



Computer Science 2A

Practical Assignment 07

2016-04-05

Time: Deadline — 2016-04-12 12h00

Marks: 60

This practical assignment must be uploaded to eve.uj.ac.za **before** 2016-04-12 12h00. Late 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 [JavaDoc comments](#). Marks will be deducted if these are not present. Every submission **must** include a batch file. See the reminder page for more details.

The Java Development Kit (JDK) has been installed on the laboratory computers along with the [Eclipse](#) Integrated Development Environment (IDE).

This practical aims to solidify your understanding of Generics and extends low level rendering.

The story continues from Practical06. After building a layout GUI you are now able to read in the location of crew members on a ship. In order to read the locations some changes to the current classes are needed.

Modify the **CrewRoster** *readRoster* method to make use of an **ArrayList** of **CrewMembers**. This will also modify code in other classes that use this method.

Locations are stored in a binary file. The file stores locations in the following sequence:

ID ROW COLUMN

where

- **ID** - ID of crew member
- **ROW** - Row of crew member
- **COLUMN** - Column of crew member

Remember that binary files store information as a sequence of bytes. In this case the ID will be stored as a UTF string while the row and column will be stored as bytes.

Create a **CrewEntity** class. This class will contain a **CrewMember** as well as the row and column of the **CrewMember**. This class will be used to draw a **CrewMember** on the **ShipPanel**. The **CrewMember** attribute must be read-only.

Create a static **readLocations** method in the **CrewRoster** class. This class will require the file name of the location file as well an **ArrayList** of **CrewMembers**. Read the locations of **CrewMembers** found in the location file and return an **ArrayList** of **CrewEntities**. Remember to match a **CrewMember** instance found in the roster file to an ID in the location file.

Inside the **paintComponent** method of the **ShipPanel** class, draw all available **CrewEntity** instances which have been read from the files. A **CrewEntity** can be drawn as a filled circle which exists in a tile at the proper row and column.

The **Main** method remains unchanged.

Mark sheet

- | | |
|---|------|
| 1. CrewEntity (attributes, accessors, mutators) | [08] |
| 2. CrewRoster | |
| (a) readRoster use ArrayLists . | [02] |
| (b) readLocations | |
| i. Read binary information from file | [04] |
| ii. Match CrewMember and create a CrewEntity | [04] |
| iii. Exception Handling and Resource Management | [02] |
| 3. ShipPanel paintComponent - draw CrewEntity | [05] |
| 4. Packages | [05] |
| 5. Coding convention (structure, layout, OO design) and commenting (normal and JavaDoc commenting). | [10] |
| 6. Correct execution. | [20] |

NB

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

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

Reminder

Your submission must follow the naming convention as set out in the general learning guide.

SURNAME_INITIALS_STUDENTNUMBER_SUBJECTCODE_YEAR_PRACTICALNUMBER

Example

| | |
|------------------|-------------|
| Surname | Berners-Lee |
| Initials | TJ |
| Student number | 209912345 |
| Module Code | CSC2A10 |
| Current Year | 2016 |
| Practical number | P07 |

Berners-Lee_TJ_209912345_CSC2A10_2016_P07

Your submission must include the following folders:

- **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. Do not include generated JavaDoc.
- **src** - (*Required*) Contains all relevant source code. Source code must be placed in relevant sub-packages!
- **data** - (*Optional*) Contains all data files needed to run your solution.
- **lib** - (*Optional*) Contains all libraries needed to compile your solution.

NB

Every submission **must** include a batch file. This batch file must contain commands which will compile your Java application source code, compile the associated application JavaDoc and 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.