

Assessment details for All students

Assessment item 2

Due Date: Thursday of Week 11 (25th May) 11.45 pm AEST
Weighing: 30%

Objectives

- 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

Assessment Task: Paired Programming

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. You will be connecting to a database programmatically and executing necessary SQL queries within your java program.

Year 11 and 12 students study five subjects. There are many assessments for these subjects in different formats and the grade of these assessments are summative contributing to a student's results. Your task is to extend the Java Application developed for Assignment One such that an administrator can enter details of students with their assessment details for all enrolled subjects and set grade. Students or parents can view the details by entering a student name and year level.

The database will have the following tables:

1. Student table having studentId (primary key): a 4 digit number, studentName: string (30 characters), and yearLevel: 2 digit number
2. Subject table: In addition to the existing attributes this table will have subjectId (primary key): 2 digit number, and studentId as the foreign key
3. Assessment table: extend the Assessment attributes from Assignment One to include an Assessment Id: a single digit number starting at 1. The primary key will be a composite key consisting of subjectId and AssessmentId
4. Grades table: store the values of Grade shown in Table 3. Use the Degree of achievement as the primary key
5. StudentGrade to store the set grade for an Assessment item of a student (use composite key of StudentId, subjectId, AssessmentId, and gradeId.

Note: The table listings only show the values to be stored as per the user requirements. The primary keys are generated during the programmatic creation of the tables and used for other database operations. These keys need not be displayed to the user. The primary keys can be auto-incremented integer values. One Student should enrol in five Subjects and each Subject is enrolled by many Students. (many – to – many). You may add additional tables as required for the correct functioning of the database. Create an ERD to understand the relationship between tables.

In addition to the GUI components you have in Assignment One, include the following components.

Save Button in the bottom panel, to save the user data to the database.

The functions of the eight buttons are described below.

Initially the ‘Create Student’, ‘Load Assessment’, ‘Save’, and ‘Set Grade’ buttons remain disabled. This is shown in Figure 1.

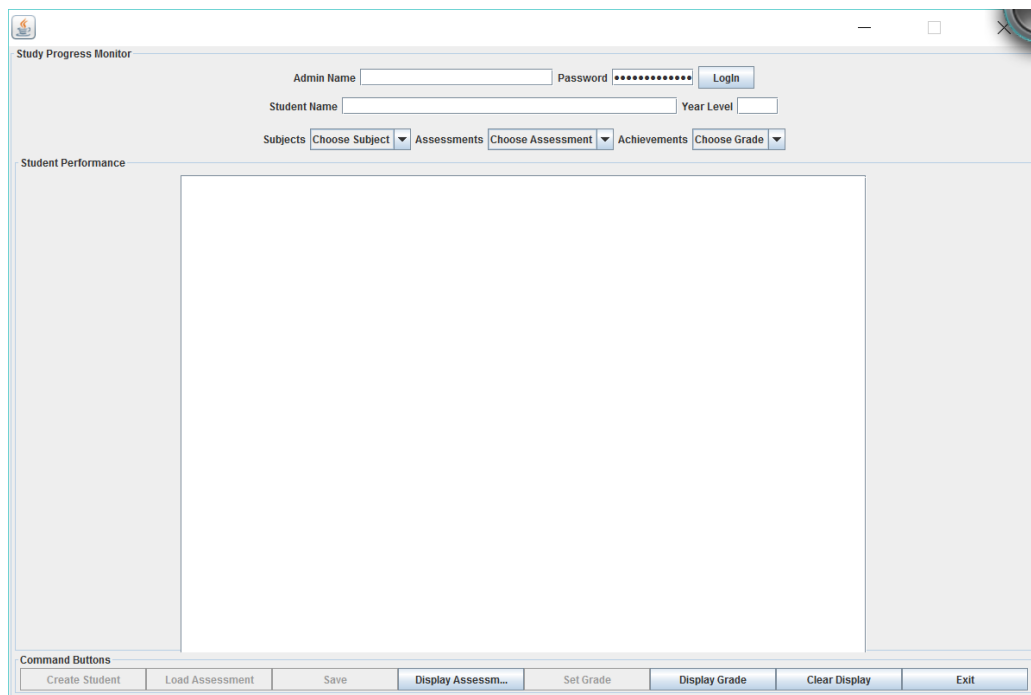


Figure 1. Screen shot of Initial Display

An administrator enters the name (admin) and password (admin), and clicks the Login button. This enables the ‘Create Student’ button for the first run.

