

Advanced Programming

COSC 1295

Assignment

Assessment Type	Assessment Type: Individual assignment; no group work. Submit online via Canvas \rightarrow Assignments \rightarrow Assignment. Marks are awarded for meeting requirements as closely as possible according to assignment specifications and the supplied rubric. Clarifications/updates may be made via announcements/relevant discussion forums.	
Due Date	Due date: end of Week 12 (18 th October), Sunday 11:59pm. No Late submissions are allowed unless special consideration has been granted.	
Marks	Weighting: 30 marks out of 100 for assignment. In addition, 5 milestones (4 marks each) to be submitted progressively as 3-minute videos adding up to 20 marks.	

1. Overview

Assessment Type: Individual assignment; no group work. Submit online via Canvas \rightarrow Assignments \rightarrow Assignment. Marks are awarded for meeting requirements as closely as possible according to assignment specifications and the supplied rubric. Clarifications/updates may be made via announcements/relevant discussion forums.

NOTE: Carefully read this document. In addition, regularly follow the Canvas assignment discussion board for assignment related clarifications and discussion.

Motivation: This individual assignment titled **Project Team Formation** requires developing a software that allows members to be assigned to project teams taking into consideration specific constraints while attempting to meet objective criteria reflecting ethical considerations, aspirations of project owners and preferences of students. This assignment is based on actual workplace practices and is reflective of tools used for making management decisions meeting multiple criteria such as increasing profits and improving employer satisfaction while meeting mandatory legal requirements. To be useful for managers such tools helping to find near optimal solutions (through the use of appropriate algorithms) must be made easily accessible using custom designed graphical user interfaces and visualization.

Using Java as a vehicle, you are required to demonstrate your understanding of object-oriented design principles, design patterns, generics, graphical user interfaces, unit testing, object relational mapping, data structures and algorithms, building on the foundations laid by the Programming Fundamentals course. This Project Team Formation assignment should incorporate appropriate visual elements and event-handling allowing managers to simulate the impact of their decisions visually (sensitivity analysis). It should also assist the manager in finding better solutions by using heuristic algorithms and appropriate data structures. To manage the complexity, the assignment is divided into five milestones (starting from week 3 and going until week 11) each carrying 4 marks and a final submission in week 12 carrying 30 marks. The initial milestones are very prescriptive to help you get started and make steady progress. The final milestone allows adequate room for creativity in interaction, development of fit for purpose algorithms and the use of appropriate design patterns to improve usability. Your assignment submitted will be marked based on functional features, qualitative aspects, reflective practices and explorative learning. During the final face to face assessment you will also be required to justify your design decisions as well as explain how your program can be extended to meet other related requirements.

Disclaimer: the specification in this assignment is intended to represent a simplified version of a system in real life and thus is not meant to be a 100% accurate simulation of any real system or service that is being used commercially.



2. Learning Outcomes

This assessment relates to all of the learning outcomes of the course which are:

- CL01: use the Java programming language in the implementation of small to medium sized application programs that illustrate professionally acceptable coding and performance standards.
- CL02: demonstrate knowledge of the basic principles of the object-oriented development process and apply this understanding to the analysis and design of solutions for small to medium scale problems.
- CLO3: describe and apply basic algorithms and data structures, in particular simple searching and sorting of data stored in data structures and manipulating data.
- CLO4: implement basic event-driven programming and graphical user interfaces in Java.

3. Assessment components and details

In this assignment and class progress demos you will be incrementally building a program to facilitate effective project team formation, considering the perspectives of project members (students) and owners (clients). The assignment is decomposed into 5 milestones which must be demonstrated to your lab supervisor (adding up to 10 marks). You will be required to demo your progress in the lab every two weeks starting from week 3 or 4, as shown in the table below. These milestones and the final assessment (30 marks) will be graded by your lab assistant who will be tracking your progress and giving you constructive feedback throughout the semester.

