

COIT20256 Assessment item 2— Assignment 2

Due date: Thursday of Week 11 (3rd Oct 2019) 11: 50 pm AEST

ASSESSMENT

Weighting: 30%

Length: NA

2

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.

- Analyse, develop, and implement software solutions
- Choose and compare appropriate data structures and algorithms in program design
- Apply classes, inheritance, polymorphism, and exception handling
- Test software implementations to ensure correctness and maintainability

2. Assessment Task

There are two design and implementation parts, Part A and Part B, in this assignment. Read the assignment description, think about the points in the description, and connect the points to understand the clear picture of the overall task that you are required to do. The better and earlier you understand the overall task, the better and earlier you can design, implement, test, and document your task. Start early!

2.1 Part A

In this assignment, you are required to write a Java Application that uses an interactive Graphical User Interface (GUI) based on the JFrame class using SWING GUI components. You will also be designing and implementing the software solution using appropriate data structures and application of classes, inheritance, and exception handling. The case study for developing a solution is given below.

The food consumption of a person is to meet the daily energy requirements. The energy requirement is estimated using a person's age group, gender, height, weight, basal metabolic rate (BMR), and physical activity level (PAL). There are data available on estimated energy requirements for various PALs and BMR. So, by choosing the correct age group, gender, height and PAL, the energy requirements can be found from the data set. *Use the given data file named COIT20256Ass1Data.csv available on the moodle, as it contains the set of data.*

In Assignment One, you have developed a **Java Application** that lets a user enter height and choose gender, age group, and the most suitable PAL; and the application displays the energy requirements in Mega Joules.

In this assignment (Assignment Two), you are **required to extend your first assignment** so that the user can save the user's details to mysql database table(s). You are required to programmatically create database and appropriate table(s) to store the user's details; and the user's details include the data such as the user's name, height, age group, gender, PAL description, PAL level or value, energy requirement, and expected weight. Your application should display all users' data from the database table(s).

In addition to the GUI components you have in Assignment One, include the following two components to the Assignment Two as outlined below:

- a. **Save** Button in the bottom panel is to save the user's data to the database table(s). The user's data as mentioned above include the user's name, height, age group, gender, PAL description, PAL level or value, energy requirement, and expected weight
- b. **Display All** Button in the bottom panel is to display all users' data from the database table(s). This Button enables the user to view all the database records that display all users' data on the display area in a given order. The given order of display includes Name, Age Group, Gender, PAL Description, PAL Value, Height, Energy Needs, Expected Weight. The database records will have a unique id. It is not essential to display the id.

More details of the functionalities of the above buttons are as follow:

A. Save

For this *Save* button in the Assignment Two, assume that a user with appropriate name and password has logged into the application and the pressed the *Load Data* button to load the data file. COIT20256Ass1Data.csv is provided for you on the Moodle website as part of the Assignment One. You have already implemented the functionalities of *Login* and *Load Data* buttons in Assignment One. Screenshots are not provided so that you can choose appropriate design to include the **Save button**. You are allowed to integrate the required parts from Assignment One solution given to you on the Moodle Website. You should not include the whole source files into your project.

A user can click on the **Save button** to store or insert the data into the database table(s). Although name was not in the data file, the user entered the name during the login; and you are required to capture the name in your query to insert the data into the table(s). The user's data as mentioned above now include the user's Name, Height, Age group, Gender, PAL description, PAL level or value, Energy requirement, and Expected weight.

B. Display All

The user can use the **Display All** button only after entering the user data and saving them by using **Save** button. This **Display All** button enables the user to view all the database records and display all users' data on the display area in a given order. The given order of display includes Name, Age Group, Gender, PAL Description, PAL Value, Height, Energy Needs, Expected Weight. The database records will have a unique id. It is not essential to display the id.

Note: All buttons that you have implemented in line with the Assignment One should function as prescribed in the Assignment One.

2.2 Part B

Your task in this part of the Assignment is to demonstrate the appropriate use of various data structures such as stack, priority queue, set, and map for the given data.

While using these data structures for the given data, explain the benefits that you can achieve by using the appropriate data structures. You need to complete the following steps in your use of various data structures.

- i. Read the names from the file and load in a priority queue. Display the values from the priority queue using the peek() method. Describe your observation of change to the list of names in the report file.
- ii. Move the list of names from the priority queue to stack. Move the list of names from stack while displaying the value to a set. Describe your observation of the stack list of names.
- iii. Display the names from the set and again describe your observation the list of names.
- iv. Create a map and store the names with associated student Id and display the values.