In subsequent runs, if the ‘Save’ button has been clicked and data saved to the database, then the admin login should load the data structures from the database. So the ‘Load Assessment’ button should remain disabled. This should also enable the buttons ‘Create Student’, ‘Set Grade’ and ‘Save’.

Administrator Functions

1. Create Student

The administrator enters Student Name and Year Level using the fields on the GUI and clicks the ‘Create Student’ button which creates a student object. While creating the Student object, three core subjects of English, Mathematics B, and Biology, and the chosen elective subjects of, Business and Communication Technologies, and Religion and

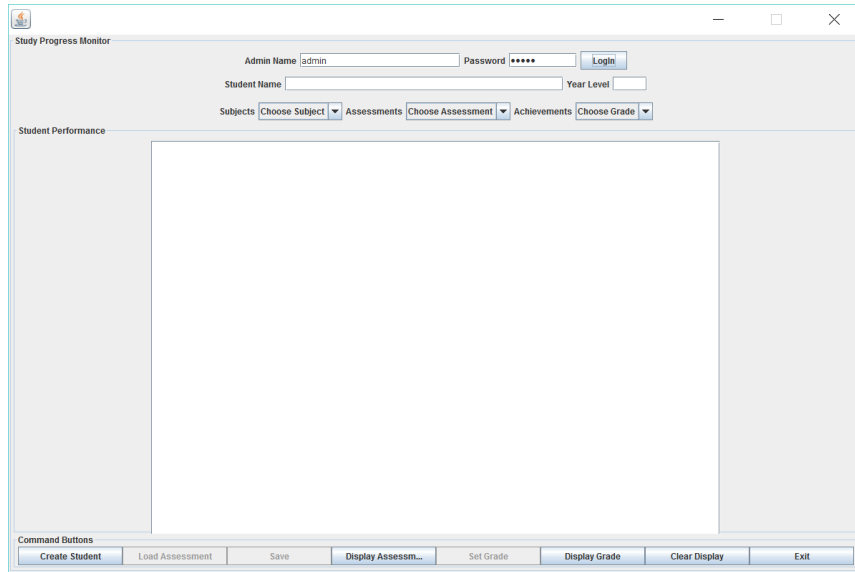


Figure 2 Screen shot after Admin Login

Ethics are added to the student object. This is done to simplify the process of going through adding each subject. The administrator can enter more than one student.

Table 1 Test Data for Students

Student Name	Year Level
John Clarke	11
Peter White	11
Lily Li	11
Lisa Soon	11
Tom Dixon	11

