MINI-PROJECT MANUAL

SC1003
Introduction to Computational Thinking and Programming

Team Allocation Simulator

Presentation: Week 13 / Week 14
File and Code Submission Deadline: One day prior to scheduled presentation
Peer Evaluation Deadline: 23 Nov 2025 (Sunday)

COLLEGE OF COMPUTING & DATA SCIENCE

NANYANG TECHNOLOGICAL UNIVERSITY

Learning Objectives

- To analyze a problem then design and express its solution in such a way that a computer can effectively carry it out.
- To apply and demonstrate programming concepts or techniques covered
- To apply and exhibit concepts covered in computational thinking

Project Background

The "Introduction to Data Science" is a Year 1 course offered at NTU. This course attracts a diverse group of students from various disciplines due to its widespread applicability and popularity.

Recently, the course has experienced a significant increase in enrollment, with 6,000 students registered. These students are organized into 120 tutorial groups, each consisting of 50 students. The course coordinator is facing challenges in efficiently forming teams for a mini-project component of the course.

To address this issue, the course coordinator seeks your expertise in developing an application capable of organizing students into teams of five for the data science mini-project. You are provided with a csv file (records.csv) consist of 6,000 student records with their Tutorial Group, Student ID, Name, School, Gender, and CGPA. Teams are to be formed only using members from the same tutorial group (i.e., no team can contain two members from different tutorial groups). The application must ensure fairness and diversity when forming teams by considering the following factors:

1. **School Affiliation:** To ensure a mix of knowledge and skills, no team should have a majority of students from the same school.
2. **Gender:** To promote gender diversity, no team should have a majority of students of the same gender.
3. **Current CGPA:** To balance academic performance, teams should not consist predominantly of students with very high or very low CGPAs.

The objective of your program is to strive for balanced and diverse team compositions, taking into account the aforementioned criteria. Some tolerance is acceptable, if it happens that a tutorial group is dominated by students from the same background or profile.

General Requirements

Algorithm Planning, Design, & Development - In this project, you and your team (4 – 5 students) must analyze the student name list (records.csv) and then design and develop an algorithm that is capable of forming teams based on the data, by taking into consideration the team members' backgrounds and profiles. Random allocation without proper logic or allocation criteria will be awarded extremely low marks.

Your program must save the final team formation and listing into a new CSV file with the following columns: “Tutorial Group”, “Student ID”, “School”, “Name”, “Gender”, “CGPA”, **“Team Assigned”**. The “Team Assigned” is the new column.

Algorithm Evaluation - Your team is required to design and implement a mechanism to evaluate and showcase the effectiveness of your algorithm by examining the diversity of the teams it forms. The greater the diversity of the teams, the more effective your algorithm will be. You may utilize Python visualization libraries/packages such as matplotlib, seaborn, or plotly to plot the aggregated outcomes for better illustration.

However, the use of other data manipulation or analytical libraries/packages such as numpy, pandas, scikit-learn, etc., are **strictly prohibited** throughout the entire project. Instead, you and your team should demonstrate your ability to manage and process the data using appropriate data structures covered in this course, such as lists, dictionaries, tuples, trees, etc. Check with your TA if you are unsure.

Computational Thinking (CT) – While working on this project, you and your team must reflect, relate, and document how you have applied computational thinking concepts or processes when solving this problem in Jupyter notebook. Each stage / process / action / decision must be supported by the computational thinking principles with justification or thought processes.

Documentation & Report – Describe and explain how your team solve this problem and working towards the project completion. Ensure that the report structure is logical and well-balanced. Figures, charts and tables are presented with clarity and useful information.

The content may include, but is not limited to, a brief introduction along with team members name (and email), problem decomposition processes, analysis of the data given, planning and designing the algorithm with flow charts or pseudocode, implementation by modularizing the components, evaluating the algorithm, challenges and issues, and lastly, conclusion.

All Python code, explanation, and Computational Thinking reflection must be presented in the form of a Jupyter Notebook. Where possible provide useful comments within the code which will provide the assessor better understanding of your code/logic.

Presentation – Present and explain your Jupyter Notebook to your TA. The presentation will be evaluated based on how well-organized, comprehensive and balanced in content.

All presenters should showcase the essential points of a problem and describe their methods and techniques clearly and precisely. The demonstration is succinct with logical flow to illustrate all required features with clear explanation of ideas. Individual evaluation include demonstrates a thorough and accurate understanding of the subject matter. clarity and coherence of presentation of the part he/she is working on.

Each team **must make appointment and schedule a time (at least a week in advance)** for the presentation with their respective TA. It is recommended to schedule the presentation in Week 13 during the tutorial / lab session. The team may choose to present earlier if they complete the project earlier (Week 12) **or** later (Week 14), **and** is subject to TA availability.

The presentation time limit is **15 minutes (including Q&A by TA)**.