Туре	Learning Assessed	CLO's & Bloom's levels	Marks	Due-Date
Demo 1	Problem Solving with Collections, files	CLO 1,2,3,4 Blooms 4	4	End of week 3
	and exceptions			
Demo 2	JUnit Test cases, Generics	CLO 1,2,3,4 Blooms 4	4	End of week 5
Demo 3	Problem Solving with Java FX and Event Handling.	CLO 1,2,3,4 Blooms 4	4	End of week 7
Demo 4	Problem Solving incorporating relational database through JDBC	CLO 1,2,3,4 Blooms 4	4	End of week 9
Demo 5	Problem Solving incorporating heuristic algorithms, threads and patterns.	CLO 1,2,3,4 Blooms 4	4	End of week 11
Assignment	Problem solving with data structures, heuristic algorithms, OO design, Object relational mapping and MVC architecture.	CLO 1,2,3,4 Blooms 5	30	End of week 12



Milestone 1: Preparing the Data (Involves File Processing, Exception Handling and Input Validation) (4 Marks)

Write a menu-driven program that helps the project manager capture the details related to each project, project owner, company, student personal details and student preferences, in separate files. There should be an additional menu option to shortlist projects based on student preferences. Note, there can be multiple project owners from the same company and multiple projects by the same project owner. This program should perform appropriate error checking and input validation by validating all user inputs (Your program should not crash at any time because of invalid inputs).

Meni

- A. Add Company
- B. Add Project Owner
- C. Add Project
- D. Capture Student Personalities
- E. Add Student Preferences
- F. Shortlist Projects

A. Add Company

The details related to the company should include a unique company ID, company name, ABN number, company URL and address. The company details added should be saved to the file companies.txt.

B. Add Project Owner

The details captured by each project owner (contact person) should include first name, surname, a unique project owner ID (of the form Own1, ...), role (such as software engineer), email and the ID of the company the project owner represents (of the form C1, ...) The project owner details added should be saved to the file owners.txt.

C. Add Project

The details captured for each project should include a title, a unique project ID (of the form pr1, ...), brief description (one line), ID of the project owner (of the form Own1, ...) and the ranking of the skills sought after for the project (4 being the highest and 1 the lowest) in the 4 different technical specialization (Skills) at MRIT university shown in the table below. The project details added should be saved to the file projects.txt.

Technical Skill Categories		
(P) Programming & Software Engineering		
(N) Networking and Security		
(A) Analytics and Big Data		
(W) Web & Mobile applications		

Possible file format for the file projects.txt

Customer Relationship Management (CRM)
Pr5
CRM will be developed by a team of 4 using Yii framework
Own3
W 4 P 3 N 2 A 1

This project owner values Web experience the most and Analytics the least

D. Capture Student Personalities

Now assume the project manager can download a file named students.txt capturing the list of students doing the project course in current semester with their average grades (HD=4, DI=3, CR=2, PA=1, NN=0) in project related technical skills (P programming, N networking, A analytics, W web) with the format shown below. Assume students have IDs S1 to S20.

file format for students.txt (You need to create such a file with 20 students))

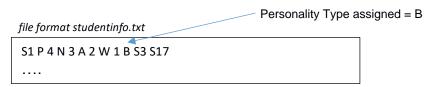
S1 P 4 N 3 A 2 W 1



The project manager interviews each of the students in the list during the first week and assigns their personality types as shown in the table below to ensure balanced teams are formed with at least one with leadership attributes in each team. It is required that the personality type assigned to students must be equally distributed across types.

Characteristics	Personality Type		
Likes to be a Leader (Director)	A		
Outgoing and maintains good relationships (Socializer)	В		
Detail oriented (Thinker)	С		
Less assertive (Supporter)	D		

In addition, the project manager allows each student to specify up to two others (student-numbers) they cannot work with in a team because of prior personal conflicts. Both the personality and conflict information must be captured and appended to existing file after the interview and saved in a text file studentinfo.txt, as shown below.



Note, all project must be of size 4. Exceptions must be thrown when the personality types of the 4 students assigned by the project manager is not balanced. For example, if there are 20 students, the maximum number of personality types A, B, C and D can only be 5, and hence any attempt to assign A,B,C, D type to more than 5 students should result in an exception.

E. Capture Student Preferences

Students having attended the presentations for 10 projects decide their preferences and submit them via discussion-board by the end of first week, with 4 for most preferred and 1 for the least. These preferences should be saved in the file preferences.txt. A student may add preferences multiple times, but only the final preference should be counted.

file-format for preferences.txt

```
S1
Pr6 4 Pr8 3 Pr6 2 Pr4 1
....
```

F Shortlist Projects based on Preferences

The student preferences for projects (from 4 down to 1) should be summed up for all the projects before discarding the 5 least popular projects in the file projects.txt.



