Computer Science 2A

Practical Assignment 02

Assignment date: 2023-02-28

Deadline 2023-03-07 12h00

Marks: 150

This practical assignment must be uploaded to eve.uj.ac.za **before** 2023-03-07 12h00. Late[1] or incorrect submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student.

Good coding practices include a proper coding convention and a good use of documentation. Marks will be deducted if these are not present. Every submission **must** include a batch file unless stated otherwise.

The **reminder page** includes details for submission. Please ensure that **ALL** submissions follow the guidelines. The reminder page can be found on the last page of this practical.

This practical aims to familiarise you with Problem Analysis and Model Classes.

The year is 3090 and intergalactic travel is a norm. The Milky Way Space Communication Board (MWSCB) oversees transmission of messages within the Milky Way. [2] You work with a department that manages interplanetary communication. Your department only transports messages to and from Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune and Pluto. The messages are delivered by Spaceships designed to transport messages.

Each Spaceship carries messages that include the language of communication, source, and destination planets. The **`Ship`** class will contain the following information:

`ID` Unique Identifier of the SpaceShip, starts with SH, then 4 additional digits eg. SH0076
`Name` Name of the Spaceship
`Messages` Array of **`Message`**s that the Spaceship is carrying

The **`Ship`** has a ***`printMessages`*** method that displays the messages transmitted by the Spaceship. The **`Message`** class contains the following information:

`ID` Unique Identifier of the **`Message`**, starts with MSG, then 6 additional digits eg. MSG007600
`Language` Language of the text being transmitted
`Contents` The text being transmitted
`SourcePlanet` The Planet from which the **`Message`** is being sent
`DestinationPlanet` The Planet to which the **`Message`** is being sent

Using the information provided in the problem statement above do the following:

[1] Alternate arrangements for exceptional circumstances will been posted on eve.

[2] Disclaimer - This series of problem statements are a work of fiction. Names, characters, businesses, places, events and incidents are either the products of the author's imagination or used in a fictitious manner. Any resemblance to actual persons, living or dead, or actual events is purely coincidental.

- Create a **`Ship`** class with the required attributes[3].
- Create a **`Message`** class with the required attributes[3].
  - Make use of an ENUM TYPE for the Planets.
- Create a **`Main`** class.
  - Create three **`Ship`** instances.
  - For each **`Ship`** add three **`Message`**s.
  - Call the ***`printMessages`*** method on each **`Ship`**.
  - Make sure the output is well formatted.

Place the relevant classes into the **`acsse.csc2a`** package[4].

Mark Sheet

1. UML class diagram **[10]**
2. **`Ship`** class
   (a) Attributes (4 marks per attribute) **[12]**
   (b) ***`printMessages`*** Method **[05]**
3. **`Message`** class
   (a) Attributes (2 marks per attribute) **[10]**
   (b) Enum Type **[03]**
4. **`Main`** class **[10]**
5. Packages **[05]**
6. Coding convention (structure, layout, OO design) **[05]**
7. Commenting (normal and JavaDoc commenting) **[05]**
8. Correct execution **[85]**

NB

Submissions which **do not compile** will be capped at 40%!

Practical marks are awarded subject to the student's ability to explain the concepts and decisions made in preparing the practical assignment solution. (Inability to explain code = inability to be given marks.)

Execution marks are awarded for a correctly functioning application and not for having related code.

[3] Hint: remember that accessor and mutator methods for attributes are required.
[4] Hint: The **`Main`** class does not need to be in a package

Reminder

Your submission must follow the naming convention below.

`SURNAME_INITIALS_STUDENTNUMBER_SUBJECTCODE_YEAR_PRACTICALNUMBER`

Example

Surname	Berners-Lee	**Module Code**	CSC02A2
Initials	TJ	**Current Year**	2023
Student number	209912345	**Practical number**	P02

`Berners-Lee_TJ_209912345_ CSC02A2_2023_P02`

Your submission must include the following folders:

Folder	State	Purpose
`bin`	*Required*	Should be empty at submission but will contain runnable binaries when your submission is compiled.
`docs`	*Required*	Contains the batch file to compile your solution, UML diagrams, and any additional documentation files. All files must be in **PDF** format. Your details must be included at the top of any **PDF** files submitted. **Do not include generated JavaDoc.**
`src`	*Required*	Contains all relevant source code. Source code must be places in relevant sub-packages! Your details must be included at the top of the source code.
`data`	*Optional*	Contains all data files needed to run your solution.
`lib`	*Optional*	Contains all libraries needed to compile and run your solution.

NB

Every submission **must** include a batch file that contains commands which will:

- Compile your Java application source code.
- Compile the associated application JavaDoc.
- Run the application.

Do not include generated JavaDoc in your submission. All of the classes/methods which were created/updated need to have JavaDoc comments.

Multiple uploads

Note that only **one** submission is marked. If you already have submitted once and want to upload a newer version then submit a newer file with the same name as the uploaded file in order to overwrite it.