Peer Evaluation – Your individual contribution to the mini-project will be judged through Peer Evaluation, where your Group mates are going to judge your contribution to the project. Each one of you will be asked to judge your own individual contribution against the contribution of each of the other Group mates working together on this.

The primary component of your marks will be awarded for the group effort, and then it will be "scaled" based on your overall peer-assessment score. An example of peer review template for teamwork evaluation are shown in **Appendix A**.

Note: There will be a scope for you to clearly comment on why you graded a team-mate the way you did. This comment is mandatory, as it will help us resolve conflicts within the team, and justify the Peer Evaluation scores to you and your team-mates, if required at any point of time.

An announcement will be posted by the teaching team once the peer evaluation link is made available and due on **23 November 2025 (Sunday)**. Student who missed the peer evaluation will be penalized.

Additional Requirements

Note: Your team must **NOT attempt** this additional (advanced) requirement unless you have completed all the general requirements. This requirement is an extension of the algorithm your team have developed. You may use the `ipywidgets` package to improve the interactivity.

Instead of having 5 students in a team, your program should provide the course coordinator an option to enter the number of students to form a team i.e. (4 – 10).

Your Jupyter notebook should showcase **both algorithms (Basic & Enhanced)**.

Submission Requirements

By the end of this project, your team should submit a **.zip file** to NTU Learn **one day prior to the scheduled presentation. No resubmission is allowed after the presentation.** Ensure you rename all the required files correctly as specified below. Failing to name it correctly may attract a mark penalty.

Zip File:

- The .zip file should contains **a Jupyter notebook**, **an output csv file**, **associated files** (images, or other related file) inside "**assets**" folder, and **requirements.txt**.
- **IMPORTANT**: **DO NOT** INCLUDE YOUR VIRTUAL ENVIRONMENT FOLDER.
- Name your zip file with the following naming conventions: <LabGroup>_<TeamNumber>_<LeaderName>.zip
    - For example: FDBA_Team1_JohnSmith.zip

requirements.txt

- Generate the requirements.txt file with `pip freeze > requirements.txt` while your virtual environment is activated.
- It's the team responsibility to ensure the correct requirements packages are list in the requirements.txt file.

Jupyter Notebook:

- This is the only "report" your team will have to submit.
- Do make sure it runs properly to avoid any error encountered during assessment by your TA. Errors can cause cascading effects, potentially resulting in a significant loss of marks.
- Include the declaration for the use of ai tool(s) in project work following the format in **Appendix B**
- **IMPORTANT**: Ensure that all content included in the .ipynb file is original and not generated using any AI tools. The team will face severe penalties, including the possibility of failing the course, if any AI-generated content is discovered.
- Name your jupyter notebook with the following naming conventions: <LabGroup>_<TeamNumber>_<LeaderName>.ipynb
    - For example: FDBA_Team1_JohnSmith.ipynb

Any associated images, figures or other files:

- Any image or other related files to be loaded from Jupyter Notebook shall be place inside a folder named "**assets**"

Output File:

- Your program should export/save the allocated team as a *new* csv file.
- Name your new student listing with assigned team number with the following naming conventions: <LabGroup>_<TeamNumber>_<LeaderName>.csv
    - For example: FDBA_Team1_JohnSmith.csv

Assessment Components

The breakdown of assessment components is as follows:

- Correctness and completeness of coding (35%)
- Problem decomposition, design consideration, flow chart, pseudocode (15%)
- Quality of demonstration presentation (20%)
- Report (10%)
- Individual Presentation (20%)

Criteria	Standards		
	Fail standard (0 - 39%)	**Pass standard (40 - 74 %)**	**High standard (75 - 100 %)**
Correctness and completeness of coding	Code contains significant errors and does not run as intended. Significant portions of the code are missing or incomplete.	Code runs with minor errors but meets most requirements. Most parts of the code are complete, with some minor parts missing.	Code runs correctly and meets all specified requirements. Code is fully complete and thoroughly addresses all requirements.
Problem decomposition, design consideration, flow chart, pseudocode	Inadequate decomposition with significant gaps. Poor or no design considerations; lacks coherence. Incomplete or incorrect flow chart with major errors. Pseudocode is incorrect or missing significant parts.	Adequate decomposition with some minor issues. Basic design considerations with room for improvement. Flow chart is mostly correct but may have minor errors. Pseudocode is mostly correct but lacks clarity in parts.	Comprehensive and effective decomposition of the problem. Thoughtful and well-integrated design considerations. Accurate, detailed, and well-structured flow chart. Clear, detailed, and correct pseudocode covering all aspects.
Quality of demonstration presentation	The presentation is disorganized, with insufficient content. The computer program fails to run or crashes during the demo. The speaker reads aloud from the slides or notes. The student shows lack of knowledge or confidence and needs much help from other students. The demonstration lacks logical flow for	The presentation covers the essential parts of a project. It is overall well organized. The speaker can speak clearly, although needs some rehearsal to improve the presentation. Some grammatical errors appear in the spoken language. The demonstration is organised with most features explained.	The presentation is well-organized, comprehensive and balanced in content. The speaker can capture the essential points of a problem and describe their methods and results clearly and precisely. The demonstration is succinct with logical flow to illustrate all required features with clear explanation of ideas.

