

Assignment OO Programming Sem 2 2017

Java

This assignment is worth 60% of your Sem 2 CA (and therefore, 30% of your OVERALL CA mark. The aim of this assignment is to develop a java application that puts into practice the programming skills taught on your OO programming course.

You have three different options for your project:

- You can pick one of the ideas described in this document. You should read them anyway to give you an idea of the type and scale of assignment expected.
- You can create your own idea e.g. a game, some sort of analysis tool, a solution to a puzzle, etc.. your choice.
- You can do a team project, with up to 3 people, and pick your own idea as a group. This will need to be a bigger project, with each team member doing an identifiable, equal amount of the development.

What next?

If you are doing one of the 6 ideas below - just go ahead and get started.

If you are doing your OWN project idea, or a group idea, please submit a description of your idea to me by Friday March 10th. Your idea must have some sort of algorithm in it – just as the sample projects shown here do – so keep that in mind. You should divide it into the core functionality, and the optional advanced features. I will approve the project or send it for redrafting if it is not suitable. The sooner you send it to me, the sooner you get it returned.

Assignment criteria

- All or most of your assignment should be written in Java. If there are extra parts (e.g. database, or some sort of non-Java GUI) that is fine.
- Your code should demonstrate the use of OO concepts. - including using classes for separate entities (as opposed to dumping everything into a single file), methods, encapsulated attributes, constructors, inheritance, interfaces, polymorphism.
- Code should follow java naming standards, be well indented, bracket aligned, comment headers, comments.
- You need to include a link to a short video with your submission to explain how you have coded it, what classes you have used and so on, how your algorithm worked. *No more than 2 minutes.* (It should be of your actual screens with your voice - as opposed to of you!).
- Include a readme file to briefly explain your project.
- Use GIT to manage your source code. Your code and video link will be collected from a your central GITHUB repository (more on this). We will cover a topic on GIT.
- If you make use of any code from a book or online source, you must show this in the comments. You will be marked on code that is your own code.
- You will demo your code in the lab as part of the marking.

Assignment OO Programming Sem 2 2017
Java

- The assignment is officially due in on by last day of term before Easter holidays (Fri 7th April) but since it won't be corrected / demo'd till first week after Easter, it can be submitted anytime **by first day post Easter (Monday April 24th)**.

Marking

The marking scheme will be:

- 10% Project management (demonstrating regular commits using GIT over the lifetime of the assignment)
- 10% video explaining your assignment – available via YouTube.
- 40% Basic core functionality: for well functioning, well implemented, using code that follows coding standard *core* functionality – i.e. the core of whatever problem you are solving.
- 40% Optional advanced features: for added functionality that makes enhances whatever you idea is.

**Plagiarism means attempting to pass off someone else's work
as your own or deliberately allowing another student to copy your work**

**Be careful: PLAGIARISM WILL NOT BE TOLERATED AND WILL BE DEALT WITH
SEVERELY**

The following ideas are available to choose from if you wish

1. LanguageAnalyser

This tool will analyse the text (e.g. a sentence, a file of text tec) – and analyse it to decide whether it is

- Slang- style language: Think yourself of how this could be identified. e.g. Typical style is shorter sentences, shorter average word length, containing slang words or bad language?
- Formal business language: Long sentences, average longer word length, "clean" language.

Ideas for additional features that could be included to include marks would be things such as:

- Have an editable dictionary list so that slang words can be extended.
- Having a GUI to run things and display the results.
- Having the ability to select documents through a "File Chooser" GUI, with graphical results rather than just uploaded a file or sentences from the console.

Assignment OO Programming Sem 2 2017

Java

- Extending the idea of how to identify slang versus formal language.
- Making it multi lingual.

2. Topic Analyser

This tool will allow you to detect whether a set of documents are “about” the same topic or not.

The tool will analyse the words in each document – and decide what the most common words are in each document.

“Stop” words should be excluded from the analysis (this is the list of words that don’t really add meaning which are not included in the similarity measure – such as “the”, “a”, “of” “he”.. etc. Sets of stop words can be found online (for example at :

<http://www.lextek.com/manuals/onix/stopwords1.html>)

Basic on the overlap in the top X words (e.g. 10), a grade of likelihood of it being about the same topic can be produced (e.g. 70% overlap in the top ten words = 70% likely to be about the same topic or whatever way you decide to do this).

Additional features that could be included to include marks would be things such as:

- Having a GUI to run things and display the results.
- Having the ability to select documents through a “File Chooser” GUI, with graphical results rather than just console
- Allowing words which appear on the list as common across documents - but which are obviously not informative about the topic – to be added to the stop word list, and the analysis re-run.
- Saving down the results persistently to a file, with the overlapping words written out too.

