1. A tool to monitor/inspect multi-threaded (concurrent) Java systems

Writing correct Java multi-threaded code [1] is a hard problem, which introduces new challenges and common concurrency bug patterns (e.g. memory consistency errors [1]). The programmer has to ensure that all accesses to shared data are coordinated. The coordination is usually done with some sort of synchronisation, which in turn might lead to further problems (e.g. deadlocks, starvation and livelock [1]). The utilisation of supporting software monitoring/inspection tools is an important countermeasure for programmers to identify such concurrency issues in programs.

The aim of this project is to develop a tool that monitors/inspects multi-threaded (concurrent) Java systems. The project is open-ended in terms of the techniques that can be utilised, and a few indicative examples are:

- Techniques for static program analysis [2], e.g. software code quality metrics ([3]): [4] and [5] are examples of two popular metrics for non-concurrent systems; or

- Runtime system monitoring techniques [6], e.g. the native task managers [7] of Microsoft Windows and macOS are two popular examples; or

- A combination of the above.

The target users will be other programmers, software engineers, and testers. Requirements gathering will mainly involve researching relevant literature (including learning the basics of Java concurrency and multi-threaded programming) and existing tools in order to adapt the selected techniques for concurrent systems. GUI implementation will be a “must-have” requirement. Some students may choose to investigate turning their tool into a plug-in for a popular Java IDE, e.g. Eclipse, IntelliJ etc.

Strong Java programming skills are essential. Furthermore, the students must familiarise with the fundamentals of the Java concurrency package [1].

[1] https://docs.oracle.com/javase/tutorial/essential/concurrency/
[2] https://en.wikipedia.org/wiki/Static_program_analysis
[3] https://en.wikipedia.org/wiki/Software_metric
[4] https://en.wikipedia.org/wiki/Source_lines_of_code
[5] https://www.tutorialspoint.com/software_testing_dictionary/cyclomatic_complexity.htm
[6] https://en.wikipedia.org/wiki/System_monitor

[7] https://en.wikipedia.org/wiki/Task_manager

2. A tool to support independent/self-directed learning within a discipline

The aim of this project is to develop a tool that helps learners to familiarise with concepts within a discipline/field as part of their independent/self-directed learning. The project is open-ended in that you may choose the discipline/field of study. One example in the context of learning programming is data structure/algorithm visualisation [1]. Another popular tool is Duolingo in the context of learning foreign languages [2].

The end-product must include functionality on the following aspects [3]:

- Assess readiness to learn;

- Set learning goals;

- Engage in the learning process; and

- Evaluate learning.

The target users will be learners and instructors within the proposed discipline/field of study. Requirements gathering and evaluation must involve users from this target audience. GUI implementation will be a “must-have” requirement. You will be allowed to choose the programming language and the GUI development framework, which may involve learning a new technology.

[1] https://www.cs.usfca.edu/~galles/visualization/Algorithms.html

[2] https://www.duolingo.com/

[3] https://uwaterloo.ca/centre-for-teaching-excellence/teaching-resources/teaching-tips/tips-students/self-directed-learning/self-directed-learning-four-step-process

3. Multiple/cross platform development

The aim of this project is to develop a piece of software that runs on multiple platforms. The project is open-ended in that you may choose: i) the topic (***subject to approval by your supervisor***) you wish to work on, and ii) the platforms you wish to target.

A few indicative examples:

- Native mobile app development for Android and iOS. This would involve programming in Java or Kotlin [1] (if you wish to learn a new programming language) for Android, and in Swift [2] for iOS.

- Cross platform non-native mobile/web app development. This would involve programming in an appropriate framework, e.g. JavaScript-specific frameworks such as React Native [3] for mobile and/or React [4] for web.

- Mobile (native or non-native)/web and desktop development. For example, if you opt for native Android and Java, this would involve familiarising with JavaFX [5] for the desktop app version of your end-product.

Apart from the above, there will be many more combinations, which you can discuss with your supervisor. The minimum expectation is that you must learn and adopt **at least one** new development technology/framework in the context of your project. Requirements gathering and evaluation must involve users from your target audience. GUI implementation will be a “must-have” requirement.

[1] https://kotlinlang.org/
[2] https://developer.apple.com/swift/
[3] https://reactnative.dev/
[4] https://reactjs.org/
[5] https://openjfx.io/

4. Mobile phone activity monitor

Mobile phones include three-axis accelerometers. These accelerometers can be used to monitor movement of the phone and therefore infer movement of the owner of the phone. The goal of the project is to write a program to use mobile phone accelerometer data to monitor the activity of the owner. The application could then prompt the user if they have been too inactive during a particular period of time.

