

Software Development Fundamentals Assessment

Date: Thursday, Oct 30 2024

Assessment Time: 0900 - 1700 (including meal breaks)

Overview

There are **2 tasks** in this assessment. Complete all tasks.

Passing mark is **40 (65%)**. Total marks are **62**.

Read this entire document before attempting the assessment. There are 11 pages in this document.

IMPORTANT: Before You Start the Assessment

You must **uninstall all AI coding extensions from your IDE**. AI coding extensions are those that generate entire solutions like ChatGPT, Github Copilot, IntelliCode, etc. If you have installed any of the following AI extensions, please uninstall them from your IDE before you start the assessment. You are also not allowed to use any IDE with AI built-in, like Cursor, Zed, etc.

Random checks will be performed during the assessment. If your IDE is found to have installed any of the above extensions or other AI coding tools, your assessment will be terminated. I forgot to uninstall is not an acceptable explanation.

Code completion extension is permissible; e.g. Angular Language Service, Angular Snippets, Java Language Support, Emmet, etc.

You cannot take this document out of the classroom during the assessment period. You cannot share this assessment document with anyone in whatever form, including scanning and taking pictures, during the assessment period. You cannot communicate with anyone using any means when you are in the assessment classroom.

If you are found to be doing this, your assessment will be terminated.

Internet Access

This is an open book assessment. You may refer to your handwritten notes, computer notes, course slides, or any reference materials.

You may go online to look up information. But you are **only limited to the following sites** listed below

- Official Java doc for Java 23 - <https://docs.oracle.com/en/java/javase/23/docs/api/index.html>
- StackOverflow - <https://stackoverflow.com/>
- Your own GitHub repository

You can access any resources prefixed by the above 3 URLs eg <https://stackoverflow.com/questions/15182496/why-does-this-code-using-random-strings-print-hello-world> is permissible.

You may use Google for searching, but you can only open links listed above.

If you access any other sites that are not in the above list, your assessment will be terminated immediately. If you accidentally open a URL that is not under any of the above lists, close the page IMMEDIATELY. If you linger and start to read its contents, your assessment will be terminated.

Assessment Setup

Unzip the assessment template

(`vttp_b5_assessment_template.zip`). In the unzipped directory, you will find the following subdirectories:

- `task01` — for Task 1
- `task02` — for Task 2

Initialize the unzipped directory as a git repository. Create a git repository on GitHub. The GitHub repository must initially be a **PRIVATE** repository. Click on the 'Private' radio button when you create the repository.

Make the GitHub repository the `origin` of your local repository. Perform a push so that your `origin` is up to date with your local repository.

Your GitHub repository should only be made **PUBLIC after Thursday, 1700 Nov 16 2023** so that the instructors can access your work.

IMPORTANT: This repository is **PRIVATE** and is only accessible to yourself and nobody during the duration of the assessment until **AFTER after Thursday 1700 Oct 30 2024.** If your work is plagiarised by others, you will be considered as a willing party in the aiding and abetting of the dishonest act and will be penalized accordingly.

Assessment

Task 1 (26 Marks)

You are to complete this task with the resources provided in `task01` directory. In the `task01` directory, you will find the following

- `src` directory containing 3 classes
- `day.csv` - data file
- `Readme.txt` - description of the data file

The main class template for this task is provided in the `src` directory. Use this as the entry point for Task 1.

There are 2 other classes in `src`. You may use or ignore these classes. No marks will be deducted from this task if you do not use them, and no extra marks will be awarded if you choose to use them. With the exception of the `Main` class, you are not allowed to modify the two provided classes. You may create additional classes.

In this task, you will be writing a Java program to find the top 5 days with the most/highest number of cyclists from the data file, `day.csv`.

Read each data point (line) from the `day.csv` file into an instance of `BikeEntry`.

Print the details of the top 5 days with the most cyclist in the following format:

```
The <position> recorded number of cyclists was in
<season>, on a <day> in the month of <month>.
There were a total of <total> cyclists. The weather
was <weather>.
<day> was <holiday>.
```

where

```
<position> has the following values - highest, second highest,
third highest, fourth highest, fifth highest
<season> has the following values - spring, summer, fall, winter
<day> has the following values - Monday, Tuesday, ..., Sunday
<month> has the following values - January, February, ...
<total> is the sum of casual and registered cyclist for that day
<weather> see the description in Readme.txt
<holiday> has the following values - a holiday, not a holiday.
```

The following shows an example of the output expected from your program.

```
The highest (position) recorded number of cyclists was in Fall
(season), on a Friday (day) in the month of September (month).
```

There were a total of 8714 (total) cyclist. The weather was Clear, Few clouds, Partly cloudy, Partly cloudy (weather). Friday (day) was not a holiday.