3. My Search Engine

This tool will allow you to search for a term across a set of text sources.

The user puts in a search term and the tool will check the contents of a set of text files and tell you which ones contain the search term. The files that have the “strongest match” against the search term should be returned at the top of the list.

At its most basic, the user can put in a single word, and the matching is done on that.

Assignment OO Programming Sem 2 2017

Java

To make the search better, you can do things such as : ability to search on multiple words – e.g. Christmas day.. although what rules you apply as to whether these are assumed to be together or separate words is up to you

To make this more advanced, you can :

- Have more sophisticated searching - e.g. exact phrase matches, comma separate words, wild cards (such as walk* to find walked, walking, walk etc).
- Have a GUI to take the search request parameters and the search results.
- Have a way for the user to pick the search space(i.e. the text files to be searched).
- What else can you come up with?

4. Robo-Reader

This idea is similar to Topic Analyser – but it concentrates on one document only

This tool will do two things

- (1) Automatically detect if it is in English, Spanish or French
- (2) If it is in English, allow a user to automatically identify what a document is about. Imagine you have a text document and you need to automatically identify what it is about. As a human that's easy – just read it. For this tool, the answer will be the ten most common words from the document - in the hope that these convey the topic – it's up to the user to put meaning on the set of words when they see the answer.

For Part (1), the easiest way is to do a match of basic words that occur everywhere (e.g. the words for “a”, “the” .. in English/ French/ Spanish

For Part (2): The most basic version of the tool is that the user enters a file name at the command line, and the system returns the top ten words (with word counts) such as `Christmas (12), train(6)` etc.

So the words returned should be informative. E.g. if there is a document about Christmas activities in Dublin City Centre, the words from the document that might convey what this document is about are:

Christmas, Dublin, City, Centre, Activities, Shops, People, Tree, Lights etc

It certainly wouldn't help to have words such as “is”, “a”, “this” .. etc.

And nouns help too - so perhaps there is a way of identifying unhelpful words?

Advanced features might include things such as:

- A GUI to run this
- A GUI to display the results back
- The ability for the user to say what words to exclude from the topic finder.
- What else?

5. Abusive text content detector

Imagine a tool that is able to automatically detect abusive content on the internet, such as “bad” instagram posts or tweets

The way that you might do this as a set of rules might be

- Is there bad language in the content?
- Are they “shouting”? YOU ARE A ..
- What else? Maybe the user name is a bit strange?

This tool will analyse a text file containing sets of posts - and determine which ones are “abusive” and which are not abusive.

It’ll follow a set of rules - to be determined by the developer – that decide whether it’s Abusive or Safe content. The most basic level is matching for bad language words, and variations of bad words.

The user should be able to edit the list of bad words as these evolve over time. The output should indicate some sort of likelihood that it is abusive – based on some sort of rule that you think is useful (e.g. 1 bad word out of 100 words will have lower chance of being abusive than 3 bad words in a 2 sentence text).

Advanced features might include things such as:

- Editable list of words or characters that are typical in abusive content that the user can edit via a GUI
- More sophisticated rules for determining abusive content likelihood – e.g. “shouting”? what else?
- A GUI to run this
- A GUI to display the results back
- The ability to add particular rules
- What else?

6. Data Expert

Data analysis is a major area within Computer Science . Apart from the Big Data generated by social media and internet generally – there is an insatiable desire on behalf of companies to analyse their data in order to reveal “knowledge” hidden in the data. For example, an Optical chain analysed their sales data to determine that sales of high end high profit glasses peaked on Friday afternoons – so they made sure that they had enough staff to service this demand.

The Irish Government have put 1000s of datasets into public use at a portal site:

<https://data.gov.ie/data>

This has data about a whole plethora of public interests and government control information e.g. about crime rates, hospitals, schools, transport, environment, energy use and so on.

The purpose of this project is to take **ONE** of these datasets – and build a tool that shows interesting facts from the dataset. The dataset formats include Comma Separated (CSV) which is probably the easier to work at – so these datasets are at : https://data.gov.ie/data/search?res_format=CSV

You don't have to use the full dataset if it is too big. But to query it, you will need to LOAD the dataset, or a subset of it, into a relational database yourself – and get a connection working between your java code and the database (using JDBC).

You could also just read in the file and do simple operations on it (e.g. how many of ??).. but searching is very limited if you stick just to file format without a database.

Your project will need to have a GUI that allows query parameters to be put in.

Extras

- Ability to see the results through the GUI too.
- Flexible queries – not just one or two hard coded