The application should provide a summary of activity levels across a period of several days. These data could be used to provide the user with a summary of their activity periods across a week or across a month. The design of the user interface and type of analysis that should be performed should be determined through a requirements capture exercise.

Mobile phone accelerometers can be monitored using the Flutter framework. Data can be stored in the NoSQL Hive database or the SQLite relational database.

5. News trends analyser

Cloud providers Microsoft Azure and Amazon Web Services (AWS) provide serverless computing as Lambda functions and Azure Functions, respectively. These services allow short computation processes to take place on demand. The computation can be triggered by events such as web requests and timers. As long as the number of requests is low enough, access to these services is free of charge.

The goal of the project is to build a function that is triggered by a web request. The function should pull data from other sources, analyse the results and return them back to the user. For example, the function could query Google news and request recent events on a current topic, analyse the content of the search results and provide a summary to the user interface. Headless Chromium can be used with Python and beautiful soup on the Azure Linux consumption plan, to allow a Python program to run queries, analyse the data and return them as JSON to a user interface. The user interface should allow the user to request news on a specific topic and be written as a single-page web interface, using either React, ReactNative of Flutter frameworks.

6. Social media dashboard

Cloud providers Microsoft Azure and Amazon Web Services (AWS) provide serverless computing as Lambda functions and Azure Functions, respectively. These services allow short computation processes to take place on demand. The computation can be triggered by events such as web requests and timers. As long as the number of requests is low enough, access to these services is free of charge.

The goal of the project is to write a function to analyse Twitter or other social media data for specific search words or perform a sentiment analysis. It should be implemented in the Python programming language and display results using a dashboard. The dashboard should present a summary of the results, in a manner that potential users may find useful. For example, users may be following particular sports events or current affairs. The search words and sentiment analysis can be combined to provide more information to the user.

7. Genetic algorithm optimisation

Genetic algorithms can be used to optimise parameter settings. A problem is expressed as a function of many input parameters, where the function returns a value. The genetic algorithm can be used to find the global maximum or minimum within a bounded multidimensional space. The algorithm is configured to have a starting number of solutions and use evolution, including elimination of the unfit, reproduction and mutation.

The goal of the project is to write an optimisation program. This could optimise university room use by classes or a another physical system where the optimal order is not easy to solved manually. The program should allow a used to describe the starting parameters, including the genetic algorithm settings, run the genetic algorithm and provide the user with a summary of the results. The software should be written as a desktop application, using Python or C# (.Net). The genetic algorithm will be provided to students.

8. Mobile application to provide location based information

This project aims to develop a mobile application which provides information to a user based on their location. To do this you will find a data set on the topic of your choice, such data could be obtained from the Scottish government or the office for national statistics. This data must have sufficiently granular location information to be useful (e.g. council areas, electoral wards).

The front end of the application will involve mobile app development and should present information in a clear way. The back end will need to be able to retrieve and manage information in a robust way. You will be able to choose the frameworks you use for this project, for example, an Android/Java front end supported by an SQL back end. However, care and consideration should be given to the technologies chosen.

This project is open ended and you could extend your application in many ways. Depending on your data set you may be able to provide good data visualisation. It may also be possible to include multiple data sets to provide more comprehensive information. Though there are many options at its core this topic should produce a working application which provides location specific information.

9. Demonstrator timetabling tool

Every semester the department must allocate demonstrators to classes. For this project you will develop a tool which helps organise this process. It should be able to save the information pertinent to classes - class times, required skills, number of demonstrators and the number of teaching hours. As well as the class information, the tool should save the information of demonstrators- their timetables, knowledge and abilities and preferred classes.

Once all class and demonstrator information has been acquired your tool should then produce an allocation of demonstrators to classes. This could be done by naive allocation or by constraint solving. It would also be possible to analyse the allocations your tool produces to determine if it is optimal.

The tool could be developed to handle more complex constraints once the initial requirements are met such as multiple departments or a system for demonstrator experience. It would also be possible to generalise this tool to a generic timetable constraint solver.

Target users would be lab demonstrators and class lecturers. As the tool would have many users it will be necessary to build a GUI for ease of use. You will be allowed to choose the language you use and GUI framework.