Assignment OO Programming Sem 2 2021
Java

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 two options for your project:

- You can pick one of the ideas described in this document. This is the most typical option.
- You can create your own idea e.g. a game, some sort of analysis tool, etc of your choice – but you need to get your idea approved first, to check it's suitable.

This assignment is worth 60% of your Sem 2 CA (and therefore, 30% of your OVERALL CA mark.

What next?

If you are doing one of the 4 ideas below - just go ahead and get started.

If you are doing your OWN project idea, please submit a description of your idea to me by Friday March 19th. 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

As well as meeting the requirements of the application:

- All or most of your assignment should be written in Java. If there are extra parts (e.g. database) 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 class), methods, encapsulated attributes, constructors, inheritance, interfaces, polymorphism.
- Code should follow java naming standards, be well indented, bracket aligned , comment headers, comments.
- Use BitBucket or GIT to manage your source code. Your code and video link will be collected from your whatever code repository you use (more on this). If you haven't used GIT, we can cover in class.
- If you make use of any code directly from a book or online source, **you must show this in the comments.** You will be marked on code that is your own code.
- The assignment is due in on by end of day **Monday 19th April**

Assignment OO Programming Sem 2 2021
Java

What you submit:

- **Code** (either as zip file or repository link);
- **Video:** 3 minutes or less, in which you demo all of your app. Show each screen being used, explain what data you are entering or what you are clicking, any error checking etc. as if you were demo'ing in person. Just show your screen with your voice - as opposed to you!
- **Readme file**: Include a readme file to explain your project. This should list:
    - List of classes, with a description.
    - Description of the **core** functionality you included;
    - Description/list of the **optional** functionality you included.
    - Explain what you would add if you had more time.

Marking

The marking scheme will be:

- 10% Project management (demonstrating regular commits using GIT over the lifetime of the assignment)
- 10% video *(and read me file)* explaining your assignment – available via YouTube or any video hosting service
- 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 enhances whatever your 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: IF YOU COPY YOUR ASSIGNMENT, YOU GET ZERO PLUS & IT GOES ON YOUR RECORD

The following four ideas are available to choose from :

Assignment OO Programming Sem 2 2021
Java

1. Machine learning model, using Naïve Bayes

This assignment allows you to learn about machine learning, including building a predictive/classification model. The machine learning classifier you will use is Naïve Bayes.

The purpose of the assignment is to build a prediction tool that predicts whether a secondary school pupil will become an entrepreneur or not (i.e. own their own business) in the future. The model uses previous examples of secondary school students and whether they later became entrepreneurs or not – and has information such as whether the student studied "business subjects", had a part time job, had a guardian/ parent who owned their own business and student gender.

Using java, the assignment should "train" a model using the dataset provided. Don't hard code the classifier in anyway – make sure it is driven by the data dynamically in your code. Using a GUI, allow for student's details to be put entered in, and the prediction model will give the probability of whether they are likely to be an entpreneur in the future.

To make this work well , you will need to have a GUI to take the new input (student details) that you want to predict about.

- Have the application read in the training set and do the training dynamically (i.e. don't hard code the rules for the classifier).
- If extra data rows, termed "instances" are added to the dataset, your classifier should still work (in fact, it will be improved as a result of new data).
- Allow for model to train on some of the dataset (e.g. 70%) and test itself on the remainder of the dataset(30%) so that it knows how "good" it is (i.e. be able to evaluate itself).

Dataset Columns: (Full dataset will be put on Brightspace)
Features: Student gender,

Features					The label
Student gender	Parent/ guardian had own business	Has a part time job	Urban or rural address	Studies business subjects	Become an Entrepreneur
Female	Yes	No	Rural	No	Yes
Etc					

Assignment OO Programming Sem 2 2021
Java

2. Topic Modeller

This tool will allow a user 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). Likewise, documents with small overlap on most comment words (e.g. < 40%) are definitely different topics.

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

- Improving the GUI quality so that it looks better than a very basic GUI (e.g. layout, colour etc).
- 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.
- Displaying the results - so that the user can easily see how many, and what topics (i.e. what word groups) are returned.
- Saving down the results persistently to a file, with the overlapping words written out too.
- Extra features that you think will enhance the application.

Assignment OO Programming Sem 2 2021
Java

3. Data Explorer

Note: I will cover SQL database connection (briefly) in class.

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).

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
  Whatever else you decide might enhance the application.

Assignment OO Programming Sem 2 2021
Java

4. My Search Engine

This tool will allow you to search for a term across a set of text sources – e.g. groups of text files.

The user puts in a search term into a GUI 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.

The user should be able to search on a single word. But to make the search better, you should be able 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; Maybe you can use "*" and logic (e.g. this word AND this word OR this word) to make the search smarter.

The above is just a basic spec. To make this more advanced for more marks, you will need to:

- 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 way for the user to pick the search space (i.e. the text files to be searched).
- Have a good ranking mechanism so the strongest match is returned first – and a ranking metric (e.g. a %) is calculated and the user can see this.
- Spelling correction where it can correct wrong spelling of search terms
- What else can you come up with?

---------- END ----------