The second highest (position) recorded number of cyclists was in Winter (season), on a Friday (day) in the month of September (month).

There were a total of 8555 (total) cyclist. The weather was Clear, Few clouds, Partly cloudy, Partly cloudy (weather). Friday (day) was not a holiday.

The third highest (position) recorded number of cyclists was in Fall (season), on a Friday (day) in the month of September (month).

There were a total of 8395 (total) cyclist. The weather was Clear, Few clouds, Partly cloudy, Partly cloudy (weather). Friday (day) was not a holiday.

The fourth highest (position) recorded number of cyclists was in Summer (season), on a Thursday (day) in the month of March (month).

There were a total of 8362 (total) cyclist. The weather was Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist (weather).

Thursday (day) was not a holiday.

The fifth highest (position) recorded number of cyclists was in Summer (season), on a Friday (day) in the month of May (month).

There were a total of 8294 (total) cyclist. The weather was Clear, Few clouds, Partly cloudy, Partly cloudy (weather). Friday (day) was not a holiday.

The description of the columns in the `day.csv` file can be found in `Readme.txt` under the 'Dataset characteristics' section.

IMPORTANT: Marks will be deducted if you commit `.class` files into your Git repository.

Marks will be awarded for proper abstraction of your program.

Task 2 (36 marks)

Use the resources provided in `task02` directory for this task.

In the `task02` directory, you will find the following:

- `src` directory with a file containing a template for main class
- `TTT` directory containing tic-tac-toe board configurations. There are 6 files in `TTT`.

Use the main class template is provided in the `src` directory as the entry point for Task 2.

In this task, you will be writing a program to evaluate all the possible cell positions where you can place an X for a given tic-tac-toe (TTT) board configuration.

Consider the following TTT board with the coordinates show on the board's edge.

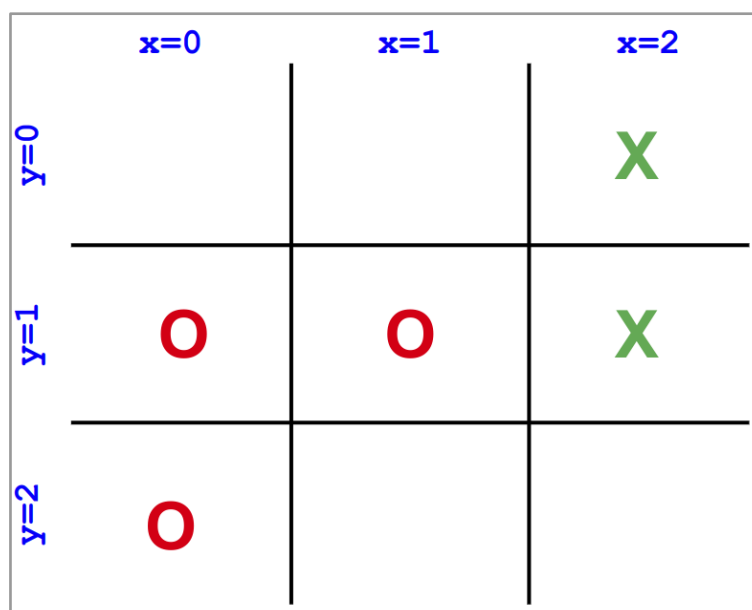


Figure 1

You are playing X and you will be making the next move. You have to determine where to place your X by performing the following:

- Find all the legal positions where you can place an X.
- Utility of placing in that position. Utility is the 'profitability' or 'returns' you gain from that position. It is a way to evaluate how 'good' is the position you have chosen in relation to all other positions. More on this in later.

The following is an example of evaluating the TTT board shown in Figure 1.

First we get a list of legal positions where you can place an X:

- (x=0, y=0)
- (x=1, y=0)
- (x=1, y=2)
- (x=2, y=2)

Next, we determine the utility of each of the positions if you were to place an X there:

- If you place an X at (x=0, y=0) - O cannot win by placing in any of the remaining available positions on its move (after you have placed your X). Neither can X. Placing an X in this position (x=0, y=0) has a neutral utility.
- If you place an X at (x=1, y=0) - O can win by placing in (x=0, y=0) on its move. Placing an X here (x=1, y=0) will cause you to lose the game. This position has a negative utility.
- If you place an X at (x=1, y=2) - This position has the same utility as placing an X at (x=1, y=0).
- If you place an X at (x=2, y=2) - You will win the game. This position has a positive utility.

