

# Assessment details for ALL students

## Assessment item 1— Assignment 1

---

|            |     | ASSESSMENT |
|------------|-----|------------|
| Weighting: | 20% |            |
| Length:    | NA  | 1          |

---

### 1. Objectives

The purpose of this assessment item is to assess your skills attributable to the following learning outcomes and achieving the expected graduate attributes of advanced level knowledge, cognitive, technical, and creative skills, and self-management.

- Design and implement appropriate data structures for application development
- Analyse, develop and implement software solutions with the focus of data structures and algorithms
- Apply classes, inheritance, polymorphism, and exception handling

### 2. Assessment task

Your task in this assessment is to analyse the given problem, model, and design the required data structures using UML class diagrams. You will be implementing the software solution, applying efficient algorithms, inheritance, polymorphism, and exception handling. The topics required for this assessment task are from Weeks 1-5. Your Java Application should have an interactive Graphical User Interface (GUI) based on the JFrame class using SWING GUI components. You should also write a report, as specified in this document, demonstrating your conceptual knowledge.

#### 2.1 Problem

Statistical data show that ready-to-eat (RTE) cereals account for the bulk (78.5%) of all breakfast foods. Often, fresh juice is accompanied by the breakfast cereals. A large variety, approximately over 4900, of products exist in the breakfast cereal category which widely varies in their nutrient content. Even though a nutrition panel with nutrition information is provided on the pack, it is very difficult for a consumer to make an informed decision quickly to choose a comparatively healthier product. Therefore, a processed food suitability assessor (PFSA) is suggested, which can enable consumers to make informed decision making on their choice of processed food items. You are invited to design and develop an initial prototype for the PFSA to provide listing of items with nutrition information and to enable a user to select a cereal and beverage which will then provide the total nutrition information for the selection. You will be developing the Java application with a GUI. You are given a .csv file named *processedFoodData.csv* (available at the Unit website) with necessary data that can be used. The processed food details available in the file include the amounts of energy in kilo joules, protein, fat, carbohydrate, sugar, and dietary fibre in grams/per serve, and sodium in milligrams

per serve. This file also contains the item name, brand name, serve size, and unit. The file includes a list of cereals and beverages.

The users accessing the PFSA should be able to view the following data:

1. List of a chosen cereal and beverage with individual and total nutrition information.

The PFSA should display the names of cereals and beverages in two separate ComboBoxes enabling the user to choose one item from each list. Once the items are selected the user should be able to click the 'Display Choices' button to display the details.

Additional processed food items in the categories of snacks and sweetened beverage will be added after testing and approving the prototype. This is not in your scope of work as it is only after the prototype gets approved.

## 2.2 Design Guidelines

You can use the following guidelines in your modelling and GUI design.

### Graphical User Interface

The application should have a GUI for user interactions. This GUI should have the necessary components to enable the user to execute all the functions as provided in Section 2.1 above. You may use the guidelines provided below for your GUI design and implementation. Variations to the provided guidelines are acceptable as long as it meets the user requirements.

#### i) *Top Panel*

The top panel can have a Label and TextField, to display the label 'User Name' and enable the user to enter the name in the TextField. The second row of the top panel can have two ComboBoxes with two associated labels to display the list of cereals and beverages.

#### ii) *Middle Panel*

This panel can have a TextArea to display welcome message to the User and the food choices and nutrition values.

#### iii) *Bottom Panel*

This Panel can have a label "Command Buttons" and can contain the required Command Buttons of 'Enter Data', 'Display Choices', 'Clear Display' and 'Quit'.

A full GUI design diagram is not given to encourage you to design using your own colour choices and other attributes for the GUI components.

The functions of the four Command Buttons are given below.

### Initialisation

Your application should start by loading the data from the data file and filling the ComboBoxes with the food and beverage list. The list of food and beverage items and associated nutrient values are given in the data file (processedFoodData.csv) available at the Unit website.

### Enter Data Button

The user enters the name using the TextField and selects food and beverage items using the list filled in the ComboBoxes. Once the user clicks the 'Enter Data' your application should store the entered data in appropriate data structures and display a message in the TextArea addressing the user by name and giving appropriate guideline to display the User's food choices and nutrition information. *Display error messages if the user hasn't entered name or chosen items.*

An example display is shown below. You can design and choose your own wordings for the message to make it user friendly.

Hello Mary Welcome to Processed Food Assessor System

Click the displayChoices button to view details of your choices

You should also display appropriate error messages to help the user to complete the entry.

### Display Choices Button

The user clicks this button to view the details of the food and beverage chosen using the ComboBoxes. Display appropriate error messages if the user clicks this button before choosing items and clicking the 'Enter Data button'.