Milestone 2: Forming Project Teams Manually and Creating the Test Cases (4 Marks)

In this milestone you are required to ensure teams are balanced in terms of personality types, each team has at least one leader type (A) and students with personality conflicts are not placed in the same team. In addition, for each team you are required to compute Team-Fitness Metrics which will allow project manager to swap members when necessary (next milestone) considering ethical considerations as well as preferences of owners and students.

Team-Fitness Metrics

- 1. Average student skill competency for each project team. This value should not vary greatly between teams to ensure fairness (ethical consideration).
- 2. Percentage of students who got their first and second preferences in each of the teams. A good allocation should try to maximize this percentage to improve student satisfaction.
- 3. Skills shortfall for each project based on categories in which the average grade for a skill category (between 0 and 4) falls below the requested level for that category. It is derived by summing up the absolute difference between skill level requested and the average skill level of team members for categories where the average skill competency level is lower than the requested level. For example, if a project requested 4 for P, 3 for A, 2 for W and 1 for N, and the average skill competency for these categories are 1.5, 2.5, 2.5 and 3 respectively, then the skills shortfall for that project is 3 being (4 1.5) + (3 2.5). The project manager aims to minimize the skills shortfall and to balance it across teams to ensure all project owners get members with skill competencies close to what was requested.

Extending the Menu Driven Program

You are required to extend the menu-driven program creating two additional options. You are expected to make use of the data stored in the files projects.txt, studentinfo.txt and selections.txt.

- A. Form Team
- B. Display Team Fitness Metrics

In addition, you are required to create 10 JUnit test cases specified below and serialize all the objects (see below).

A. Form Teams

Project manager should be allowed to specify the project ID followed by the IDs of all the students to be assigned to the project. You are expected to develop the appropriate classes.

B. Display Team Fitness Metrics

Create a sub-menu to allow the user to display the required team fitness metrics. This menu should only be enabled after all the teams are formed.

Test Cases (You are required to write both positive and negative test cases)

- a) Average skill competency for a team is computed correctly
- b) Percentage of project members getting first and second project preference is computed correctly
- c) Skill shortfall for any team is computed correctly.
- d) InvalidMember exception thrown when an attempt is made to add a student already assigned to another project team
- e) StudentConflict exception thrown when students members who have indicated prior conflicts are assigned to the same team
- f) PersonalityImbalance exception whenever a team has less than three different personality types.
- g) RepeatedMember exception thrown when an attempt is made to add a student twice to the same team
- h) NoLeader exception thrown when a team does not have a leader personality type
- i) Standard deviation (see formula in last page) in skill competency across projects computed correctly.
- j) Standard deviation for percentage of project members getting first and second project preferences across projects computed correctly
- k) Standard deviation of shortfall across teams is computed correctly.

Serialization

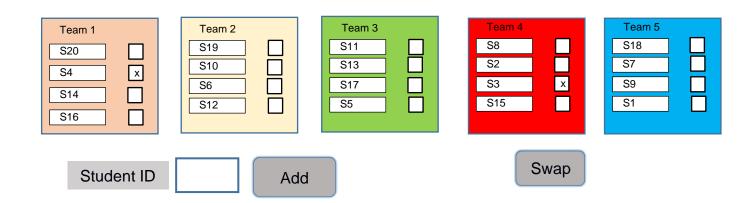
Your program should allow all project and student objects to be serialized and retrieved.

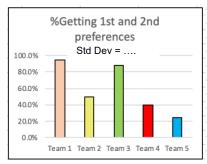


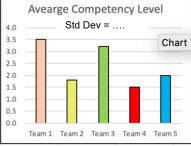
Milestone 3: Visual Sensitivity Analysis (4 Marks)

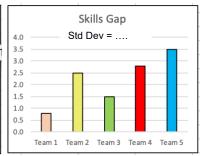
The main aim for this milestone is to assist project manager improve fairness and overall satisfaction of all stakeholders as measured by the metrics by assigning and swapping members between teams. To meet this goal, you are required to incorporate appropriate GUI and event handling for adding and swapping members. You are required to combine the model class from the previous milestone with view and controller classes developed with either Scene Builder and fxml or Java FX directly.

