

Assessment details for All students

Assessment item 1

Due Date:	Thursday of Week 6 (21st April) 11.45 pm AEST
Weighing:	15%

Objectives

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

Assessment Task

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. Your task is to develop a **Java Application** that lets the user to enter height and choose gender, age group, and the most suitable PAL so that the application displays the energy requirements in Mega Joules. The application should contain a GUI as shown below. The GUI components should consist of the following panels.

1. A top panel that contains four (4) Text Fields, one Password Field, two Combo Boxes (drop down list), seven Labels, and two Radio Buttons and a Login command button.
2. A middle panel that contains a Text area to display the user's estimated energy requirement details.
3. A bottom panel that contains four Buttons which are "Load Data" Button, "Display Energy" Button, "Clear Display" Button and "Quit" Button.

The functions of four buttons are described below.

1. Load Data

Initially the 'Load Data' Button remains disabled as shown in Figure 1. To enable it the user enters the name and password using the fields on the GUI, and clicks the

Login button. The Load Data Button can be used to read the data file which contains height, weight, BMR, and energy requirements for various PALs for males and females in the age groups of 19-30, and 31-50. After loading the file, fill the Combo Box with the description of PAL as given in the table below. Also, the Age group Combo Box is filled with the list of age groups read from the file. One row in

Table 1 PAL Description and Value

Description	Value
bed rest	1.2
very sedentary	1.4
light	1.6
moderate	1.8
heavy	2
vigorous	2.2

the file contains data for one age group, but for both genders.

Note: Use the given data file named *COIT20256Ass1Data.csv* available on the course website and it contains the set of data.

Figure 1: Initial Screen Display

2. Display Energy

Before clicking on the “Display Energy” button, the user has to enter his/her height in the Text Field, choose the gender by clicking the Radio button and select the age group. This should display the ‘Expected Average Weight’ for that height in the Text Field as provided in the file. When user clicks ‘Display Energy’ after selecting a

suitable PAL from the drop down list, the corresponding PAL number should be displayed on the Text Field and energy requirements in the Text Area.

3. Clear Display

Clicking on the “Clear Display” button should clear all contents from the display area and set the Text Fields and Combo Boxes to their default values.

The screenshot shows a software window with a light blue border. At the top, there are input fields for 'User Name' (containing 'John'), 'Height' (empty), and 'Expected Average Weight' (empty). Below these are 'Password' (masked with asterisks), 'Physical Activity Level (PAL)' (a dropdown menu showing 'Bed Rest'), and 'PAL level' (empty). There are radio buttons for 'Male' (selected) and 'Female', and an 'Age Group' dropdown showing '19 - 30'. A 'Login' button is to the left of the radio buttons. In the center is a large empty text area. At the bottom are four buttons: 'Load Data', 'Display energy', 'Clear', and 'Quit'.

Figure 2: Screen Display After Load Data

This screenshot shows the same application window as Figure 2, but with data populated. The 'User Name' is 'John', 'Height' is '1.7', and 'Expected Average Weight' is '63.6'. The 'Physical Activity Level (PAL)' dropdown now shows 'Vigorous', and the 'PAL level' field contains '2.2'. The 'Age Group' dropdown now shows '31 - 50'. The central text area now contains the following text: 'Hello John for your age group of 31-50, with PAL lvel vigorous, 2.2, and height of 1.7 m, you require 14.8 Mega joules energy per day. The expected average weight for your hieght is 63.6 kg.' (Note the typos 'lvel' and 'hieght' in the original image). The 'Load Data' button is now disabled, while the other buttons remain active.

Figure 3: Screen Display on Clicking “Display Rosters”

4. Quit

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

Data Structures.

You can use the classes given below as a guideline for your design.

1. enum PAL - Physical Activity Level: The descriptions and corresponding values of PALs are modelled as an enum. This is already completed and provided for your use in the course website in file, PAL.java

2. PALdata class

This is to store the PAL and corresponding energy requirement. This class is already completed and provided for your use in the course website in file, PALdat.java

3. *BMRdata class*

The purpose of this class is to store the values of age group, gender, height, weight, and bmr read from each row of the file.

- a) Include appropriate fields to store:

Height, weight, bmr which are real numbers to be stored in float data type, age group and gender which are strings.

- b) A constructor and necessary accessor and mutator methods to get and set values

- c) A toString() method that displays the BMRdata attributes.

4. *EnergyData Class*

- a) This class should have the fields to store:

BMRdata, and set of energy needs for various PALs corresponding to the BMRdata. This could be stored as an ArrayList of PALdata.

- b) It should also include constructor, mutator, and accessor methods,

- c) A toString() method for displaying the stored data.

5. *User class – an abstract class*

- a) This class should have fields to store user entered values such as name, gender, age group, and height.

This class should have:

- b) Constructor, accessor and mutator methods.

- c) A toString() method for displaying the User details.

6. *UserEnergyNeeds class*

- a) This class extends the User class. This class should have a member fields to store expected average weight, bmr corresponding to the user's age group, gender and height, and the PALdata to store chosen PAL and corresponding energy requirement. The height entered should be between 1.5 and 2m inclusive. Use validation and exception handling to handle this error.

7. *GUI Components class*

- a) This class should have the GUI components listed above.
- b) It should have the methods to set up the GUI components and the event handling methods.
- c) This class should have method to read data from the file and load data structures appropriately.
- d) This also contains main method.

5. **Software Tools for Building the Application**

You can build your application using the TextPad Editor or NetBeans. It is highly recommended that you create the GUI components using code rather than 'designer' and 'click and drag' of the NetBeans. This creates code which is not maintainable. Remember this is a good prototyping tool, but not recommended for coding.

Note: Commence with one class at a time, test it and then incrementally add the next.

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)

- BMRdata.java – Source code for the Parishioner class
- EnergyData.java – Source code for the Roster class
- User.java – Source code for Minister class
- UserEnergyNeed.java
- UserEnergyInterfaceGUI.java – Source code for the GUI components and main method.
- Report.docx – File containing UML class diagrams for the classes, BMRdata, EnergyData, User, and UserEnergyNeed. Include a test plan providing test cases, test data, expected result and description of actual result. Provide few screen shots to demonstrate evidence of your testing

Assessment Item 1 Marking criteria

S.No	Total Marks - 15	Marks Allocated	Marks Scored
1	Graphical User Interface Presentation (Use appropriate sized components and alignments)	2	
2	Design and use of appropriate data structures	2	
3	Designing classes and using objects, methods, and inheritance	3	
4	Use of exception handling (use appropriate exception classes and informative messages)	1	
5	“LogIn”, and “Load Data” button function	2	
6	“Display energy” button function	2	
7	“Clear Display” and “Quit” buttons functions	1	
8	Good coding practices(Indentation, Comments, Naming Conventions, Readability)	1	
9	Well presented report with student details, UML class diagrams, test plan, and evidence of testing	1	

Note:

1. *If your program doesn't compile or run, partial marks will be allocated by inspection of the source code.*
2. Please clarify any doubts you have by one of the means of discussing with your tutor, posting a query in the Q& A forum, or discussing with your colleagues.
3. Please do not share your source code files or report with your colleagues which may lead to plagiarism.
4. Commence your assignment work early and show your progress to your tutor from Week 4 onwards.