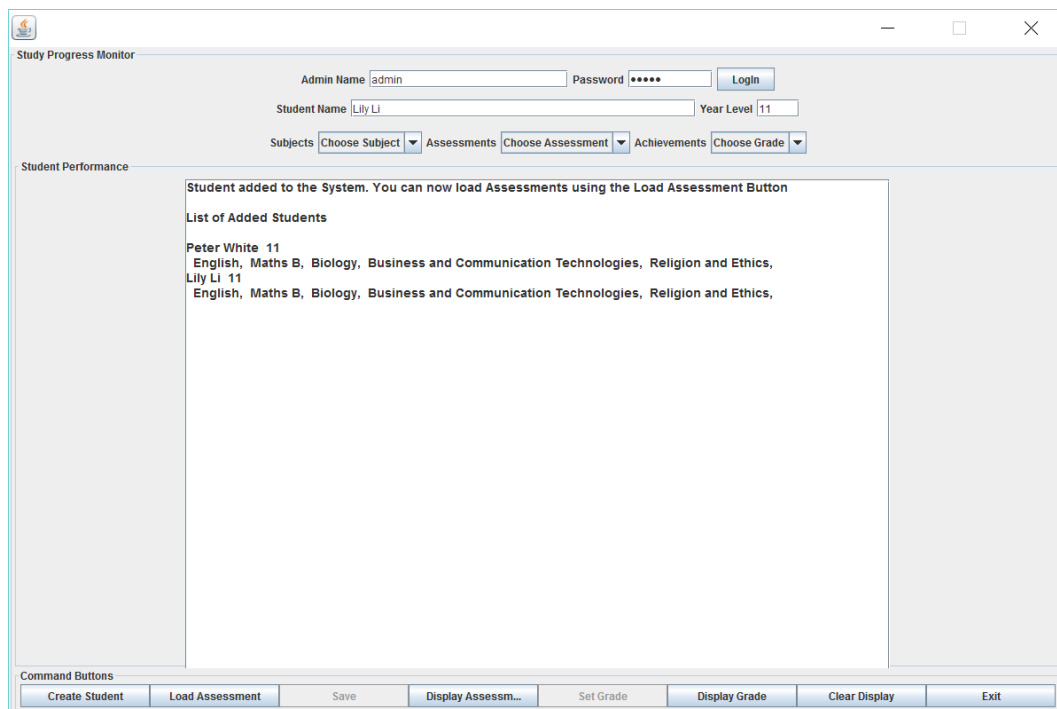


Figure 3 Screen shot after Create Student

2. Load Assessments

The 'Load Assessments' Button will be enabled after the 'Create Student' button is clicked creating a student object. The 'Load Assessments' Button can be used to read the data file which contains the Assessment List for all the subjects. After loading the file, the loaded Assessment details are added to the corresponding subjects. The first field in each assessment item in the file is the subject name and this should be used to correctly add the assessments to the corresponding subjects. The 'Load Assessments' Button should also populate the Combo Boxes for Subjects, Assessments, and Degree of achievements with corresponding set of values. After loading the assessments enable the 'Save' and 'Set Grade' buttons.

Table 2 Example Assessment Details

Description	Value
Subject	English
Assessment Id	11.1
type	Multimedia presentation
topic	interior monologue 3-4 mins Australian Identity
format	speaking
Due date	Wed 9 - Wed 16 March 2017

Note: Use the given data file named *COIT20256Ass2Data.csv* available in the Unit website and it contains the set of data required.

3. Set Grade

Clicking on the "Set Grade" button allows the administrator to set the Degree of achievement for a chosen Assessment Item, for a chosen student, and subject. The grading contains levels of achievements, knowledge, and skill. These values are given in the following Table 2. Once the administrator selects the Degree of achievement using the Combo Box, corresponding levels of knowledge and skill should be added.

Table 3. Grading Details

<i>Degree of achievement:</i>	<i>Degree of knowledge and understanding:</i>	<i>Degree of skill and use of skill:</i>
Very high	thorough understanding	uses a high level of skill in both familiar and new situations
High	clear understanding	uses a high level of skill in familiar situations, and is beginning to use skills in new situations
Sound	understanding	uses skills in situations familiar to them
Developing	understands aspects of	uses varying levels of skill in situations familiar to them
Emerging	basic understanding	beginning to use skills in familiar situations

4. Save

This button should allow the entered Student data, subject, assessment details and grades to be saved to the database.

General Functions

5. Display Assessments

This button has the same functionality as in Assignment One. This should display all the assessment details of a chosen subject in the Text Area as shown in Figure 4. The last value 'false' indicates the assessment item is not yet graded. You can use 'not graded' instead of false which may be more meaningful.

The screenshot shows the 'Study Progress Monitor' window. At the top, there are fields for 'Student Name' (Merlin) and 'Year Level' (11). Below these are dropdown menus for 'Subjects' (English), 'Assessments' (English, 11.1, multimedia presentation), and 'Achievements' (very high). The main text area, titled 'Student Performance', displays the following text:

```
English:
11.1. multimedia presentation, interior monologue 3-4 mins Australian Identity, speaking, Wed 9 - Wed 16 Mar
11.2. Assignment, Film Study Australia and Australians, writing, Friday 13 May 2017, false
11.3. exam, essay on the novel 90 mins, writing, Mon 10 - Fri 14 April 2017, false
```

At the bottom, there is a 'Command Buttons' section with buttons for 'Create Student', 'Load Assessment', 'Display Assessment', 'Set Grade', 'Display Grade', 'Clear Display', and 'Exit'.