- When adding project manager should specify the Student ID and tick an empty location on one of the teams before pressing Add.
- When swapping user should click the two student IDs to be swapped before clicking the Swap button.
- All the metrics must be immediately updated and shown visually. Compute and display the standard deviation for all three metrics.
- You are expected to display error messages in appropriate dialogs (not console mode) when invalid add or swap attempts are made (such as attempting to swap 3 elements at a time).
- You are required to clearly separate the code for the model, view and the controller.
- Appropriate layout managers to ensure GUI window is consistent even when resized











Milestone 4: Object Relational Mapping of all the Project related Data Captured and Derived (4 Marks)

This milestone requires you to develop the code for object relational mapping. You are required to write all the client, project and student related data to tables in the relational database. The tables created will be used as inputs to external systems. Your tables must include the necessary primary and secondary keys to make the correct associations. This milestone may be demonstrated in console mode (as with milestones 1 and 2) or GUI mode (milestone 3).

Milestone 5: Using Heuristic Techniques for Creating Well Balanced Teams (4 Marks)

This milestone should only be attempted only after completing first three milestone as this will be much more demanding and open ended and you will be expected to learn and use additional patterns and algorithms. You will also be expected make use of multithreading to provide suggestions to the project manager in an unobtrusive way using the console (writing to GUI may interfere with the Java FX threads). The main requirements are:

- Develop algorithms and implement t them to suggest appropriate swaps so that team-fitness metrics can be improved by narrowing the gap between different teams for all three metrics (lowering the standard deviation).
 You need to demonstrate following these suggestions will help improve the metrics.
- o The suggestions should be offered unobtrusively using a separate thread. Your suggestions must not violate any of the hard constraints and lead to a new state which improves all 3 metrics specified.
- o Incorporate an undo button allowing previous state to be restored making use of appropriate patterns.

Assignment Details (30 Marks)

You are required to combine and refactor the code you developed as part of your milestones after fixing any issues identified. 50% of the marks is allocated for the functional aspects specified in the 5 milestones. The remaining 50% of the marks are allocated for code quality, following good object-oriented design, reflective practices (report 1), scalability (report 2) and lessons learnt (report 3). Please refer to the Canvas rubrics for details.

- Correctly preprocessing data used in team formation (refining milestone 1) and handling all the error conditions and exceptions.
- Correctly carrying out at least 10 test cases that demonstrate an in-depth understanding of the problem domain (refining milestone 2) including the metrics used, enforcing all the constraints associated with teams and standard deviation.
- Effectively facilitating visual sensitivity analysis through the use of visual elements and event handing (refining milestone 3). The graphical user interface designed and implemented must be effective, intuitive, usable and fit for purpose. Effectiveness will be measured by how easily team members can be assigned and swapped, how well the visual elements depict the metrics of interest and how error states are avoided.
- Correctly mapping the data from object-oriented paradigm to relational database tables capturing all the relevant attributes and associations (refining milestone 4).
- Use of heuristic techniques, multithreading and patterns (for undo) to assist the project manager form well balanced teams in an unobtrusive and convenient way (refining milestone 5).
- You are required to demonstrate effective use of object-oriented programming constructs (encapsulation, inheritance, polymorphism, abstract classes, interfaces and composition), principles (SOLID) and GoF patterns (such as singleton)
- You code developed as part of the milestone should be refactored adhering to established coding standards to make it readable and maintainable. Your code must be refactored to have a clear separation between model, view and controller classes.
- A one-page document justifying the data structures, collections, generics, algorithms and design techniques used. If you want to describe your design, you may use UML diagrams. You may be asked to explain these.
- A one-page report on scalability (as the number of students and teams increase) considering possible limitations of the algorithms and GUI you developed. You may be asked to explain these.
- A one-page document on lessons-learnt reflecting on the problems encountered and how you would avoid them in a future project. You may be asked to explain these.



4. Submission

You have to zip and submit through Canvas all the relevant materials including the code developed for your console and GUI versions of the assignment, any existing files containing data and pdf versions of the two reports. We strongly recommend you attach a README file with instructions on how to run your application.

No late submissions will be allowed unless you have special consideration. Your assignment will be marked using a collaborative tool during a slot selected by you in week 13 or 14.

Assessment declaration:

When you submit work electronically, you agree to the assessment declaration:

https://www.rmit.edu.au/students/student-essentials/assessment-and-exams/assessment/assessment-declaration

5. Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes
 material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to <a href="https://www.rmit.edu.au/students/studen

When you submit work electronically, you agree to the <u>assessment declaration</u>.

Appendix

Standard deviation is a measure of the amount of variation given by:

$$\mathrm{SD} = \sqrt{\frac{\sum |x - \bar{x}|^2}{n}}$$