	audience better understanding.		
Quality of written report	The content of report is insufficient to cover the required components. The writing is sketchy, or contains numerous grammatical or typographical errors. The logical structure is shaky, e.g. the conclusion cannot be supported by evidence given.	The content contains most of the required components. The writing is comprehensible, but contains some errors or ambiguity. The report may miss out crucial information, and should give more space for relatively more important parts. Some proofreading and revision may be needed to improve the quality of writing.	The content is not only correct and complete, but also well-written (e.g. logical, coherent, clear). The structure of the report is logically well organized and balanced. There are almost no grammatical or typographical errors. Figures and tables are presented with clarity and useful information.
Individual Presentation	Knowledge and Understanding: Demonstrates little to no understanding of the subject matter. Responses are largely incorrect or irrelevant. Clarity and Coherence: Answers are poorly articulated, lack structure, and are difficult to follow. Frequent grammatical errors and unclear language. Depth and Insight: Minimal to no depth in responses. Little to no evidence of critical thinking or insight into the topic. Use of Evidence and Examples: Rarely uses appropriate evidence or examples to support answers. When examples are used, they are often incorrect or irrelevant.	Knowledge and Understanding: Demonstrates a basic understanding of the subject matter. Responses are generally correct, with some minor inaccuracies. Clarity and Coherence: Answers are mostly clear and structured, though some parts may be difficult to follow. Some grammatical errors, but they do not impede understanding. Depth and Insight: Provides some depth in responses, with occasional critical thinking or insight. Answers go beyond surface-level understanding. Use of Evidence and Examples: Uses appropriate evidence and examples to support answers, though they may not always be fully integrated or relevant.	Knowledge and Understanding: Demonstrates a thorough and accurate understanding of the subject matter. Responses are consistently correct and relevant. Clarity and Coherence: Answers are clear, well-structured, and easy to follow. Few to no grammatical errors. Depth and Insight: Provides in-depth responses with significant critical thinking and insight. Demonstrates a deep understanding of the topic. Use of Evidence and Examples: Effectively uses appropriate evidence and examples to support answers. Examples are well-integrated and highly relevant.

APPENDIX A: EXAMPLE OF PEER REVIEW TEMPLATE FOR TEAMWORK EVALUATION

Criteria (Weighs)	**Score from 1 to 9*** (**1**: Never; **3**: Rarely; **5**: Occasionally; **7**: Frequently; **9**: Always)				
(For 6-members team)	*Member A*	*Member B*	*Member C*	*Member D*	*Member E*
Member name					
a. Fulfilling one's responsibilities duly (15%)					
Behaved responsibly--such as attend meetings punctually and regularly; participate in discussion; complete assigned tasks/roles punctually.	*Score from 1 to 9*				
	Qualitative comments/reasons				
b. Fulfilling one's responsibilities effectively (25%)					
Behaved and contributed effectively--such as quality of work produced; creativity of ideas; extensiveness of research and thinking.	*Score from 1 to 9*				
	Qualitative comments/reasons				
c. Managing interpersonal relationships (30%)					
Listened attentively to and sought inputs from others; helped team resolve conflicts and achieved common understanding to function effectively; promoted respect for others and differences; fostered camaraderie.	*Score from 1 to 9*				
	Qualitative comments/reasons				
d. Providing support to others to achieve goals (30%)					
Behaved fairly and ethically—such as sharing responsibilities and giving credits. Exhibited group citizenship behavior--such as helping others to learn and complete their work through guidance and encouragement; standing up for others when needed.	*Score from 1 to 9*				
	Qualitative comments/reasons				

* Score of 1 should be given only when a team member does not really deserve to be awarded any mark for the team assignment (i.e., zero mark) because the member either has not or has barely participated and/or contributed to the team assignment in any meaningful manner.

APPENDIX B: USE OF AI TOOL(S) IN PROJECT WORK

Each team member should indicate either A or B.

A. I affirm that my contribution(s) to the lab work is my own, produced without help from any AI tool(s)

B. I affirm that my contribution(s) to the lab work has been produced with the help from AI tool(s)

Full Name	Date	A or B

By including this information in your jupyter notebook, you declare that the above affirmation made is true and that you have read and understood NTU's policy on the use of AI tools.

If any team member answered B, the team member(s) must indicate and replicate the table below for every instance AI tool(s) is used.

Name of AI tool	*< For example, ChatGPT>*
Input prompt	*< Insert the question that you asked ChatGPT >*
Date generated	
Output generated	*< Insert the response verbatim from ChatGPT >*
Output screenshot	
Impact on submission	*< Briefly explain which part of your submitted work was ChatGPT's response applied >*