An example display is shown below.

| Food Type | Item Name                      | Brand          | Serve Size | Unit | Energy | Protein | Fat  | Carb  | Sugar | Fibre | Sodium |
|-----------|--------------------------------|----------------|------------|------|--------|---------|------|-------|-------|-------|--------|
| cereal    | Uncle Tobys Plus Antioxidant   | Uncle Tobys    | 40         | gm   | 595.00 | 2.90    | 0.60 | 28.10 | 8.50  | 4.50  | 80.00  |
| Beverage  | Original Juice Co.Orange Juice | Original Juice | 200        | ml   | 356.00 | 1.60    | 0.60 | 18.00 | 16.00 | 1.20  | 14.00  |
| Total     |                                |                |            |      | 951.00 | 4.5     | 1.20 | 46.10 | 24.50 | 5.70  | 94.00  |

### Clear Display Button

Clicking on the "Clear Display" button should clear all contents from the TextArea, the TextField, and the data structures storing the user's selection. The ComboBoxes will still have the list loaded.

### Quit

The "Quit" button should allow the user to exit from the application.

### Data Structures

You may follow the class design given below. You should read the specification carefully and design the classes using UML class diagrams and include the attributes with correct data types and all the methods, clearly indicating public or private. Clarify your doubts during the design stage so that your implementation can be less error prone and faster to complete.

#### i) *Item class*

This is a generic class and include appropriate fields to store:

an itemName

a category.

This class should have:

a parameterised constructor, accessor, mutator methods,

a toString() method.

**ii) Nutrient class**

This class stores the nutrient name and nutrient amount. This class should have a parameterised constructor, accessor, mutator methods, and a toString() method.

**iii) DataFile class**

This class will have a parameterised constructor that takes the filename to be processed.

Include an ArrayList<> type attribute to read and load the data from the file.

Include a method to load data from the given file named 'processedFoodData.csv' available from the Unit website and return the ArrayList<>.

Use necessary file handling exceptions.

**iv) ProcessedFood class**

This class extends the Item class.

In addition to the inherited attributes, this class stores:

- the brand name,

- serve size,

- serve unit (gm, or ml)

- an ArrayList of Nutrient objects to store the nutrient details.

This class should have a parameterised constructor, copy constructor, a default constructor, accessor, mutator methods, and a toString() method.

**v) ProcessFoodGUI class**

This class should have all the GUI components as described under the *Graphical User Interface Design* section. This class stores the data file name, creates the DataFile object and invokes its method to read and load the data from the file. This class will initialize the GUI components and use the loaded data from the file to set values for the comboBoxes. This class will have a userName attribute to store the entered user name. This class will have the main method and starts the application.

*Hints and tips: Refer to Week 3 Lecture and tutorial solutions, and the Textbook chapter on Polymorphism for the correct implementation of polymorphic behaviour.*

### 3. Coding

Include necessary accessor, mutator methods, constructors, and toString() method for each class. Also, follow good coding practices, using meaningful names, camel case notation for naming, constants as necessary, and include meaningful comments. You can use NetBeans to develop your application.

*Data: The data required is available in the file named 'processedFoodData.csv' available at the Unit website under the Assessment folder on the top of the page. This is a text file and you can use text file processing to load the file.*

### 4. Report

You should submit a report containing the following details.

1. UML class diagrams for the classes

2. Test plan showing input data, expected results, and actual results

## 5. Assignment Submission

You should submit the following source code files and word document using the Moodle online submission system as one zip file (Note: the file names/class names could be changed to meaningful names). *Please do not zip the whole project folder and submit it.*

- Item.java – Source code for the Item class
- Nutrient.java – Source code for the Nutrient class
- ProcessedFood.java – Source code for the ProcessedFood class
- ProcessedFoodGUI.java – Source code for ProcessedFoodGUI class
- DataFile.java – Source code for DataFile class
- Report.doc – Your word document containing the report.

## Assessment Item 1 Marking criteria

| S.No | Total Marks - 20  | Marks Allocated | Marks Scored |
|------|---|-----------------|--------------|
| 1    | Graphical User Interface Presentation   | 2               |              |
| 2    | Design and use of appropriate data structures and UML class diagrams  | 2               |              |
|      | Implementing classes and using objects, methods, aggregation, and polymorphism  | 4               |              |
|      | Use of exception handling   | 1               |              |
|      | Correct reading and loading of data from the file   | 2               |              |
| 6    | "Enter Data" button function  | 2               |              |
|      | "Display Choices" button function   | 2               |              |
|      | "Clear Display" and "Quit" buttons functions  | 1               |              |
| 12   | Good coding practices including comments, indentation, use of constants as required, use of meaningful names, and camelCase notation for names  | 1               |              |
| 13   | Well presented report with student details, UML class diagram, and demonstration of testing all functions of the program  | 3               |              |
|      | Penalties   |                 |              |
|      | Late Penalty (-1.5 marks (5% of total allocated marks) per calendar day)  |                 |              |
|      | Plagiarism (source code found entirely different from the styles followed in the Unit or containing constructs outside of this Unit or a pre-requisite unit will be scrutinised for plagiarism) |                 |              |
|      | <b>Total</b>  | <b>20</b>       |              |

COIT20256

Data Structures and Algorithms

Term 3 2018