# Assignment 2

## 1) Introduction

You will be required to write some Java code that solves specific tasks, and to give a brief written report on that code.

The assignment allows you to demonstrate to what degree you have been able to achieve the following Learning Outcomes: 1-3. Its worth towards the final mark is 55%.

**Please read every section of this document very carefully before starting on your work. Apart from specific the tasks and questions, it also contains information on how your submission will be marked, very important information about Academic Integrity and instructions on how to submit. Not following the instructions will affect your grade.**

### i) Marking Scheme

The assessment will be marked against the university Common Marking Scheme (CMS).

Your mark/grade will be based on how well your submission addresses all the tasks listed in Section 2 below and their combined submitted quality. The tasks are not weighted as such, but they all contribute to assessing your attained achievement towards the Learning Outcomes. Furthermore, good programming style, consistency, legibility and tidiness of program layout will also contribute towards your mark.

Here is a summary of what you need to achieve to gain a grade in the major grade bands:

| Grade | Requirement |
|---|---|
| Fail | Your submission only addressed a small portion of the tasks assigned and entirely skipped the other(s). Does not securely demonstrate the intended learning outcomes. |
| 3rd | You submit sufficient working code that is legible to show that you have mastered the basics of the coding tasks and written text that shows an effort to understand some of the basics of OO design and describes the code written but with little justification of choices. |
| 2:2 | Your code is mostly correct and legible, you have successfully completed most, but not all of the coding tasks, and your written documentation shows variable depth of understanding of the tasks. |
| 2:1 | Your code is working, correct, and tidy (legible), you have successfully completed all of the coding tasks, and your documentation shows appreciable depth of understanding of the tasks by using reasonable justifications and explanations. |
| 1st | You code is working, correct, and very tidy (legible) and you have produced a creative solution. Your documentation is justified, well thought-out, and shows great understanding of the task and underlying assumptions. |

Note that in the overall scheme, the documentation/report weighs equally to the programming task and these descriptions above assume equal effort has been put into both aspects of the assignment. If your achievements in different parts of your submission matches different grade descriptions above, your final grade for the assignment will be the average of the estimated grade category of all the parts as deemed by the academic judgement of the marker.

The full details of the CMS can be found here

https://www.stir.ac.uk/about/professional-services/student-academic-and-corporate-services/academic-registry/academic-policy-and-practice/quality-handbook/assessment-policy-and-procedure/appendix-1-undergraduate-common-marking-scheme/

## ii) Submission

Submit your work on Canvas as per the relevant instructions, details on what your submission should contain can be found at end of this document.

## iii) Academic Integrity (Read this section carefully)

This is an individual assignment, and so all submitted work must be fully your own work. You are free to look up any online resources during your work on the assignment, however any attempt to communicate details of your solutions or approaches with others, whether other students or people outwith the University, is not acceptable.

The University of Stirling is committed to protecting the quality and standards of its awards. Consequently, the University seeks to promote and nurture academic integrity, support staff academic integrity, and support students to understand and develop good academic skills that facilitate academic integrity.

In addition, the University deals decisively with all forms of Academic Misconduct.

Where a student does not act with academic integrity, their work or behaviour may demonstrate Poor Academic Practice or it may represent Academic Misconduct, and the consequences for the student can be severe.

### Poor Academic Practice

Poor Academic Practice is defined as: "The submission of any type of assessment with a lack of referencing or inadequate referencing which does not effectively acknowledge the origin of words, ideas, images, tables, diagrams, maps, code, sound and any other sources used in the assessment."

### Academic Misconduct

Academic Misconduct is defined as: *"any act or attempted act that does not demonstrate academic integrity and that may result in creating an unfair academic advantage for you or another person, or an academic disadvantage for any other member or member of the academic community."*

The University recognises, amongst others, the following forms of academic misconduct:

- **Plagiarism:** *"a specific form of cheating which usually occurs when a student is working independently on an assessment (e.g. essays, reports, presentations or dissertations). Examples of other people’s ‘work’ can include anything taken from* *any form of publications, internet sources, the spoken word, graphics, data and written text. "*

- **Self-plagiarism:** where *"Duplicate submission of an item of written work in any other circumstance is not allowed and constitutes academic misconduct."* unless expressly allowed/required.
- **Inappropriate use of proof-reading:** *"Students cannot ask other people to write their work for them and should not use software (e.g. 'spinning' web sites, re-wording web sites or translation software) to generate text for them."*
- **Collusion:** *"When a student copies the work of another student either with or without the knowledge of the original author; or when two or more students work together to produce individual assessments."* This also applies to those that share their work with other students, even if they then do not copy the work and regardless of the intent.
- **Contract Cheating:** *"takes place when a student submits work for assessment that was completed by a third-party either for payment or for free. It is a broad category that includes, but is not limited to, work bought from so-called essay mills, customised work commissioned from ghost writers, and selling or exchanging work for use by others. A further example would be a friend or family member completing an assessment for a student. Work in this category covers the whole spectrum of assessment types. Any form of contract cheating constitutes academic misconduct, often of the most serious form."* More details about contract cheating can be found here: Contract Cheating - Don't take the risk
- **Dishonest Practice:** *"includes a wide variety of activities that aim to obtain an unfair advantage through (note this is not an exhaustive list):*
    - *Making false declarations to Faculties, Academic Staff members, Boards of Examiners or Appeal Panels.*
    - *Attempts to circumvent the similarity checking programmes that the University uses.*
    - *Submitting documents which have been forged in any way.*
    - *Attempting to gain or gaining access to examination or class test papers prior to their release and/or sharing examination or class test papers prior to their release.*
    - *Deliberate avoidance or refusal to engage with the relevant ethics review and approval process."*

