution with javadoc

Note that at the time of setting the coursework, you will not have been taught all the material relevant to this exercise (for example, main method, console I/O, testing). You may need to research relevant Java Class libraries in order to get help with certain features.

2. Scenario: Misty Mountains

Mountain climbers like to keep collections of the mountains they have ascended, so that they can compare achievements with other climbers. In order to help them to do this, you are asked to write a java program to track details of mountains climbed, and allow members of a climbing club to compare their achievements.

3. Problem specification and Mark Scheme

Define a Java class **Mountain** with appropriate fields, methods and constructor to store and retrieve information about the name and height (in metres) of the mountain. [4 marks]

Define a Java class **Climber** with appropriate fields, methods and constructor to store and retrieve the name of the climber, their age and gender, and a list of Mountains that they have climbed. **Note that different climbers may record details for the same mountain: these would be stored as separate objects. There is no need to record a master list of all mountains.** Include methods to return:

- The highest mountain recorded by this climber.
- The average height mountain recorded by this climber.
- A list of all mountains recorded by the climber with a height greater than a given level. [8 marks]

Define a Java class **Club**, which holds information about all climbers. Include methods to return:

- The climber who has recorded the highest **average** mountain height.
- The highest mountain recorded by a member of the club.
- A list of all mountains that have been recorded, that are higher than a given level.

[8 marks]

Finally, define a class **ClubStats**, with a main method which does the following:

- Presents the user with a menu (printed to the console) of features:
 - add a new climber to the club, given their name, age and gender;
 - add details of a mountain (name, height) to the record for a given climber; **different climbers may record details for the same mountain: these would be stored as separate objects. There is no need to record a master list of all mountains.**
 - provide statistics: the climber who has recorded the highest average mountain height; the highest mountain recorded by a club member; and all mountains recorded with a height greater than a given level;
 - exit
- Takes user input from the console to choose one of the menu features
- Allows the user to input, via the console, the details of climbers and mountains they have recorded
- After executing one of the features, returns the user to the menu to choose another option

[8 marks]

All classes should be properly documented with javadoc [2 marks] and include appropriate testing methods [5 marks].

4. What to submit

You must submit 2 files to Ness, as follows:

- A zip file named 'CSC8001_coursework_pt1_2017_FirstnameLastname.zip', where 'Firstname' is replaced with your first name, and 'Lastname' is replaced with your last name. The zip file must contain:
 - The source code of your Java classes.
 - The generated javadoc for your classes.
- A PDF document named 'CSC8001_coursework_pt1_2017_FirstnameLastname.pdf' following the same conventions as above, containing a reflective commentary on your implementation (approx.. 500 words), saying what went well and what you would do to improve or extend your program [5 marks].

5. Mark Scheme

35 marks are available for the code, testing and Javadoc. 5 marks are available for the reflective report.