Figure 4 Display Assessment

6. Display Grade

This can be used to display the grades of various assessments set already in the TextArea as shown in Figure 5. A parent or student has to enter the student name and year level before clicking the Display Grade button.

The screenshot shows the 'Study Progress Monitor' window. At the top, there are fields for 'Admin Name' and 'Password' with a 'Login' button. Below these are fields for 'Student Name' (Lily Li) and 'Year Level' (11). Below these are dropdown menus for 'Subjects' (Choose Subject), 'Assessments' (Choose Assessment), and 'Achievements' (Choose Grade). The main text area, titled 'Student Performance', displays the following text:

```
English 11.1. Multimedia presentation, interior monologue 3-4 mins Australian Identity, speaking
Achievement: Very high
Knowledge: thorough understanding
Skill: uses a high level of skill in both familiar and new situations

Maths B 11.1 Assignment undefined writing
Achievement: Sound
Knowledge: understanding
Skill: uses skills in situations familiar to them
```

At the bottom, there is a 'Command Buttons' section with buttons for 'Create Student', 'Load Assessment', 'Save', 'Display Assessm...', 'Set Grade', 'Display Grade', 'Clear Display', and 'Exit'.

Figure 5 Display Assessment

An admin can display the grades of all students enrolled for a chosen subject, or a chosen Assessment.

7. Clear Display

User clicks this button to clear the TextArea, TextFields, and set the ComboBoxes to the default value.

8. Exit

This should allow the user to quit the application.

4. Quit

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

Data Structures.

DatabaseUtility class: Create this class to implement all the tasks related to creation of database and table, extracting records from the database. Use Prepared statements to get the user input from GUI interface and use them for populating database.

Also use prepared statements to execute query to extract data and store the records in a linked list.

You can use the classes used in Assignment One. You don't have to create a class for the admin user. Use Linked list instead of ArrayList to store data belonging to multiple objects.

Paired Programming

Complete the assignment working in pairs. You can choose your partner yourself. You should ensure that you inform your tutor about the partner and if you have any problem with choosing a partner, please inform your tutor. Distance students also can work in pairs. If there are any problems please contact the Unit Coordinator.

Individual Report

Your report should be written individually and submitted by each one of you. In your report clearly show your partner for the application development.

1. Write how the Study Progress Monitor can be extended to allow teachers teaching each subject can enter their results. You need to write about the additional data structures required for this.
2. Provide your view of using any data structures such as stacks, queues or sets to improve coding efficiency.
3. Clearly indicate the tasks / parts of application development carried out by you.
4. Write you experience of using paired on programming clearly indicating the advantages and disadvantages if any.
5. Include a test plan, and user instructions.

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 creating each class saving to the database tables and progress step by step.

Assignment Submission

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

- Student.java – Source code for the Student class
- Subject.java – Source code for the Subject class
- Assessment.java – Source code for Assessment class
- MarkedAssessment.java
- DatabaseUtility.java – Source code for programmatic operation of the database
- StudyMonitorInterface.java – Source code for the GUI components, other required methods as outlined above and the main method.

The application program should create the database and necessary tables. So there is no need to submit any database files.

Individual Submission

- **Report.docx** – File containing, test plan, and test details

Assessment Item 2 Marking criteria

S.No	Total Marks - 30	Marks Allocated	Marks Scored
1	Graphical User Interface Presentation (Use appropriate sized components and alignments)	3	
2	Design and use of appropriate data structures	3	
3	Design and implementation of DatabaseUtility class enabling correct creation of database and access	4	
4	Use of exception handling (use appropriate exception classes and informative messages)	2	
5	“LogIn” button function	2	
6	Save button function	2	
7	Correct Display grade button function for the student/ parent and admin using different options	3	
8	Programmatic creation of tables and correct database connectivity	4	
9	Good coding practices (Indentation, Comments, Naming Conventions, Readability)	2	
10	Individual Report: Well presented report with student details, views on extending the system, use of data structures, test plan, and test details	4	
11	Report Format: correct spelling, grammar, document format, and readability	1	
	Penalties		
	Late Penalty - 1.5 marks(5% per working day)		
	Plagiarism (as per policy)		
	Total	30	

Note: if your program doesn't compile or run, partial marks will be allocated by inspection of the source code.