The following table summarises all the positions and their respective utilities.

Position	Utility
(x=0, y=0)	Neutral
(x=1, y=0)	Negative (you lose)
(x=1, y=2)	Negative (you lose)
(x=2, y=2)	Positive (you win)

The TTT directory contains a set of pre-configured TTT boards. X and O represent the positions claimed by the X and O players. A dot (.) represents an empty space viz. you can place an X in these locations.

Figure 1 board would be represented as

```

. .X
OOX
O. .

```

Write a program to evaluate the utility of all the legal positions on a board that you can place an X.

Use the provided `Main` class in the `src` directory as this task's entry point. The TTT board configuration should be passed into the program as a parameter on the command line, as shown below:

```
java -cp classes vttp.batch5.sdf.task02.Main TTT/figure1.txt
```

where `classes` is the directory of your class files.

After evaluation, your program should output the following:

- the TTT board
- list of all the legal positions that you can place an X and its corresponding utility. If the utility is neutral, display a zero (0), positive a one (1) and negative a minus one (-1).

An example of task02 output is shown below:

```
Processing: TTT/figure1.txt
```

```
Board:
```

```

. .X
OOX
O. .

```

```
-----
```

```
y=0, x=0, utility=0
```

```
y=0, x=1, utility=-1
```



```
y=2, x=1, utility=-1
y=2, x=2, utility=1
```

Display an appropriate error message if no TTT configuration file is provided.

You should test your program against all the given board configurations in `TTT` directory.

The following is a high-level pseudocode to determine the utility of a position. Feel free to use it or use your own algorithm.

```
tttboard = read board configuration file
empty_pos := tttboard.get_all_empty_pos()
utility := map
for every pos in empty_pos begin
  new_tttboard = clone tttboard
  new_tttboard.place(X, pos)
  evaluate horizontal, vertical, and diagonal rows on new_tttboard
  if there are 3 X
    utility[pos] := 1
  else if there are 2 O and 1 SPACE
    utility[pos] := -1
  else
    utility[pos] := 0
end
```

No marks will be deducted if you do not use the above algorithm; no extra marks will be awarded if you do implement the above algorithm.

IMPORTANT: Marks will be deducted if you commit `.class` files into your Git repository.

Marks will be awarded for proper abstraction of your program.

Do not delete the `run.sh` file.

Submission

You must submit your assessment by pushing it to your repository at GitHub.

Only commits on or before **Wednesday 1700 Oct 30 2024** will be accepted. Any commits after **Wednesday 1700 Oct 30 2024** will not be accepted. No other form of submission will be accepted (eg. ZIP file).

Remember to **make your repository public after 1700/5PM Wednesday 1700 Oct 30 2024** so the instructors can review your submission. If you forget to make your repository public, the instructors will not send an email to remind you to make your repository public. The instructor will also not mark your assessment.

After committing your work, post the following information to Slack channel `#01-sdf-submission`.

1. Your official name (as it appears in your NRIC)
2. Your email
3. Your group: Group A or Group B
4. Git repository URL. Remember to make your repository **PUBLIC after 1700/5PM Wednesday 1700 Oct 30 2024**

It is your responsibility to ensure that all the above submission requirements are met. Your assessment submission will not be accepted if

1. Any of the 4 items mentioned above is missing from the `#01-sdf-submission` channel, and/or
2. Your information did not comply with the submission requirements, eg. not providing your full name and/or
3. Your repository is not publicly accessible after **1700/5PM Wednesday 1700 Oct 30 2024**

You should post the submission information to the Slack channel #01-sdf-submission no later than **Wednesday 1700 Oct 30 2024**

Academic Integrity

This is an open book assessment. You may search the Internet for resources or use reference books during the assessment. The assessment must be your own work. You cannot ask a third party to write any part of this assessment or use AI tools such as ChatGPT to generate output and submit it as part of your assessment. This will result in an automatic disqualification from the assessment.

The NUS ISS takes a strict view of cheating in any form, deceptive fabrication, plagiarism, and violation of intellectual property and copyright laws. Any student who is found to have engaged in such misconduct will be subject to disciplinary action by NUS ISS.

You are to ensure the integrity and working condition of your PC/notebooks (eg. wireless/internet connection, battery, screen, accidents like water spillage) during the assessment. NUS ISS will not accept any of these as a reason for deferring or retaking your assessment.