3. Data Structures

In addition to the classes that you have in Assignment One, you are required to create and use the *DatabaseUtility* class and *DataStructures* class as outlined below for your design.

DatabaseUtility class needs to implement all the tasks related to

- i. creating the database and the table, userEnergy programmatically; Give meaningful names to database and table(s).
- ii. inserting the data into the database table(s) programmatically
- iii. retrieving or extracting records from the database table(s) programmatically.

Include a **LinkedList** of *UserEnergyNeeds* class to store the extracted data from the database. Use *PreparedStatement*s to get the user input from GUI interface and use them for populating database table(s). Also use *PreparedStatement* to execute query to extract data and store the records in the linked list.

4. Coding

You can build your application using NetBeans. It is highly recommended that you create the GUI components using code rather than 'designer' and 'click and drag' of the NetBeans. This ('designer' and 'click and drag' of the NetBeans) creates code which is not maintainable. Remember, this is a good prototyping tool, but not recommended for coding. *Commence with one class at a time, test it and then incrementally add the next.*

Follow good coding practices by using meaningful names, camel case notation for naming, constants to avoid numbers inside the body of the methods, as necessary. *You should follow the Java Coding Standards that you can find under the section of Unit Resources.* Include meaningful comments.

5. Assignment Submission

You should submit one zip file containing the following files using the Moodle online submission system. (Note: the file names/class names could be changed to meaningful names)

- i. BMRdata.java – Source code for the *BMRdata* class
- ii. EnergyData.java – Source code for the *EnergyData* class
- iii. User.java – Source code for the *User* class
- iv. UserEnergyNeeds.java – Source code for the *UserEnergyNeeds* class
- v. DatabaseUtility.java – Source code for the class to programmatically create database and table(s), insert data into and retrieve data from the table(s).
- vi. UserEnergyInterfaceGUI.java –Source code for the GUI components and the main method.
- vii. DataStructures.java – Source code containing the use of PriorityQueue, Stack, Set, Map or HashMap as per the Part B.
- viii. A report (Report.docx) containing the following details:
 - a. Brief outline of your database and table(s) related to the Part A. Use diagram(s) or screenshots where appropriate.
 - b. Include Test plans showing input data, expected results, and actual results; provide clear screen shots to demonstrate the evidence of your testing
 - c. Take screenshots of the functionality of the *Save* button. As you can recall from the Assignment One that after appropriately clicking on *Login* button (as you have done in the Assignment One), all data from the given file (*COIT20256Ass1Data.csv*) become available on the different components of the user interface; clicking on the *Save* button (as you are required to do in the Assignment Two) will save or insert the user's data to the database table(s). The user's data as mentioned above include the user's name, height, age group, gender, PAL description, PAL level or value, energy requirement, and expected weight. You need to take the meaningfully linked screenshots to justify the functionality of the *Save* button

- d. Take screenshots of the functionality of the *DisplayAll* button. Assuming that you have entered or inserted five users' data using the *Save* button, a click on the *DisplayAll* button will display all of the five users' data on the display area.
- e. Write your observations of using the various data structures in Part B to store the list of names and including studentId.

Please do not zip the whole project folder, and submit it.

Assessment Item 2 Marking Criteria

S.No	Total Marks - 30	Marks Allocated	Marks Scored
1	Graphical User Interface Presentation (Use appropriately sized components and alignments) (0 mark if program doesn't run or any malfunction)	2	
2	Design and use of appropriate data structures	2	
3	Design and implementation of DatabaseUtility class enabling correct creation of database, table(s) and access to the database and the table(s)	4	
4	Use of exception handling (use appropriate exception classes and informative messages)	2	
5	Functionality of <i>Save</i> button	2	
6	Functionality of <i>DisplayAll</i> button	2	
7	Correct use of Priority Queue to store names	3	
	Correct use of Stack to store names and display	2	
	Correct use of set to store names	2	
	Correct use of Map to store a name with studentId	3	
9	Good coding practices(Indentation, Comments, Naming Conventions, Readability)	1	
10	Well presented report with student details, observation of data structure characteristics , test plan, and test details	5	
	Penalties		
	Penalty for the late submission (-1.5 per day or part of a day)		
	Source code found entirely different from the styles followed in the Unit or containing constructs outside of this Unit will be penalised (-1 to -10 marks depending on the amount of source code)		
	Plagiarism (penalty as per the plagiarism policy)		
	Total	30	