The University of Stirling's full policy on Academic Integrity can be found at:

https://www.stir.ac.uk/about/professional-services/student-academic-and-corporate-services/academic-registry/academic-policy-and-practice/quality-handbook/academic-integrity-policy-and-academic-misconduct-procedure/

## 2) Assignment Instructions

In this assignment you will design, write, and justify an object-oriented Java program that manages information for an architect firm. The program must be able to read data from a comma separated values (CSV) file, accept input from a very simple graphical user interface, and display selected information in a legible and tidy manner.

The premise: You have been hired as a consultant by a large architect firm, Jenga Design Ltd. They need a system to manage their portfolio of projects. They currently keep their records in a large CSV file that you have been provided with, but they want to have a management system implemented.

## i) Resources

Your starter package only contains 3 files:

- jenga_projects.csv – A file containing Jenga Design's records of their projects in a comma separated value format. You can open this in a spreadsheet software to inspect the column names and contents. (e.g., MS Excel, or Google Sheets)
- zzzzzzz_report.docx – a report template for the completion of tasks Y and Z below.
- Assignment_2_instructions.docx – This file

**Assume the data within the CSV file is correct.** The CSV file probably contains some discrepancies in terms of for example sizes, budgets, durations, start and end dates, etc. This is not a concern, and you can treat it as just odd project details and ignore. Your main concern is with the column headers and which columns contain data for which project types.

## ii) Tasks

To complete the assignment, you must properly address each of the tasks below. They are quite open-ended and give you opportunities to showcase ingenuity and creativity but remember that sometimes simpler is better and you must also be able to concisely justify your approaches.

A. **Design:** Inspect the CSV file and the requirements listed in task B below, record the following in your report:
    i. Draw an appropriate UML class diagram describing your recommended OO design of the information system. Include as much detail as you can and make sure to include attributes, methods, inheritance, etc.
    The design must include a showcase of inheritance with respect to the information management point of view (i.e., the data in the CSV file).
    ii. Very briefly (max 500 words) explain why you recommend this design for Jenga Design's system.

B. **Programming:** Implement your design from Task A in Java. Apart from adhering to your recommended design described in task A above, your system:
    i. Should **not** access any database or save anything to local storage.
    For your purposes, when the software stops, everything is forgotten.
    ii. Must be able to read the CSV file into memory and construct appropriate Java objects from each row.
    iii. Must be able to accept information through a simple GUI to both record a new project, and adjust the information of an existing one. Make sure you include proper fields and buttons, and that they function correctly.

iv. Must provide 3 distinct search options. You are free to choose search options you think will be useful to the user, for example including but not limited to: searching for project types, locations (area), cost or date range, project manager, and/or project status.
Whatever you choose, the results should be displayed in a text field such that they can be directly copied to reports for Jenga Design's board of directors (i.e., neatly structured text that includes useful information and not just a csv list or a table).

**C. Post-coding documentation:** Write a short description (max 1000 words) of the approach you took in task B, the programming. Specifically mention your choices for part iv. Do not describe the code line-by-line, but discuss the approach in general, justify a selected number of choices you made in your implementation and, if possible, mention alternatives and why these were rejected.

*Note: If you source references and citations to support any justifications and claims made in A and C then there is no requirement for a specific referencing style, although the IEEE referencing style is suggested. For B, if any of your code is inspired or adapted from external sources, make sure to reference those sources as well, this can be done as URIs in inline comments but should preferably be done properly in the document containing your answers to A and C where you can then describe how that source inspired you and was adapted into your approach.*

## iii) Prepare for submission

You will need to submit your work on Canvas as two files: a report, and your code.

- **Report** containing your answers to Tasks A and C:
  - Name as **your_student_number**.[doc, docx, pdf, odt, or rtf]
  - Should include a **Class_diagram** for Task A. This may be hand-drawn (must be legible!) or you can use software such as lucidchart or UMLet.
- **architect_system** – a zip file containing a folder with your source code for the Java system you have implemented
  - This must include all .java source files you wrote, and all .jar files that your system depends on (or a proper Maven or Gradle dependency configuration).
  - This need (should) not include compiled binary files of your code.
  - This can include unit tests of your choosing *(recommended but not required)*
  - Name the zip file: **YYYYYYY.zip** (replacing the Ys with your 7-digit student number)