

## Cardiff School of Computer Science and Informatics

### Coursework Assessment Pro-forma

**Module Code:** CM1210  
**Module Title:** Object Orientated Programming with Java  
**Lecturer:** Yuhua Li  
**Assessment Title:** Practical implementation of algorithms and data structures  
**Assessment Number:** 2  
**Date Set:** 22<sup>nd</sup> March 2019  
**Submission Date and Time:** 10<sup>th</sup> May 2019 at 9:30am.  
**Return Date:** 7<sup>th</sup> June 2019

This assignment is worth 50 % of the total marks available for this module. The penalty for late or non-submission is an award of zero marks.

Your submission must include the official Coursework Submission Cover sheet, which can be found here:

<https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf>

### Submission Instructions

Your solution must include:

- A filled-in copy of the official Coursework Submission Cover sheet
- A typeset PDF report (approximately 500 words/roughly 2 pages of text; diagrams, screenshots, charts etc. are encouraged) detailing the following:
  - An introduction to the report as a whole, specifying what you shall be presenting in the report
  - An overview of your program design and implementation (include code snippets for each sorting algorithm and each of the points in part C below, and briefly describe what each snippet does)
  - Results of the timing and move/swap count experiments for each sorting algorithm (as tables, graphs, etc. as appropriate)
  - A brief user guide describing how to run the sorting algorithms
  - How to run your code (provide this in a “README” file)
- All Java source files, and related files (e.g. dependent third-party libraries and graphics), submitted as a single zip file collection.
- You should include your student number on the report.

Ensure that your student number is included as a comment at the top of each Java file that makes up your submission.

Description		Type	Name
Cover sheet	<b>Compulsory</b>	One PDF (.pdf) file	[student number].pdf
<b>ONE</b> PDF file (and no more than one) which contains the report points listed above (“A type set PDF...”)	<b>Compulsory</b>	One PDF (.pdf) file	CM1210_CW2_[student number].pdf
<b>ONE</b> ZIP file (and no more than one) of all the source code written	<b>Compulsory</b>	One ZIP (.zip) archive	CM1210_CW2_[student number].zip

You should upload the above to Learning Central. Any deviation from the submission instructions above (including the number and types of files submitted) may result in a mark of zero for the assessment or question part.

---

## Assignment

### INSTRUCTIONS

- A) Stop words are high-frequency words that are removed from the representation of natural language data. Write a method `removeStopwords(GPT2, stopwords)` that removes stop words listed in the file **stopwords.txt** from the file **GPT2.txt** (both files are available on Learning Central) and **returns non stop words** in an *ArrayList*. We recommend that the data is stored in an *ArrayList* within the Java program. (This part should be achievable now.)
- B) Implement the two algorithms insertion sort and merge sort in order to sort the words obtained from above in alphabetical order (the pseudocode for these methods is available in the lecture notes). For each of the algorithms write a separate Java method e.g. *insertionSort(GPT2)* and *mergeSort(GPT2)*. For each of the sorting algorithms measure
- time that is needed to sort 100 of the words, 200 of the words, 300 of the words, etc. up to all words by each of the two algorithms
  - count the moves and swaps that occur while sorting elements
    - (This part should be achievable now. Before attempting this exercise you should work through the Algorithms lab exercises, available on Learning Central. The techniques used there will help you to work out how to approach this part of the coursework, in particular there are examples of how to time algorithms and count the moves and swaps.)
- C) You should create two methods for a data structure of circular array *Queue*. Your data structure should have the class name *MyArrayQueue*. The two methods that should be implemented are:
- Adding element to the queue:
 

```
public void enqueue(Object theElement) {...}
```
  - Deleting an element from the queue and return the deleted element:
 

```
public Object dequeue() {...}
```

In both methods handle error properly (e.g. what happens when adding an element to a full queue).

There will be a skeleton code of *MyArrayQueue* available on Learning central that contains: *MyArrayQueue* class with constructor, *isEmpty()*, *getFrontElement()*, *getRearElement()*, methods and signatures for two methods *enqueue(Object theElement)*, *dequeue()* that you should implement.

You can reuse any implemented methods in the skeleton code. You ARE NOT allowed to change any parts of the methods.

(This part should be achievable after the lectures on *Queues*.)

All code must be written by you, although you can use the lecture notes (and lab exercises), textbooks, and the Oracle Java website for guidance.

---

## Learning Outcomes Assessed

4. Implement basic data structures and algorithms
  5. Analyse and describe the performance of data-structures and algorithms
  6. Deliver technical reports on Object-Oriented systems
- 

## Criteria for assessment

Credit will be awarded against the following criteria.

- removeStopwords(GPT2, stopwords) method [5 marks]
- Insertion sort algorithm implementation [5 marks]
- Merge sort algorithm implementation [6 marks]
- enqueue implementation [8 marks]
- dequeue implementation [4 marks]
- Report introduction and description of key points of your implementation [7 marks]
- Results of timing experiments (for algorithms only) [10 marks]  
Results of move/swap experiments (for algorithms only)
- User guide for how to use each program (The code provided may be run to check it works properly, and will be manually inspected.);  
Readability / Interpretation of the results shown;  
Report conclusion / summary. [5 marks]

(Total: 50 marks)

Feedback on your performance will address these criteria.

The level of attainment against the appropriate award.

- 1st (70-100%)
- 2.1 (60-69%)
- 2.2 (50-59%)
- 3rd (40-49)
- Fail (0-39%)

## Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned on the 7<sup>th</sup> June 2019 via Grade Centre/Learning Central.

Feedback from this assignment might be useful for *CM2307 Object Orientation, Algorithms and Data Structures*.